EXPONENTIALLY-CONVERGENT MONTE CARLO FOR THE ONE-DIMENSIONAL TRANSPORT EQUATION

A Thesis

by

JACOB ROSS PETERSON

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

MASTER OF SCIENCE

Chair of Committee,	Jim Morel
Co-Chair of Committee,	Jean Ragusa
Committee Member,	Bani Mallick
Head of Department,	Yassin Hassan

May 2014

Major Subject: Nuclear Engineering

ABSTRACT

An exponentially-convergent Monte Carlo (ECMC) method is analyzed using the one-group, one-dimension, slab-geometry transport equation. The method is based upon the use of a linear discontinuous finite-element trial space in position and direction to represent the transport solution. A space-angle h-adaptive algorithm is employed to maintain exponential convergence after stagnation occurs due to inadequate trial-space resolution. In addition, a biased sampling algorithm is used to adequately converge singular problems. Computational results are presented demonstrating the efficacy of the new approach. We tested our ECMC algorithm against standard Monte Carlo and found the ECMC method to be generally much more efficient. For a manufacture solution the ECMC algorithm was roughly 200 times more effective than the standard Monte Carlo. When considering a highly singular pure attenuation problem, the ECMC method was roughly 4000 times more effective.

ACKNOWLEDGEMENTS

First and forememost I thank my advisors Dr. Jim Morel and Dr. Jean Ragusa for their help and guidance. Without them none of this would have been possible.

Thank you to Dr. Bani Mallick as a committee member, and to Dr. Yassin Hassan as department head. I also thank the faculty and staff in the Nuclear Engineering department for all they have done.

Finally, I thank my family and friends. My wife Katherine Peterson has given me neverending love and support. Without her I couldn't handle the stress of graduate school. My parents Sara Peterson and Chris Peterson have always being there for me. They have believed in me my entire life without fail. And my buds in the Singing Cadets for helping me keep my sanity during the process.

NOMENCLATURE

ECMC	Exponentially-Convergent Monte Carlo	
$\mathbf{L}$	linear system	
u	system solution	
q	system source	
$\tilde{u}$	ECMC solution approximation	
r	residual associated with the approximation	
ϵ	pointwise error in ECMC approximation	
$\hat{\mathbf{L}}^{-1}$	solution method to approximate $\mathbf{L}^{-1}$	
δ	approximation to ϵ from $\hat{\mathbf{L}}^{-1}$	
x	position	cm
μ	angular cosine	
ψ	angular flux	$\frac{n-cm}{cm^3-ster-s}$
σ_t	total neutron cross section	$\frac{\text{reactions}}{n-cm}$
σ_s	scattering neutron cross section	$\frac{\text{scatterings}}{n-cm}$
σ_a	absorption neutron cross section	$\frac{\text{absorptions}}{n-cm}$
ϕ	scalar flux	$\frac{n-cm}{cm^3-s}$
Q	internal source	$\frac{n}{cm^3-ster-s}$
T	domain length	cm
ψ^+_{inc}	incoming flux from the left	$\frac{n-cm}{cm^3-ster-s}$
ψ^-_{inc}	incoming flux from the right	$\frac{n-cm}{cm^3-ster-s}$
$\tilde{\psi}$	FEM representation of the angular flux	$\frac{n-cm}{cm^3-ster-s}$
$\tilde{\phi}$	FEM representation of the scalar flux	$\frac{n-cm}{cm^3-s}$
$\tilde{Q}$	FEM representation of the internal source	$\frac{n}{cm^3-ster-s}$

$\psi^a_{i,m}$	FEM projection coefficient: average value	$\frac{n-cm}{cm^3-ster-s}$
$\psi^x_{i,m}$	FEM projection coefficient: slope in position	$\frac{n-cm}{cm^3-ster-s}$
$\psi^\mu_{i,m}$	FEM projection coefficient: slope in direction	$\frac{n-cm}{cm^3-ster-s}$
h_i	width of cell column i in position	cm
h_m	width of cell row m in direction	
$\tilde{r}$	residual associated with the FEM representation	$\frac{n}{cm^3-ster-s}$
$\tilde{r}^{int}$	cell interior residual piece	$\frac{n}{cm^3-ster-s}$
$\tilde{r}^{fac}$	cell face residual piece	$\frac{n}{cm^3-ster-s}$
$I^{int}_{i,m}$	absolute integral of the interior residual in cell (i,m)	$\frac{n}{s}$
$I^{fac}_{i,m}$	absolute integral of the face residual in cell (i,m)	$\frac{n}{s}$
ρ	random number generated $\in (0,1)$	
s	path length traced in a cell	cm
$\bar{x}$	average position a particle has in a cell	cm
w	particle weight	
N_p	number of particle histories in a batch	
x'	next position sampled	cm
α	magnitude of the residual-norm ratio	
ξ_i	the jump error indicator for cell i	
N_C	number of cells in the mesh	

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Brief History of the Monte Carlo Method

The Monte Carlo method was first utilized by Los Alamos National Laboratory. While various statistical methods had been theorized prior, it was not until the invention of the computer that the full simulations could be realized. In 1945, researchers at the University of Pennsylvania in Philadelphia created the ENIAC: the first electronic computer. Scientists at Los Alamos reviewed the device in 1946 and were intrigued by the possibilities. They tested the method in 1947, and the method was deemed effective and favorable. In 1952 Los Alamos built their own computer, the MANIAC, for the Monte Carlo simulations [1]. They called the simulation code they created MCS. Features were added to this code over the years, and in 1977 the codes culminated in Los Alamos National Lab's Monte Carlo N-Particle transport code (MCNP).

Other entities have developed Monte Carlo codes over the years. Notable ones include ITS TIGER developed at Sandia National Laboratory [2], TART developed at Lawrence Livermore National Laboratory [3, 4], VIM developed at Argonne National Laboratory [5, 4], TRIPOLI developed by the Commissariat à l'énergie atomique, CEA/SACLAY, Cedex, France [6, 4], SHIELD developed at the Institute for Nuclear Research of the Russian Academy of Science, Moscow, Russia [7, 4], MONK/MCBEND developed by Serco in the United Kingdom [8, 4], Geant4 a worldwide collaboration initially developed at CERN [9], and OpenMC developed originally at Massachusetts Institute of Technology [10].

1.2 Variance Reduction Techniques

The Monte Carlo method involves randomly sampling variables, applying a system to these variables, and tallying the calculated results. The tallies are averaged together to obtain the quantities of interest. By the Central Limit Theorem, this algorithm has a convergence rate of $1/\sqrt{N}$, where N is the number of histories ran. That is, to double the accuracy, four times as many particles need to be ran. This can make complicated systems take prohibitively long to simulate.

Variance reduction methods reduce the error without using more particles, thus allowing these complicated systems to be feasibly solved. Presented here are a few of the common methods, as utilized in MCNP [11]. They are splitting, weight dispersion, exponential transformation, forced collisions, implicit absorption, correlated sampling, and source biasing. All of these involve the weight of the particle, which is discussed more in Chapter 2.

Splitting is utilized when there are areas in the space-energy domain of less importance relative to that region. If a particle enters the less important region from a more important region, there is a certain probability it will be terminated. The particles that survive are given a higher weight to maintain accuracy in the simulation. Conversely, if a particle enters a more important region, more particles are created. These split particles are given lower weights to maintain the solution.

Weight dispersion helps keep the particle weights within a certain bound. If the weights fall too low, the particle will terminate with a certain probability. If not, the weight is raised to account for the fewer particles. Similarly, if the weight of a particle becomes too large the code will split the particle and decrease the weights.

When large transport distances need to be simulated, exponential transformation is used. The distance between collisions is artificially increased and the weight is

correspondingly artificially decreased.

A particle can be forced to undergo certain collisions, namely when it enters a region almost transparent to it. The particle is split into two parts, collided an uncollided. The weights are appropriately set to represent the probability of collision.

In a normal Monte Carlo simulation, the particle is killed with an absorption interaction. Implicit absorption doesn't kill the particle, but rather reduces the weight of the particle based on the absorption probability. This permits important particles to survive longer.

Slight changes in the problem specification can create small variations in the solution. These can be masked by the statistical fluctuations in a normal Monte Carlo simulation. Normally a particle's random number sequence begins where the previous particle ended. Changing the problem specification has the potential to change every particle's random walk. Correlated sampling always starts the i^{th} history at a specified point within the random number sequence, regardless of the previous particles' simulations. This reduces the statistical fluctuations caused by slight changes in the problem.

Source biasing generates more particles in areas of phase space with more importance. To maintain the correct solution, the weights are lowered in these regions. This method is utilized in this Thesis, as explained in Section 2.5.

1.3 Previous Work on Exponentially-Convergent Monte Carlo

As opposed to the $1/\sqrt{N}$ convergence of regular Monte Carlo methods, it is possible to define a method which converges exponentially [12]. Exponentially-convergent Monte Carlo (ECMC) methods have previously been shown to be practical for finite-dimensional systems such as the discretized radiative diffusion equations [13]. They have also been demonstrated for the continuous neutron transport equation [14, 15],

but only for homogeneous 1-D problems using Case eigenfunction-based algorithms impractical for realistic problems. A successful approach was demonstrated for the 3-D spatially-continuous S_n equations using global polynomial trial spaces [16], but further generalizations of the approach were not successful. It has recently been demonstrated for the simple two-stream transport equations that an ECMC approach based upon a spatial linearly-discontinuous finite-element trial spaces can yield exponential convergence [17].

1.4 Motivation for Current Work

The failure of the global polynomial approach was related to the use of highly oscillatory trial spaces that generated excessive numerical error in the solution. The finite-element approach has significant advantages when compared to previous approaches. First, one can achieve convergence of the solution without increasing the oscillatory nature of the trial space by refining the mesh. Conversely, global polynomial trial spaces become increasingly oscillatory as the order is increased. Second, finite-element trial spaces can be used to efficiently represent fully-realistic solutions. Third, the use of finite-element spaces will enable the seamless coupling of Monte Carlo transport physics with other physics components in multiphysics calculations.

2. ALGORITHM DESCRIPTIONS

2.1 ECMC General Algorithm

The ECMC algorithm can be used on any system which is solvable via Monte Carlo, expressed in the following form:

$$\mathbf{L}[u] = q, \tag{2.1}$$

where $\mathbf{L}$ is the linear system with source q, and u is the solution to this system. The i^{th} iteration solution approximation is represented by $\tilde{u}^{(i)}$. The i^{th} residual is thus

$$r^{(i)} = q - \mathbf{L}[\tilde{u}^{(i)}]. \tag{2.2}$$

Substituting Eq. (2.1) for the source in Eq. (2.2) yields

$$r^{(i)} = \mathbf{L}[u] - \mathbf{L}[\tilde{u}^{(i)}] = \mathbf{L}[u - \tilde{u}^{(i)}] = \mathbf{L}[\epsilon^{(i)}], \tag{2.3}$$

where $\epsilon^{(i)}$ is the pointwise error of the i^{th} solution approximation. Note that this formulation is the same linear system as Eq. (2.1) with the residual as the source. Any method that adequately solves the original system can also solve Eq. (2.3). Introducing $\hat{\mathbf{L}}^{-1}$ as the method to solve the system, the general algorithm becomes as follows:

$$r^{(i)} \leftarrow q - \mathbf{L}[\tilde{u}^{(i)}] \tag{2.4a}$$

$$\delta^{(i)} \leftarrow \hat{\mathbf{L}}^{-1}[r^{(i)}] \tag{2.4b}$$

$$\tilde{u}^{(i+1)} \leftarrow \tilde{u}^{(i)} + \delta^{(i)} \tag{2.4c}$$

Since $\hat{\mathbf{L}}^{-1}$ may not exactly represent $\mathbf{L}^{-1}$, $\delta^{(i)}$ is the typical Monte Carlo approximation of $\epsilon^{(i)}$. The algorithm begins with $\tilde{u}^{(0)}$ uniformly 0, so the first iteration is simply solving a single batch on the original system. If the Monte Carlo solver estimates the solution with sufficient accuracy, each successive batch estimates a smaller component of the solution and exponential convergence is obtained as a function of the number of batches (and hence as a function of the number of particle histories).

Because the transport equation is continuous, the angular flux solution has an infinite number of degrees of freedom and hence must be approximately represented in general. For a given angular flux representation, exponential convergence will be obtained until the remaining error in the Monte Carlo solution can no longer be accurately represented in the trial space. At this point the Monte Carlo solution stagnates at an average fixed error level. Note that the solution obtained using the ECMC approach does not represent a standard finite-element solution but rather represents a least-squares projection of the exact solution onto the trial space. This is in general a far more accurate approximation than a standard finite-element approximation. In standard Monte Carlo calculations, one usually computes projections of the solution rather than the solution itself. For instance, the average scalar flux within a volume represents a projection of the solution. Thus no desired information relating to the solution is necessarily lost by using the ECMC method.

2.2 Transport Equation

2.2.1 One-Group, One-Dimension Transport

For this research the steady-state, one-dimensional form of the transport equation was used:

$$\begin{aligned} \mu\frac{\partial}{\partial x}\psi(x,\mu)+\sigma_t(x)\psi(x,\mu)=\frac{\sigma_s(x)}{4\pi}\phi(x)+Q(x,\mu), \quad x\in[0,T],\ \mu\in[-1,+1], \\ \psi(x=0,\mu>0)=\psi^{+}_{inc}(\mu), \\ \psi(x=T,\mu<0)=\psi^{-}_{inc}(\mu). \end{aligned} \tag{2.5}$$

This formulation has one energy group, isotropic scattering, and no fission. The scalar flux is defined as

$$\phi=2\pi\int_{-1}^{+1}\psi\,d\mu. \tag{2.6}$$

2.2.2 Linear-Discontinuous Discretization

The finite-element trial space is defined on a space-direction mesh with rectangular cells. The indexing for a single cell before adaptation occurs is illustrated in Fig. 2.1.

The initial grid is defined by spatial vertices, $\{x_i\}_{i=1/2}^{N+1/2}$, where $x_{1/2}=0$ and $x_{N+1/2}=T$; and direction vertices, $\{\mu_m\}_{m=1/2}^{M+1/2}$, where $\mu_{1/2}=-1$, $\mu_{M+1/2}=+1$. The finite-element representation for the solution associated with cell (i,m) is as follows:

$$\tilde{\psi}(x,\mu)\big|_{i,m}=\tilde{\psi}_{i,m}(x,\mu)=\psi^{a}_{i,m}+\psi^{x}_{i,m}\frac{2}{h_i}(x-x_i)+\psi^{\mu}_{i,m}\frac{2}{h_m}(\mu-\mu_m). \tag{2.7}$$

This representation is upwinded at the cell boundaries. This means that the value on a spatial face is determined by the cell the particle is leaving, as shown in Fig. 2.2.

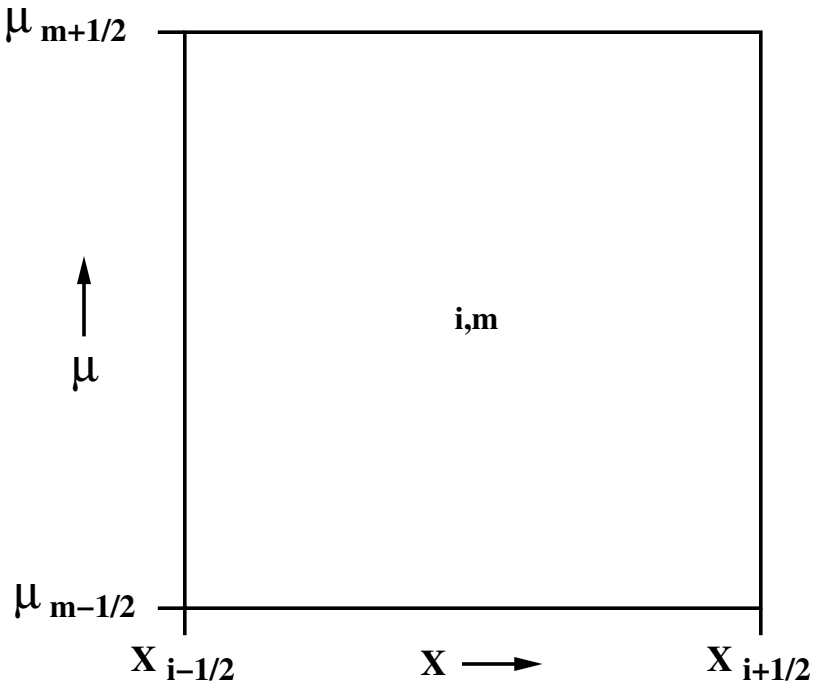


Figure 2.1: Indexing and Bounds for the Space-Direction Cell (i, m).

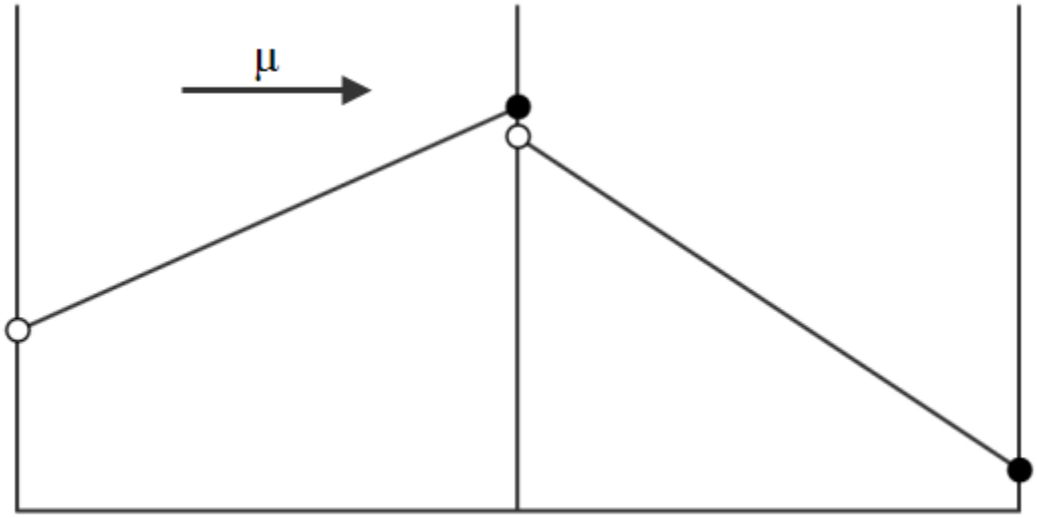


Figure 2.2: Upwind Example for $\mu > 0$. The face value is represented by the closed circle.

Thus the bounds for the projection in cell (i, m) are

$$\begin{cases} x \in (x_{i-1/2}, x_{i+1/2}],\ \mu \in (\mu_{m-1/2}, \mu_{m+1/2}] & \mu > 0, \\ x \in [x_{i-1/2}, x_{i+1/2}),\ \mu \in [\mu_{m-1/2}, \mu_{m+1/2}) & \mu < 0. \end{cases} \tag{2.8}$$

This ensures a unique definition for $\tilde{\psi}$ on each spatial face in the domain. The definition at $\mu = 0$ is not necessary because there are no μ-derivatives in the transport equation. Similar discretizations are defined for Q and ϕ.

2.3 Sampling and Counting

2.3.1 Residual Calculation

For this equation the residual is defined as

$$\tilde{r} = \tilde{Q} + \frac{\sigma_s}{4\pi}\tilde{\phi} - \sigma_t\tilde{\psi} - \mu\frac{\partial\tilde{\psi}}{\partial x}. \tag{2.9}$$

Due to the discontinuities of $\tilde{\psi}$ at each cell interface, the derivative involves a delta function.

$$\frac{\partial\tilde{\psi}_{i,m}}{\partial x} = \begin{cases} \psi^x_{i,m}\frac{2}{h_i} + \delta(x - x_{i-1/2})\left(\tilde{\psi}_{i,m} - \tilde{\psi}_{i-1,m}\right) & \mu > 0, \\ \psi^x_{i,m}\frac{2}{h_i} + \delta(x - x_{i+1/2})\left(\tilde{\psi}_{i,m} - \tilde{\psi}_{i+1,m}\right) & \mu < 0. \end{cases} \tag{2.10}$$

The delta term can be treated as a face source for the cell. The residual is thus expressed as two pieces, internal and face, as follows:

$$\tilde{r}^{int}(x,\mu)\big|_{i,m} = r^a_{i,m} + r^x_{i,m}\frac{2}{h_i}(x - x_i) + r^\mu_{i,m}\frac{2}{h_m}(\mu - \mu_m), \tag{2.11a}$$

$$r^a_{i,m} = Q^a_{i,m} + \frac{\sigma_s}{4\pi}\phi^a_{i,m} - \sigma_t\psi^a_{i,m} - \mu_m\frac{2}{h_i}\psi^x_{i,m}, \tag{2.11b}$$

$$r^x_{i,m} = Q^x_{i,m} + \frac{\sigma_s}{4\pi}\phi^x_{i,m} - \sigma_t\psi^x_{i,m}, \tag{2.11c}$$

$$r^\mu_{i,m} = Q^\mu_{i,m} - \sigma_t\psi^\mu_{i,m} - \frac{h_m}{h_i}\psi^x_{i,m}; \tag{2.11d}$$

$$\tilde{r}^{fac}(x,\mu)\big|_{i,m} = \delta_{i,m}\mu\left(r^{c1}_{i,m} + r^{c2}_{i,m}\frac{2}{h_m}(\mu - \mu_m)\right), \tag{2.12a}$$

$$\delta_{i,m} = \begin{cases} \delta(x - x_{i-1/2}) & \mu > 0, \\ \delta(x - x_{i+1/2}) & \mu < 0, \end{cases} \tag{2.12b}$$

$$r^{c1}_{i,m} = \begin{cases} \left(\psi^a_{i-1,m} + \psi^x_{i-1,m}\right) - \left(\psi^a_{i,m} - \psi^x_{i,m}\right) & \mu > 0, \\ \left(\psi^a_{i+1,m} - \psi^x_{i+1,m}\right) - \left(\psi^a_{i,m} + \psi^x_{i,m}\right) & \mu < 0, \end{cases} \tag{2.12c}$$

$$r^{c2}_{i,m} = \begin{cases} \psi^\mu_{i-1,m} - \psi^\mu_{i,m} & \mu > 0, \\ \psi^\mu_{i+1,m} - \psi^\mu_{i,m} & \mu < 0. \end{cases} \tag{2.12d}$$

For cells on the edge of the domain, $\left(\psi^a_{i-1,m} + \psi^x_{i-1,m}\right)$, $\left(\psi^a_{i+1,m} - \psi^x_{i+1,m}\right)$, $\psi^\mu_{i-1,m}$, and $\psi^\mu_{i+1,m}$ are determined from the face source distributions ψ^+_{inc} and ψ^-_{inc}. The L^1

norm of the residual is calculated over the entire domain:

$$
\begin{aligned}
||\tilde{r}||_{L^1} &= \int_0^T \int_{-1}^{+1} |\tilde{r}| d\mu \, dx \\
&= \sum_{i=1}^{N} \sum_{m=1}^{M} \left[\int_{x_{i-1/2}}^{x_{i+1/2}} \int_{\mu_{m-1/2}}^{\mu_{m+1/2}} |\tilde{r}| d\mu \, dx \right] \\
&= \sum_{i=1}^{N} \sum_{m=1}^{M} \left[\int_{x_{i-1/2}}^{x_{i+1/2}} \int_{\mu_{m-1/2}}^{\mu_{m+1/2}} |\tilde{r}^{int} + \tilde{r}^{fac}| d\mu \, dx \right] \\
&\approx \sum_{i=1}^{N} \sum_{m=1}^{M} \left[\int_{x_{i-1/2}}^{x_{i+1/2}} \int_{\mu_{m-1/2}}^{\mu_{m+1/2}} |\tilde{r}^{int}| d\mu \, dx + \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{\mu_{m-1/2}}^{\mu_{m+1/2}} |\tilde{r}^{fac}| d\mu \, dx \right] \\
&= \sum_{i=1}^{N} \sum_{m=1}^{M} \left[I_{i,m}^{int} + I_{i,m}^{fac} \right],
\end{aligned}
\tag{2.13}
$$

where $I_{i,m}^{int}$ is the value of the interior integral and $I_{i,m}^{fac}$ is the value of the face integral. The integrals over a cell are calculated exactly. Since it is possible for the residual to be negative within a cell, these integrals must be taken with care. The exact integration is shown in Appendix A.

2.3.2 Source Sampling and Rejection Method

The ECMC algorithm uses the residual as the source. To determine the starting cell the relative magnitudes of the L^1 norms within the cells are used, as described in the following algorithm:

generate random number $\rho \in (0, 1)$
define temporary sum $TS = 0$
FOR $i = 1, N$
  FOR $m = 1, M$
    $TS \leftarrow TS + \frac{I^{int}_{i,m} + I^{fac}_{i,m}}{||\vec{r}||_{L^1}}$
    IF $\rho < TS$
      define the starting cell as (i, m)
      EXIT the algorithm
    END
  END
END

Once the starting cell is determined, the code determines whether the particle starts on the face or within the cell. Another random number ρ is generated and compared to $\frac{I^{fac}_{i,m}}{I^{int}_{i,m} + I^{fac}_{i,m}}$. If ρ is less than this quantity, then the particle is sampled on the face; else the particle is sampled within the interior of the cell.

The position and direction of the particle are determined via the rejection method. The properties are determined using the absolute value of the residual function. The first step in the rejection method is determining the maximum value of the function.

For the interior particles, this is simply the values on the corners. Namely,

$$\begin{aligned}\mathrm{MAX}\left(\left|\psi^a_{i,m}+\psi^x_{i,m}+\psi^\mu_{i,m}\right|,\left|\psi^a_{i,m}+\psi^x_{i,m}-\psi^\mu_{i,m}\right|,\right.\\ \left.\left|\psi^a_{i,m}-\psi^x_{i,m}+\psi^\mu_{i,m}\right|,\left|\psi^a_{i,m}-\psi^x_{i,m}-\psi^\mu_{i,m}\right|\right).\end{aligned} \tag{2.14}$$

On the cell edges, the residual is quadratic in μ. Thus it is possible for the maximum value to occur elsewhere than the corners. The maximum value the residual can take is thus

$$\begin{aligned}\mathrm{MAX}\left(\left|\mu_m+\frac{h_m}{2}\right|\left|r^{c1}_{i,m}+r^{c2}_{i,m}\right|,\left|\mu_m-\frac{h_m}{2}\right|\left|r^{c1}_{i,m}-r^{c2}_{i,m}\right|,\right.\\ \left.\left|\mu_r\left(r^{c1}_{i,m}+r^{c2}_{i,m}\frac{2}{h_m}(\mu_r-\mu_m)\right)\right|\right).\end{aligned} \tag{2.15}$$

where μ_r is the location of the maximum value:

$$\mu_r=\frac{\mu_m}{2}-\frac{h_m}{4}\frac{r^{c1}_{i,m}}{r^{c2}_{i,m}}. \tag{2.16}$$

The last term is only considered if $\mu_r\in(\mu_m-\frac{h_m}{2},\mu_m+\frac{h_m}{2})$, or

$$\left|\frac{r^{c1}_{i,m}}{r^{c2}_{i,m}}+2\frac{\mu_m}{h_m}\right|<2. \tag{2.17}$$

The rejection method applied to an arbitrary function $f(x, \mu)$ (with f_{max} pre-calculated) is as follows:

BEGIN LOOP

- generate 3 random numbers $\rho_1, \rho_2, \rho_3 \in (0, 1)$
- calculate $f_1 \leftarrow \rho_1 f_{max}$
- calculate $x_2 \leftarrow x_i + h_x \left(\rho_2 - \frac{1}{2}\right)$
- calculate $\mu_2 \leftarrow \mu_m + h_m \left(\rho_3 - \frac{1}{2}\right)$
- calculate $f_2 \leftarrow f(x_2, \mu_2)$
- IF $f_1 < f_2$
  - define the starting position as x_2 and the starting direction as μ_2
  - EXIT the algorithm
- END

END

The function used for the rejection method is the absolute value of the residual function. The weight of the particle is either $+1$ or -1, depending on the sign of the residual at the sampled location.

2.3.3 Tallies

The particle tallies compute the flux average, the x-slope, and the μ-slope. The formulas are derived by taking moments of Eq. (2.7). The zeroth moment is as

follows:

$$\begin{aligned}\int_{cell(i,m)} \tilde{\psi}_{i,m} dx d\mu &= \int_{cell(i,m)} \psi^a_{i,m} + \psi^x_{i,m}\frac{2}{h_i}(x - x_i) + \psi^\mu_{i,m}\frac{2}{h_m}(\mu - \mu_m) dx d\mu \\ &= \psi^a_{i,m} \int_{cell(i,m)} dx d\mu \\ &= \psi^a_{i,m} h_i h_m.\end{aligned} \tag{2.18}$$

Thus $\psi^a_{i,m}$ is an estimate of the integral

$$\frac{1}{h_i h_m} \int_{cell(i,m)} \psi dx d\mu. \tag{2.19}$$

This integral is approximated by averaging the path lengths within the cell. Consider a single particle streaming through cell (i, m), from x_1 to x_2 with direction μ_0. The integral of the resulting angular flux is

$$\int_{cell(i,m)} \psi_0 dx d\mu = \int_{x_1}^{x_2} \frac{1}{\mu_0} dx = \frac{x_2 - x_1}{\mu_0} = s. \tag{2.20}$$

The first x-moment over the cell is weighted with respect to the cell midpoint:

$$\begin{aligned}\int_{cell(i,m)} (x - x_i)\tilde{\psi}_{i,m} dx d\mu &= \psi^x_{i,m}\frac{2}{h_i} \int_{cell(i,m)} (x - x_i)^2 dx d\mu \\ &= \psi^x_{i,m}\frac{h_i^2 h_m}{6}.\end{aligned} \tag{2.21}$$

Thus $\psi^x_{i,m}$ is an estimate of the integral

$$\frac{6}{h_i^2 h_m} \int_{cell(i,m)} (x - x_i)\psi dx d\mu. \tag{2.22}$$

Considering the single particle,

$$\int\limits_{cell(i,m)} (x-x_i)\psi_0 dx d\mu = \int\limits_{x_1}^{x_2} \frac{x-x_i}{\mu_0} dx = \frac{x_2-x_1}{\mu_0}\left(\frac{x_2+x_1}{2}-x_i\right) = s(\bar{x}-x_i). \quad (2.23)$$

The first μ-moment over the cell is weighted with respect to the cell midpoint.

$$\begin{aligned}\int\limits_{cell(i,m)} (\mu-\mu_m)\tilde{\psi}_{i,m} dx d\mu &= \psi^{\mu}_{i,m}\frac{2}{h_m}\int\limits_{cell(i,m)} (\mu-\mu_m)^2 dx d\mu \\ &= \psi^{\mu}_{i,m}\frac{h_i h_m^2}{6}.\end{aligned} \quad (2.24)$$

Thus $\psi^{\mu}_{i,m}$ is an estimate of the integral

$$\frac{6}{h_i h_m^2}\int\limits_{cell(i,m)} (\mu-\mu_m)\psi dx d\mu. \quad (2.25)$$

The angular distribution of the single particle is a delta function. Thus the integral is

$$\int\limits_{cell(i,m)} (\mu-\mu_m)\psi_0 dx d\mu = \int\limits_{x_1}^{x_2} \frac{\mu_0-\mu_m}{\mu_0} dx = \frac{x_2-x_1}{\mu_0}(\mu_0-\mu_m) = s(\mu_0-\mu_m). \quad (2.26)$$

The flux average tally is thus computed as

$$\delta^a_{i,m} \approx \frac{1}{h_i h_m N_p}\sum_{j=0}^{N_p} s_{j,i,m} w_j, \quad (2.27)$$

where $s_{j,i,m}$ is the path length particle j travels within cell (i,m) and w_j is the weight associated with particle j. The x-slope tally is

$$\delta^x_{i,m} \approx \frac{6}{h_i^2 h_m N_p} \sum_{j=0}^{N_p} (\bar{x}_{j,i,m} - x_i) s_{j,i,m} w_j, \tag{2.28}$$

where $\bar{x}_{j,i,m}$ is the average position within cell (i,m) for particle j. The μ-slope tally is

$$\delta^\mu_{i,m} \approx \frac{6}{h_i h_m^2 N_p} \sum_{j=0}^{N_p} (\mu_j - \mu_m) s_{j,i,m} w_j, \tag{2.29}$$

where μ_j is the direction associated with particle j.

2.3.4 Particle Tracking

Once the starting position is determined, the particle history is continued until it escapes the system or it is absorbed. First, the distance to the next collision is randomized. The probability that a particle will travel a distance s to its next collision is given by

$$\sigma_t e^{-\sigma_t s}. \tag{2.30}$$

To properly determine s, the random variable ρ is set equal to the cumulative distribution function, as follows:

$$\begin{aligned} \rho &= \int_0^s \sigma_t e^{-\sigma_t x} dx \\ &= \frac{\sigma_t}{-\sigma_t} \left(e^{-\sigma_t s} - e^{-\sigma_t 0} \right) \\ &= 1 - e^{-\sigma_t s}. \end{aligned} \tag{2.31}$$

Since ρ is uniformly distributed $\in (0,1)$, $(1-\rho)$ has the same probability distribution as ρ. To simplify the procedure, the distance is sampled as

$$\rho = e^{-\sigma_t s}, \tag{2.32}$$

leading to

$$s = \frac{1}{-\sigma_t} \ln(\rho). \tag{2.33}$$

The next position is then given by

$$x' = x + s\mu. \tag{2.34}$$

If this value lies outside of the current cell, then the particle history is moved to the cell boundary and the tallies are performed. Assuming this cell boundary is not on the domain boundary, the tracking process loops back to determine a new distance to collision. Since the probability of interaction is independent of how far the particle has already traveled, the same process as before is used.

If the next position x' occurs within the cell, then an interaction occurs. Whether a scatter or absorption occurs is randomized by comparing another generated ρ with $\frac{\sigma_a}{\sigma_t}$. If ρ is less than this ratio, then the particle is absorbed and the history is terminated. Else, the particle scatters. At this stage only isotropic scatter is considered, but anisotropic scattering could be added. The code randomly determines the new direction and continues the particles history.

The entire algorithm is as follows:

initialize the tallies $\delta^a_{i,m}$, $\delta^x_{i,m}$, and $\delta^\mu_{i,m}$ to 0

FOR $j \in (1, N_p)$

- sample cell (i, m), position x, direction μ, and weight w
- BEGIN LOOP: particle
  - generate random number $\rho \in (0, 1)$
  - calculate distance to next collision $s \leftarrow -\frac{1}{\sigma_t} \ln(\rho)$
  - calculate next position $x' \leftarrow x + s\mu$
  - IF $x' \in (x_i - \frac{h_i}{2}, x_i + \frac{h_i}{2})$
    - [the particle interacted within the current cell]
  - ELSE
    - [the particle interacted outside of the current cell]
  - END
- END

END

For interactions within the current cell, the algorithm continues as follows:

[the particle interacted within the current cell]

calculate mean position $\bar{x} \leftarrow \frac{x+x'}{2}$

$\delta^a_{i,m} \leftarrow \delta^a_{i,m} + \frac{sw}{h_x h_m N_p}$

$\delta^x_{i,m} \leftarrow \delta^x_{i,m} + \frac{6}{h_x}\frac{(\bar{x}-x_i)sw}{h_x h_m N_p}$

$\delta^\mu_{i,m} \leftarrow \delta^\mu_{i,m} + \frac{6}{h_m}\frac{(\mu-\mu_m)sw}{h_x h_m N_p}$

generate random number $\rho \in (0,1)$

IF $\rho < \frac{\sigma_a}{\sigma_t}$

- the particle is absorbed, EXIT LOOP: particle

ELSE

- [the particle scattered]
- $x \leftarrow x'$
- generate random number $\rho \in (0,1)$
- calculate scattered direction isotropically $\mu \leftarrow 2\rho - 1$
- determine new cell (i,m)

END

For interactions outside of the current cell, the algorithm continues as follows:

[the particle interacted outside of the current cell]

determine final position x_b: $\leftarrow x_i - \frac{h_x}{2}$ IF $\mu < 0$

$\leftarrow x_i + \frac{h_x}{2}$ IF $\mu > 0$

calculate mean position $\bar{x} \leftarrow \frac{x+x_b}{2}$

calculate distance traveled $s \leftarrow \frac{x_b - x}{\mu}$

$\delta^a_{i,m} \leftarrow \delta^a_{i,m} + \frac{sw}{h_x h_m N_p}$

$\delta^x_{i,m} \leftarrow \delta^x_{i,m} + \frac{6}{h_x}\frac{(\bar{x}-x_i)sw}{h_x h_m N_p}$

$\delta^\mu_{i,m} \leftarrow \delta^\mu_{i,m} + \frac{6}{h_m}\frac{(\mu-\mu_m)sw}{h_x h_m N_p}$

IF x_b is not on the boundary

> [move the particle to the next cell]
>
> $x \leftarrow x_b$
>
> determine cell (i, m)

ELSE IF the boundary is reflective

> $\mu \leftarrow -\mu$
>
> $x \leftarrow x_b$
>
> determine cell (i, m)

ELSE

> the particle leaked, EXIT LOOP: particle

END

2.4 Adaptivity

2.4.1 General Concept

As previously discussed, the error in the ECMC solution will saturate at some level if the true solution does not exist in the linear-discontinuous trial space. The error level at which the method stagnates is directly proportional to the projection error, as the ECMC solution is the projection of the true solution onto the trial space. To reach further convergence the mesh must be made finer. Previous work shows that starting with a finer initial mesh does increase the convergence [17]. Adaptivity is implemented to automate the procedure, as well as to allow local refinement.

The refinement is executed by separating the rectangular cell into four equal-sized pieces. Since this adaptation removes the regularity of the mesh, the previous (i, m) cell denotation no longer applies. Thus, each cell is denoted by a single index i and must "remember" its location and dimensions. The quantities needed for cell i are the left and right boundary positions, the top and bottom boundary directions, the cell midpoint, and the widths in the position and direction. Note that not all of these are independent, and thus only half of them need to be stored. The rest are calculated on the fly.

Adding in adaptation also changes the algorithms needed to calculate the residual integral and the scalar flux, as well as the initial cell sampling and the methods to find the neighboring cell. These are performed in a recursive manner, starting with the parent cells on the initial mesh. In the case of the residual and scalar flux, the algorithm performs the calculations on the four daughter cells then sums it all together. If the cell hasn't been refined, the calculations are performed as described previously. For the sampling algorithm, the parent cell is first sampled. Then the code recursively samples in which of the daughter cells to initialize the particle.

Once a non-refined cell has been determined, the rejection method is performed as previously described.

In order to simplify the implementation of mesh adaptivity, a "single-level difference" is enforced: each cell can be no more than one refinement level away from each cell adjacent. Without this enforcment, two adjacent cells could have vastly different dimensions, and thus the projection onto the cells becomes highly discontinuous and irregular. This also allows for easier calculation of the delta terms in the residual, as these terms must account for the different refinement levels. If each cell is at most one level away, there are only three options for the calculations in Eq. (2.12). If both the current cell and the adjacent cell are at the same refinement level, then

$$r_i^{c1} = \begin{cases} \left(\psi_j^a + \psi_j^x\right) - \left(\psi_i^a - \psi_i^x\right) & \mu > 0, \\ \left(\psi_j^a - \psi_j^x\right) - \left(\psi_i^a + \psi_i^x\right) & \mu < 0, \end{cases} \tag{2.35a}$$

$$r_i^{c2} = \begin{cases} \psi_j^\mu - \psi_i^\mu & \mu > 0, \\ \psi_j^\mu - \psi_i^\mu & \mu < 0, \end{cases} \tag{2.35b}$$

where j represents the adjacent cell (left for positive μ, right for negative μ). If the adjacent cell is more refined than the current cell, then there are actually two cells adjacent. The equations become as follows:

$$r_i^{c1} = \begin{cases} \frac{1}{2}\left(\psi_j^a + \psi_j^x + \psi_k^a + \psi_k^x\right) - \left(\psi_i^a - \psi_i^x\right) & \mu > 0, \\ \frac{1}{2}\left(\psi_j^a - \psi_j^x + \psi_k^a - \psi_k^x\right) - \left(\psi_i^a + \psi_i^x\right) & \mu < 0, \end{cases} \tag{2.36a}$$

$$r_i^{c2} = \begin{cases} \frac{1}{4}\left(\psi_j^\mu + \psi_k^\mu + 3\left(\psi_j^a + \psi_j^x - \psi_k^a - \psi_k^x\right)\right) - \psi_i^\mu & \mu > 0, \\ \frac{1}{4}\left(\psi_j^\mu + \psi_k^\mu + 3\left(\psi_j^a - \psi_j^x - \psi_k^a + \psi_k^x\right)\right) - \psi_i^\mu & \mu < 0, \end{cases} \tag{2.36b}$$

where j represents the adjacent cell relatively on top, and k represents the cell

relatively on bottom. Again, for positive μ, j and k refer to the cells on the left, and they refer to the right adjacent cell for negative μ. If the current cell is more refined than the adjacent cell, there are two possibilities: the current cell is either relatively on the bottom or relatively on the top. If the current cell is relatively on the top,

$$r_i^{c1} = \begin{cases} \left(\psi_j^a + \psi_j^x + \frac{1}{2}\psi_j^\mu\right) - \left(\psi_i^a - \psi_i^x\right) & \mu > 0, \\ \left(\psi_j^a - \psi_j^x + \frac{1}{2}\psi_j^\mu\right) - \left(\psi_i^a + \psi_i^x\right) & \mu < 0, \end{cases} \tag{2.37a}$$

$$r_i^{c2} = \begin{cases} \frac{1}{2}\psi_j^\mu - \psi_i^\mu & \mu > 0, \\ \frac{1}{2}\psi_j^\mu - \psi_i^\mu & \mu < 0. \end{cases} \tag{2.37b}$$

Whereas, if the current cell is relatively on bottom,

$$r_i^{c1} = \begin{cases} \left(\psi_j^a + \psi_j^x - \frac{1}{2}\psi_j^\mu\right) - \left(\psi_i^a - \psi_i^x\right) & \mu > 0, \\ \left(\psi_j^a - \psi_j^x - \frac{1}{2}\psi_j^\mu\right) - \left(\psi_i^a + \psi_i^x\right) & \mu < 0, \end{cases} \tag{2.38a}$$

$$r_i^{c2} = \begin{cases} \frac{1}{2}\psi_j^\mu - \psi_i^\mu & \mu > 0, \\ \frac{1}{2}\psi_j^\mu - \psi_i^\mu & \mu < 0. \end{cases} \tag{2.38b}$$

2.4.2 Detecting Convergence Stagnation

Convergence stagnation is generally detected based on the L^2 norm of the residual. In each batch, the magnitude of the norm is compared with magnitude of the norm calculated in the previous batch, as

$$\alpha = \ln\left(\frac{||\tilde{r}^{(i-1)}||_{L^2}}{||\tilde{r}^{(i)}||_{L^2}}\right). \tag{2.39}$$

Since ECMC should exhibit exponential convergence, this α should remain roughly constant and represent the rate of convergence. Thus,

$$||\tilde{r}^{(i)}||_{L^2} = Ce^{-\alpha i}, \tag{2.40a}$$

$$-\alpha i = \ln\left(||\tilde{r}^{(i)}||_{L^2}\right) - \ln C, \tag{2.40b}$$

$$\begin{aligned} \alpha &= -\alpha(i-1) + \alpha i \\ &= \ln\left(||\tilde{r}^{(i-1)}||_{L^2}\right) - \ln C - \ln\left(||\tilde{r}^{(i)}||_{L^2}\right) + \ln C \\ &= \ln\left(\frac{||\tilde{r}^{(i-1)}||_{L^2}}{||\tilde{r}^{(i)}||_{L^2}}\right). \end{aligned} \tag{2.40c}$$

If this value falls below a preset tolerance or goes negative, then the solution has stopped converging exponentially. To avoid any "hiccups" or "blips" induced by the stochastic nature of the Monte Carlo method, the average of the the past three ratios is compared with the tolerance. These ratios are initialized to 1 at the beginning and after each adaptation occurs.

2.4.3 Determining Refined Cells

The refined cells are determined based on the discontinuities in the estimated solution at the cell interfaces [18]. There are four of these "jump error indicators" per cell, and each one has a different formulation based on the relative refinement levels. There are thus twelve equations total, fully specified in Appendix B. For an example, this is the equation for the left boundary when both cells have the same refinement level:

$$\xi_i^{left} = \int_{\mu_i^B}^{\mu_i^T} \left| \left(\psi_j^a + \psi_j^x + \psi_j^\mu \frac{2}{h_j^\mu}(\mu - \mu_j^C)\right) - \left(\psi_i^a - \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C)\right) \right| d\mu, \tag{2.41}$$

where j represents the cell to the left, μ_i^B is the bottom direction boundary for cell i, μ_i^T is the top direction boundary for cell i, and μ_i^C is the cell center direction for cell i. It should be noted that in this instance, $h_j^\mu = h_i^\mu$ and $\mu_j^C = \mu_i^C$. A prescribed percent of the total number of current cells is refined based on the maximum jump error in the cells. Once these cells are refined, the code refines additional cells as required such that the refinement level is always one between adjacent cells.

2.5 Biased Sampling

2.5.1 General Concept

To ensure enough particles reach the smaller cells created during refinement, a biased sampling method is employed. Instead of sampling the starting cell for the particle based on the magnitude of the residual, an equal number of particles is generated in each cell. As with the non-biased method, the particle is still randomly started on either a face or the interior of the cell, and the position and direction are still sampled from the cell-based distribution. To maintain the correct solution the weights of the particles are altered to reflect the relative magnitudes of the residual.

Consider the true probability distribution P with associated weighting w. The biased probability and weighting are defined as P^* and w^*, respectivley. In order for these to yield the same solution, the two distributions must be related by

$$Pw = P^* w^*. \tag{2.42}$$

In this case, P is the real distribution based on the residual. Thus w is uniformly 1. Since the biased sampling generates an equal number of particles in each cell, P^* is uniformly $\frac{1}{N_C}$, where N_C is the number of cells. The particle weights for the biased

sampling are thus

$$w^* = PN_C. \tag{2.43}$$

2.5.2 Sampling Algorithm

The biased sampling algorithm is similar to the normal algorithm. The main difference is the starting cell is set instead of randomized. Given the cell, face or linear is determined as before and the rejection method is applied. The weight is then set as described before. Namely, for cell i the weight is

$$w_i^* = \frac{I_i^{int} + I_i^{fac}}{||\tilde{r}||_{L^1}} N_C. \tag{2.44}$$

As before, this value is set negative if the residual evaluates negative at the sampled position and direction. Very little changes in the Monte Carlo algorithm. Instead of running through all the particles and randomizing the cells, the code loops through every cell and runs a specified number of particles starting in that cell. Thus the total number of particle histories N_p needs to be set as a multiple of the number of cells N_C. The algorithm appears as follows:

initialize the tallies δ_i^a, δ_i^x, and δ_i^μ to 0
FOR $i \in (1, N_C)$
  FOR $j \in (1, \frac{N_p}{N_C})$
    given cell i, sample position x, direction μ, and weight w^*
    BEGIN LOOP: particle
      ...
    END
  END
END

3. TEST PROBLEM DESCRIPTIONS

3.1 Constant Solution

If the angular flux is constant in space and angle, then the following equations hold:

$$\frac{\partial \psi}{\partial x} = 0, \tag{3.1a}$$

$$\phi = 4\pi\psi. \tag{3.1b}$$

Substituting these into Eq. (2.1) yields

$$\sigma_t \psi = \sigma_s \psi + Q,$$

$$\psi = \frac{Q}{\sigma_a}. \tag{3.2}$$

Thus Q must have the same functional form as σ_a. In this case, they are both constant. To have a constant solution throughout the domain, the incoming boundary fluxes are set equal to this value. Since this solution exists in the disconinuous-linear trial space, exponential convergence should be achieved until the ECMC solution is exact to round-off. This is true for any starting mesh and without any adaptation. As such, this problem is intended to verify the algorithms and research code.

3.2 Manufactured Solution

Consider the solution of the form

$$\psi = C(1+\mu^2)x(T-x)e^{-(x-x_0)^2} = Cf(x)g(\mu), \tag{3.3}$$

where C is a constant chosen to normalize the source to 1. This solution has no singularities, is always non-negative, is continuous and smooth, and is zero at the boundaries. Given this form, the following are true:

$$\begin{aligned}\frac{\partial\psi}{\partial x} &= Cg(\mu)\frac{\partial f(x)}{\partial x} \\ &= Cg(\mu)e^{-(x-x_0)^2}\left(T-2x-2x(T-x)(x-x_0)\right) \\ &= Cf(x)g(\mu)\left(\frac{T-2x}{x(T-x)}-2(x-x_0)\right),\end{aligned} \tag{3.4}$$

$$\phi = 2\pi\int_{-1}^{+1}\psi\,d\mu = 2\pi Cf(x)\int_{-1}^{+1}(1+\mu^2)\,d\mu = \frac{16\pi}{3}Cf(x). \tag{3.5}$$

Thus,

$$\begin{aligned}Q &= \mu\frac{\partial\psi}{\partial x}+\sigma_t\psi-\frac{\sigma_s}{4\pi}\phi \\ &= \mu Cf(x)g(\mu)\left(\frac{T-2x}{x(T-x)}-2(x-x_0)\right)+\sigma_t Cf(x)g(\mu)-\frac{\sigma_s}{4\pi}\frac{16\pi}{3}Cf(x) \\ &= Cf(x)\left(\mu g(\mu)\left(\frac{T-2x}{x(T-x)}-2(x-x_0)\right)+\sigma_t g(\mu)-\frac{4}{3}\sigma_s\right).\end{aligned} \tag{3.6}$$

Given that the exact solution is zero at the boundaries,

$$\psi^-_{inc} = \psi^+_{inc} = 0. \tag{3.7}$$

To normalize the solution, C is chosen such that the integral of Q is 1:

$$\begin{aligned}
1 &= 2\pi \int_{-1}^{+1} \int_{0}^{T} Q \, dx \, d\mu \\
&= 2\pi \int_{-1}^{+1} \int_{0}^{T} C f(x) \left(\mu g(\mu) \left(\frac{T-2x}{x(T-x)} - 2(x-x_0) \right) + \sigma_t g(\mu) - \frac{4}{3}\sigma_s \right) dx \, d\mu \\
&= C2\pi \int_{0}^{T} f(x) \left(\int_{-1}^{+1} (\mu g(\mu)) \, d\mu \left(\frac{T-2x}{x(T-x)} - 2(x-x_0) \right) + \sigma_t \int_{-1}^{+1} g(\mu) \, d\mu - \frac{4}{3}\sigma_s 2 \right) dx \\
&= C2\pi \int_{0}^{T} f(x) \left(\frac{8}{3}\sigma_t - \frac{8}{3}\sigma_s \right) dx = C\frac{16\pi}{3}\sigma_a \int_{0}^{T} f(x) \, dx.
\end{aligned} \tag{3.8}$$

Let I be the integral of f, as follows:

$$\begin{aligned}
I &= \int_{0}^{T} x(T-x) e^{-(x-x_0)^2} \, dx \\
&= \frac{1}{2}(x_0 - T + x) e^{-(x-x_0)^2} \Big[_{0}^{T} - \frac{\sqrt{\pi}}{4}(2x_0^2 - 2x_0 T + 1)\mathrm{erf}(x-x_0) \Big[_{0}^{T} \\
&= \frac{1}{2} \left(x_0 e^{-(T-x_0)^2} + (T-x_0) e^{-x_0^2} \right) - \frac{\sqrt{\pi}}{4}(2x_0^2 - 2x_0 T + 1) \left(\mathrm{erf}(T-x_0) + \mathrm{erf}(x_0) \right).
\end{aligned} \tag{3.9}$$

Thus,

$$C = \frac{3}{16\pi\sigma_a I}. \tag{3.10}$$

Since this solution exists outside of the trial space, the algorithm is expected to stagnate at some error. At this point, the adaptivity algorithms will refine the mesh and continue reducing the error. Due to the potential of small cells, the biased algorithm is utilized.

3.3 Pure Attenuation

Consider the transport equation with no scattering, no internal source, and an isotropic incoming flux:

$$\mu\frac{\partial\psi}{\partial x}+\sigma_a\psi=0, \qquad x\in[0,T],\ \mu\in[-1,+1], \tag{3.11a}$$

$$\psi^+_{inc}=\frac{1}{2\pi}, \tag{3.11b}$$

$$\psi^-_{inc}=0. \tag{3.11c}$$

The exact solution for this is

$$\psi=\begin{cases}\frac{1}{2\pi}e^{-\frac{\sigma_a x}{\mu}} & \mu>0,\\ 0 & \mu\le 0,\end{cases} \tag{3.12}$$

$$\phi=\int_0^1 e^{-\frac{\sigma_a x}{\mu}}d\mu=\int_1^\infty\frac{e^{-\sigma_a xz}}{z^2}dz=\mathrm{E}_2(\sigma_a x). \tag{3.13}$$

As with the manufactured problem, the solution lies outside the trial space. This means the adaptivity algorithms must be utilized. Note the singularity as $\mu\to 0^+$. In order to sufficiently resolve the mesh around $\mu=0$ the cell dimensions must be quite small. Thus the biased sampling must be used to control the error.

3.4 Internal Source with Scattering

Consider the full transport equation with zero incoming flux.

$$\mu\frac{\partial\psi}{\partial x}+\sigma_t\psi=\frac{\sigma_s}{4\pi}\phi+Q,\qquad x\in[0,T],\ \mu\in[-1,+1], \tag{3.14a}$$

$$\psi^+_{inc}=\psi^-_{inc}=0. \tag{3.14b}$$

The exact solution will not lie within the trial space, meaning the adaptivity algorithms must be utilized. Due to the potential of small cells, the biased algorithm is utilized. Since the exact solution for this equation is not known, the results were visually compared with an S_n transport code solution.

The relative L^2 norm of the actual error in the scalar flux is used as the error measure when the exact solution is known. That is,

$$\begin{aligned}
\epsilon^{(i)} &= \frac{||\phi^e - \tilde{\phi}^{(i)}||}{||\phi^e||} \\
&= \frac{\left(\int_{x_{1/2}}^{x_{N+1/2}} \left(\phi^e(x) - \tilde{\phi}^{(i)}(x)\right)^2 \, dx\right)^{\frac{1}{2}}}{\left(\int_{x_{1/2}}^{x_{N+1/2}} \left(\phi^e(x)\right)^2 \, dx\right)^{\frac{1}{2}}} \\
&= \frac{\left(\sum\limits_{j\in\{cells\}} \int_{x_j^L}^{x_j^R} \left(\phi^e(x) - \phi_j^{(i)}(x)\right)^2 \, dx\right)^{\frac{1}{2}}}{\left(\sum\limits_{j\in\{cells\}} \int_{x_j^L}^{x_j^R} \left(\phi^e(x)\right)^2 \, dx\right)^{\frac{1}{2}}},
\end{aligned} \tag{4.1}$$

where ϕ^e is the exact solution, and $\tilde{\phi}^{(i)}$ is the ECMC generated approximation in batch i. When the exact solution is not known, the error measure used is the L^2 norm of the residual normalized by the L^2 norm of the source. This is as follows:

$$\begin{aligned}
\epsilon^{(i)} &= \frac{||r^{(i)}||}{||Q||} = \frac{||\mathbf{L}[\psi^e - \tilde{\psi}^{(i)}]||}{||\mathbf{L}[\psi^e]||} \\
&= \frac{\left(\int_{x_{1/2}}^{x_{N+1/2}} 2\pi \int_{-1}^{+1} \left(r^{(i)}(x,\mu)\right)^2 \, d\mu dx\right)^{\frac{1}{2}}}{\left(\int_{x_{1/2}}^{x_{N+1/2}} 2\pi \int_{-1}^{+1} \left(Q(x,\mu)\right)^2 \, d\mu dx\right)^{\frac{1}{2}}} \\
&= \frac{\left(\sum\limits_{j\in\{cells\}} 2\pi \int_{x_j^L}^{x_j^R} \int_{\mu_j^B}^{\mu_j^T} \left(r^{(i)}(x,\mu)\right)^2 \, d\mu dx\right)^{\frac{1}{2}}}{\left(\sum\limits_{j\in\{cells\}} 2\pi \int_{x_j^L}^{x_j^R} \int_{\mu_j^B}^{\mu_j^T} \left(Q(x,\mu)\right)^2 \, d\mu dx\right)^{\frac{1}{2}}},
\end{aligned} \tag{4.2}$$

where Q also includes the incoming flux. These integrals are calculated using a two-dimensional Gaussian quadrature rule.

For all problems the domain thickness was set at 3 cm, and the total interaction cross section was set at 1 cm. Thus the domain was 3 mean-free-paths thick. The starting grid was 100 cells, 10 each for position and direction. The total source is always normalized to 1 $\frac{n}{cm^3-ster-s}$.

4.1 Constant Solution

The error as a function of the number of batches is given in Fig. 4.1, with NP being the number of histories run per cell in each batch.

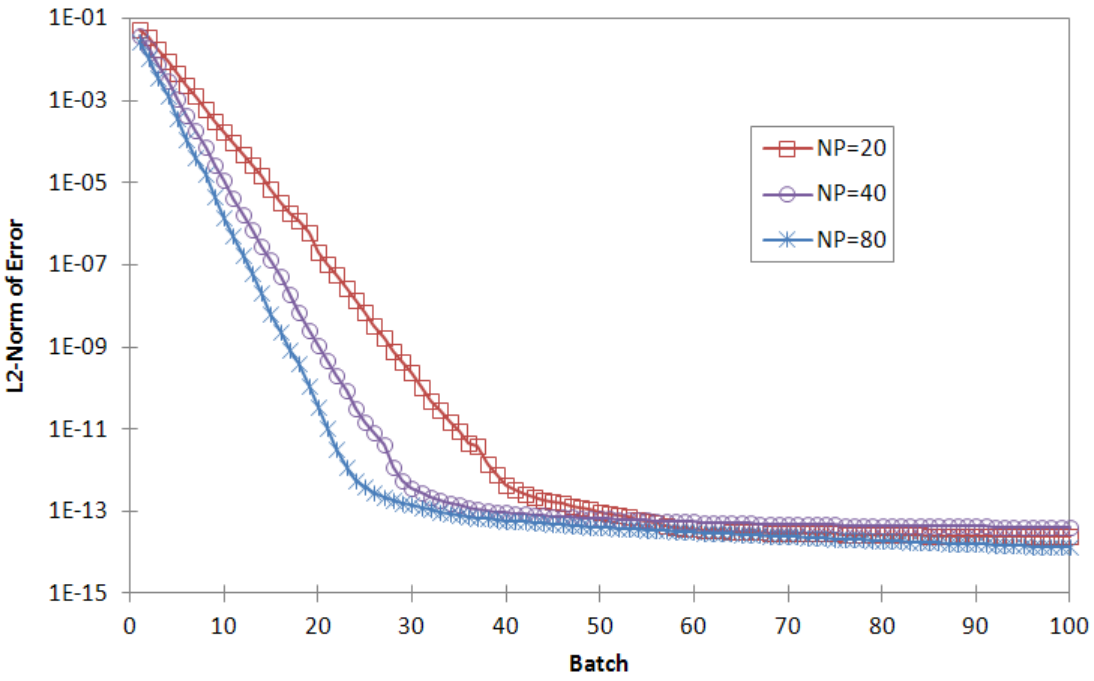


Figure 4.1: Convergence for Constant Solution Problem.

The results and observations from this problem are similar to the results obtained

previously [17]. There are several important characteristics to be noted from Fig. 4.1. The first is that the error decreases linearly on a semi-log plot. Thus exponential convergence is achieved. The second is that exponential convergence is achieved until the error reaches the tolerance (10^{-12}) and then saturates. This is expected because the solution lies within the trial space and it strongly indicates that the algorithms and research code are valid.

The third is that the convergence rate increases as the number of histories per batch is increased. This is a very important property. Exponential convergence will only be achieved if a sufficient number of histories are run per batch, i.e., if the error per batch is sufficiently reduced. This makes a very important point about the potential applicablity of the ECMC method. This approach will not make problems that are extremely difficult easier, but it has the potential to reduce the statistical error associated with tenable problems to negligible levels. In addition, this explains why highly oscillatory global polynomial trial spaces can be a very poor choice. The highly oscillatory nature of the trial space functions yields large and nearly equal positive and negative contributions to the expansion coefficients, resulting in excessive statistical errors that ultimately cause immediate error saturation or even divergence.

Note from Fig. 4.1 that the error for the NP=80 case is reduced from roughly one to about 10^{-12} after 25 batches. With standard Monte Carlo, the error reduction after 25 batches would be be expected to be about $1/\sqrt{25 * 80} \approx 0.022$, and 10^{22} batches would be required to achieve the same error reduction as the ECMC method. In this case, the improvement in efficiency with the ECMC algorithm relative to the standard algorithm is astonishing. In general, the efficiency realized for any particular problem will be a trade-off between the extra cost associated with the estimation of the full solution at all points in the problems and the rapid reduction in error achieved

with an ECMC algorithm. ECMC algorithms are clearly most promising when the solution is required throughout the problem and the statistical error must be reduced to negligible levels.

4.2 Manufactured Solution

For the manufactured solution problem the scattering ratio was set to 0.05. The error as a function of the number of batches is given in Fig. 4.2, for both the problem executed with adaptivity and without.

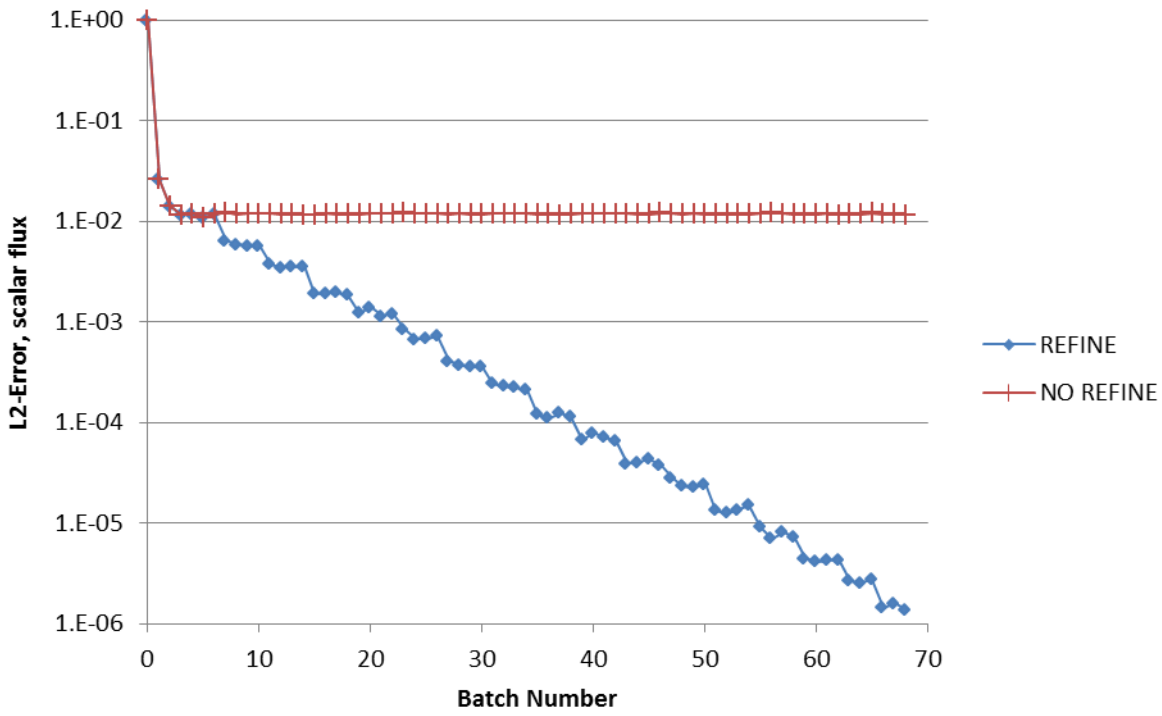


Figure 4.2: Convergence for Manufactured Solution Problem.

As discussed previously, the solution exists outside of the trial space. Thus without refinement the error saturates. When the adaptivity algorithm is used, the code

detects this saturation and refines the mesh. For this problem 30% of the cells were refined at each adaptivity step. The meshes are given in Appendix C.

The refinement did indeed maintain the convergence of the problem. However, exponential convergence is not observed between the refinement steps. This is due to the relatively small gain in accuracy with the finer mesh. The decreases in error before the first saturation exhibit an exponential quality, and these jumps are much larger than the differences in accuracy between refinement levels. The method does converge exponentially, it is just hidden in the error saturation.

Displayed in Fig. 4.3 is the time associated with running a batch as a function of the number of particles in that batch.

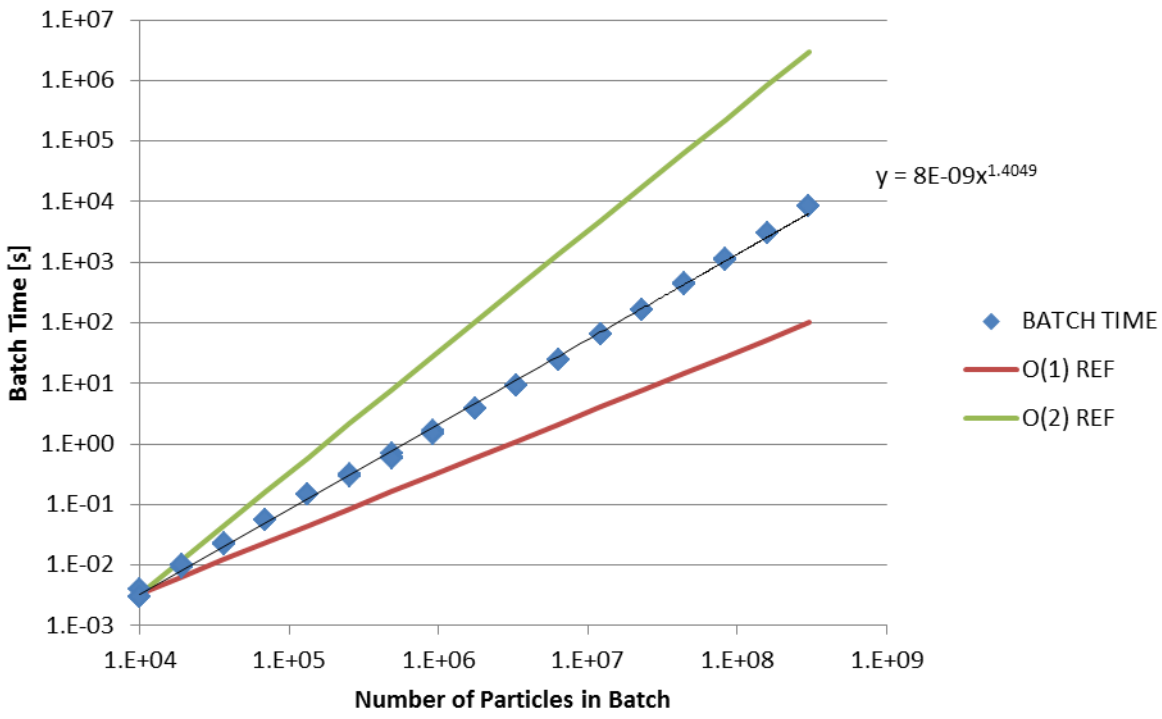


Figure 4.3: Batch Time for Manufactured Solution Problem.

Without any mesh refinement the time per batch would be directly proportional to the number of particles. However, it takes longer to run a single history with a finer mesh. Thus, the time in a given batch should increase at a rate higher than the increase in the number of particles. As shown in Fig. 4.3, this problem exhibits order 1.4 time increase.

This same problem was run with a normal Monte Carlo script. This code used the mesh from the ECMC code corresponding to a scalar flux error of 10^{-4}. The error in the scalar flux from the Monte Carlo code as a function of number of particles is given in Table 4.1. Also given is the constant for the data to represent $1/\sqrt{N}$ convergence, and the resulting number of particles required to reach an error of 10^{-4}.

Table 4.1: Regular Monte Carlo on Manufactured Solution

N_P	L^2 error, scalar flux	Fit constant	N_P to converge
3.35E+06	3.4423E-03	6.30	3.96E+09
1.34E+07	1.5964E-03	5.84	3.41E+09
2.01E+07	1.3588E-03	6.09	3.71E+09
2.68E+07	1.1726E-03	6.07	3.68E+09
3.35E+07	1.1425E-03	6.61	4.37E+09
4.35E+07	9.0087E-04	5.94	3.53E+09

As shown in the table, about 3.8 x 10^9 total histories would need to be run to reduce the error to 10^{-4}. The ECMC algorithm only ran 1.8 x 10^7 particles, including the "wasted" batches between stagnation and refinement. This demonstrates the efficiency of the algorithm over standard Monte Carlo.

4.3 Pure Attenuation

The error in the attenuation problem as a function of the number of batches is given in Fig. 4.4, for both the problem ran with biased sampling and without.

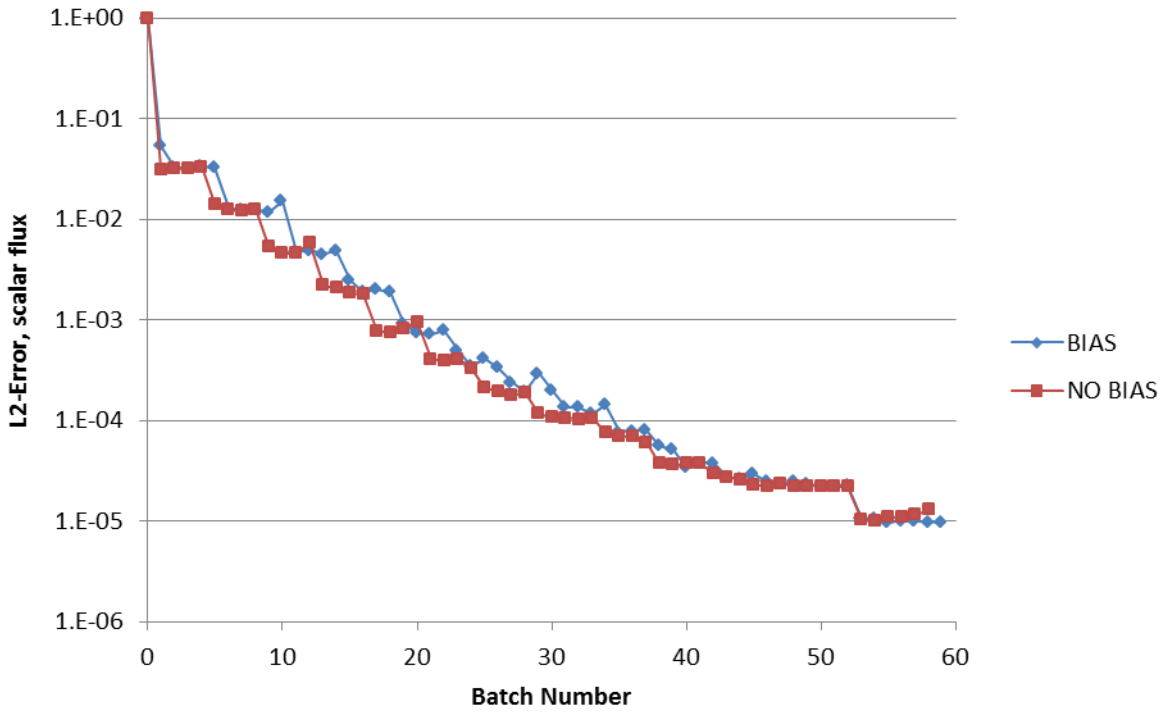


Figure 4.4: Convergence for Attenuation Solution Problem - Scalar Flux.

When the problem is ran without biased sampling, the error blows up. The effect is easier to see when looking at the L^2 error of the angular flux, as given in Fig. 4.5.

This is due to relatively few particles reaching the smaller cells near the boundary. While these cells have the largest residual magnitudes, the areas are very small. This means the sampling algorithm is less likely to sample particles in those regions. With the biased sampling, particles are guaranteed to be generated in those areas. The

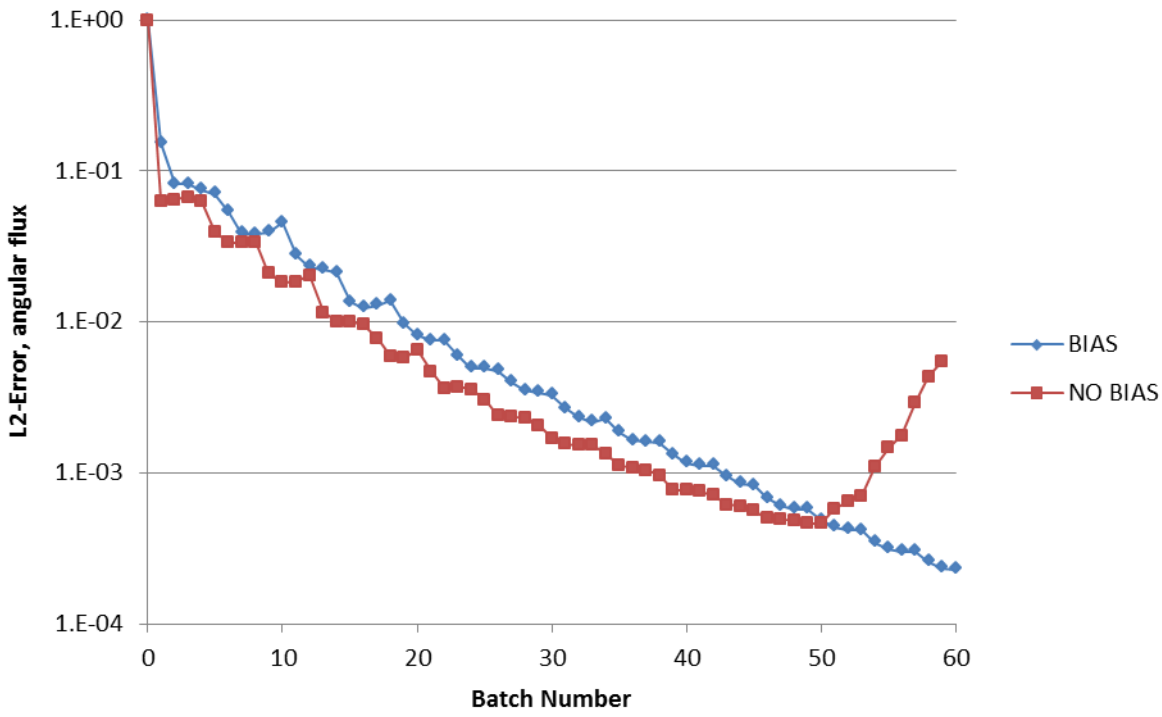


Figure 4.5: Convergence for Attenuation Solution Problem - Angular Flux.

adapted meshes are given in Appendix D.

One interesting thing to note is how the problem ran with the biased samplin ghas consistently higher error and adapts at later batches than the problem ran without biased sampling . This is due to the relatively fewer particles in the biased sampling that have meaningful weight. Consider the first batch. Both methods ran 100 particles per cell, starting with a 10x10 cell grid. There is no internal source, and only a surface source on the left boundary. Thus only five cells contain a source. Without the biased sampling, all 10,000 histories are sampled on the far left edge. With the biased sampling, only the 500 particles that start in these five edge cells have any weight; the other 9,500 have a zero weight and no contribution to the tallies. In subsequent batches a similar issue occurs with the cells with a negative

direction. The particles that are generated below the midline have no contribution to the tallies. Thus, in this specific instance, the biased sampling method has a higher error and thus lags behind the real-sampling. That is, until the real sampling error blows up and the biased sampling continues converging.

Fig. 4.6 shows the scalar flux solution at various batch steps for this problem.

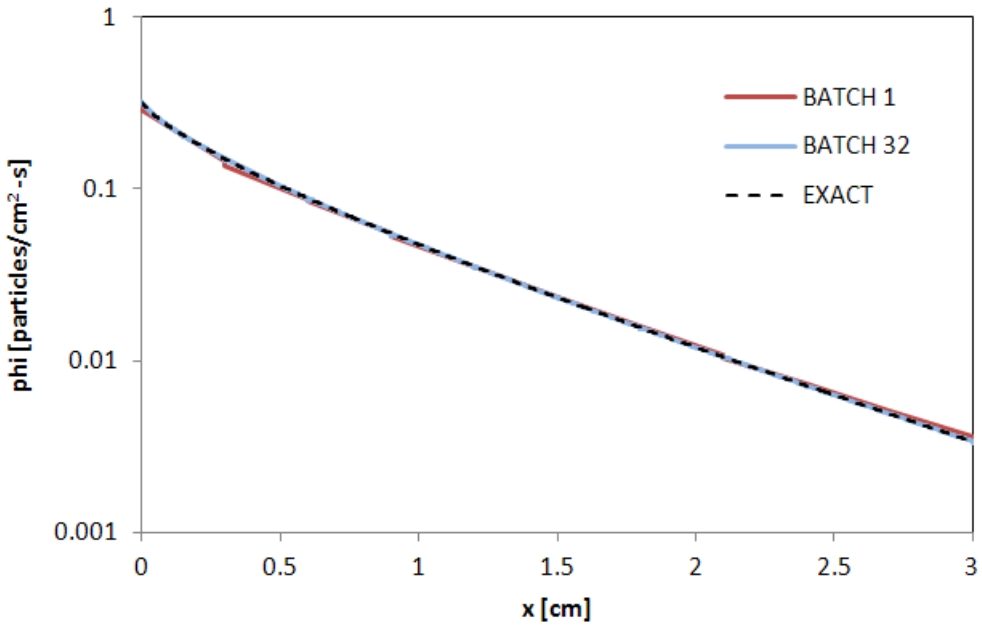


Figure 4.6: Scalar Flux for Attenuation Solution Problem.

After a single batch the error in the ECMC result is quite obvious. However, after 32 batches the human eye can discern virtually no error. This verifies the code is indeed converging to the correct solution.

Displayed in Fig. 4.7 is the time associated with running a batch as a function of the number of particles in that batch.

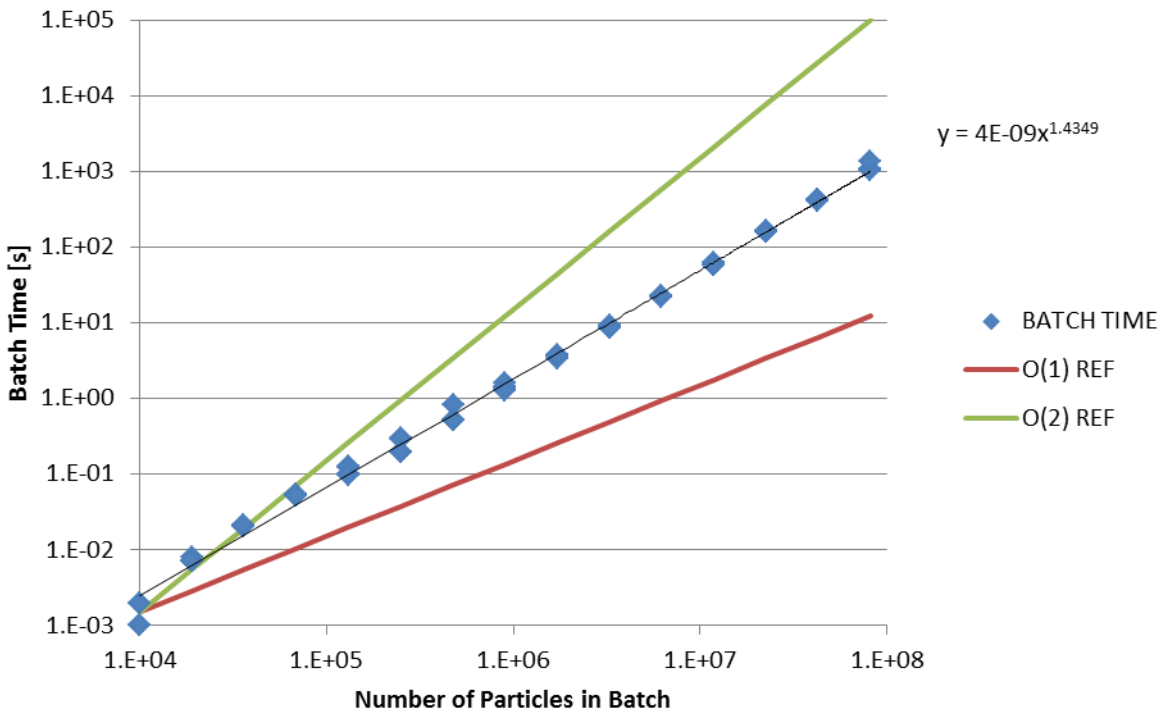


Figure 4.7: Batch Time for Attenuation Solution Problem.

As previously stated, the time in a given batch should increase at a rate higher than the increase in the number of particles. As shown in Fig. 4.7, this problem exhibits order 1.43 time increase.

Again, this same problem was run with a normal Monte Carlo script to a scalar flux error of 10^{-4}. The values are given in Table 4.2.

Table 4.2: Regular Monte Carlo on Attenuation Solution

N_P	L^2 error, scalar flux	Fit constant	N_P to converge
1.73E+06	1.7193E-02	22.5	5.10E+10
3.45E+06	1.1868E-02	22.0	4.86E+10
6.90E+06	7.2870E-03	19.1	3.66E+10
1.38E+07	5.3572E-03	19.9	3.96E+10
2.59E+07	3.4646E-03	17.6	3.11E+10

For the attenuation problem, about 4.1 x 10^{10} total histories would need to be run to reduce the error to 10^{-4}. The ECMC algorithm only ran 9.3 x 10^{6} particles, including the "wasted" batches between stagnation and refinement. This again demonstrates the efficiency of the algorithm over standard Monte Carlo.

4.4 Internal Source with Scattering

For the scattering problem, the scattering ratio was set to 0.75. The residual as a function of the number of batches is given in Fig. 4.8. Displayed are both the L^1 norm, which is simply the absolute integral of the residual, and the L^2 norm, as discussed earlier in this section. The adapted meshes are given in Appendix E.

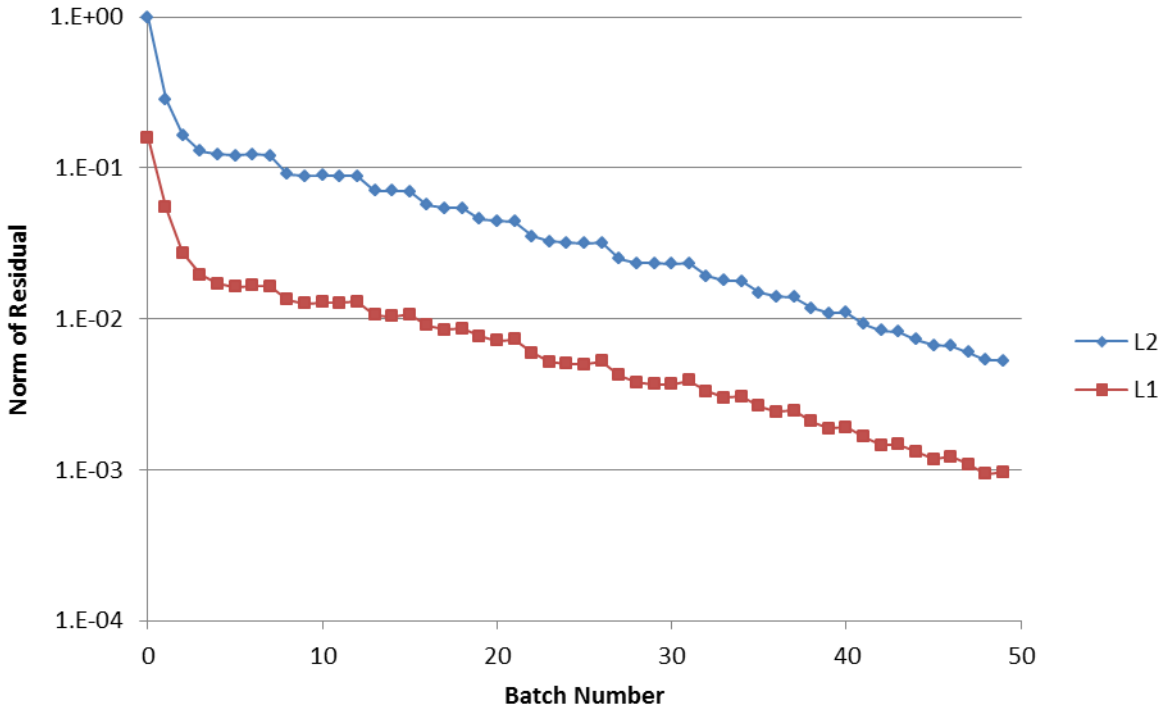


Figure 4.8: Convergence for Internal Scattering Problem.

This problem exhibits the same behavior as the previous ones. One thing to note is the residual norm is normally an order of magnitude or more higher than the scalar flux in the previous problems. Thus the apparent slow convergence and higher errors.

Displayed in Fig. 4.9 is the time associated with running a batch as a function of the number of particles in that batch.

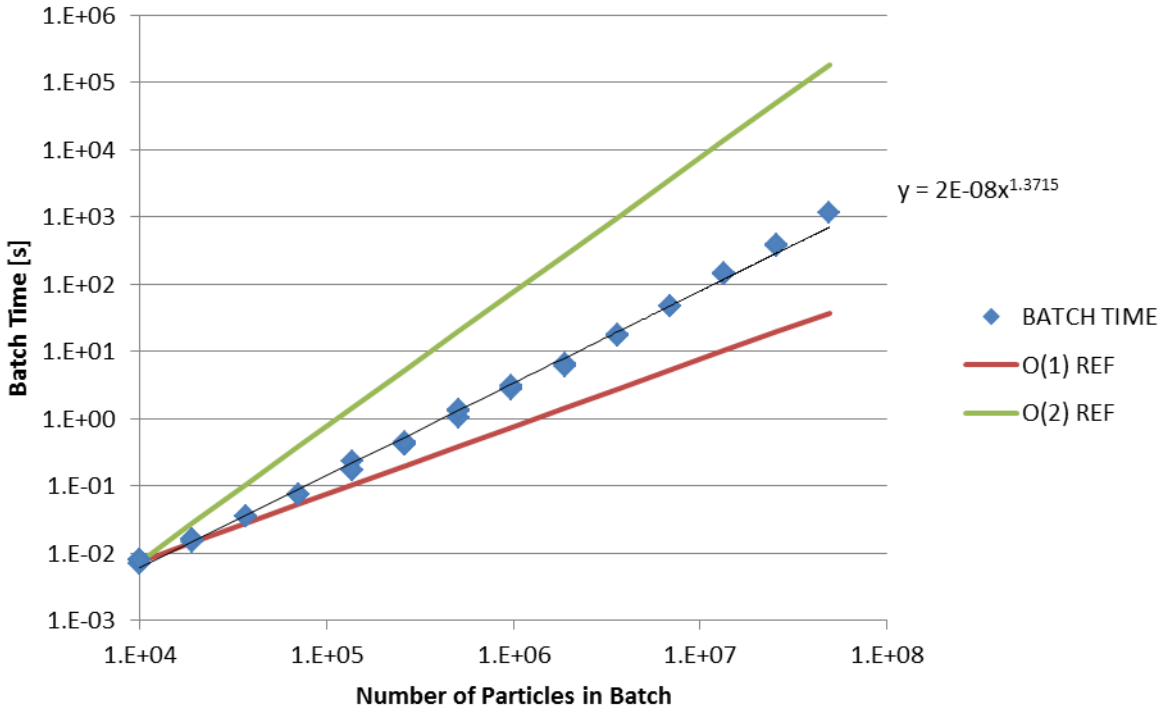


Figure 4.9: Batch Time for Internal Scattering Problem.

As previously stated, the time in a given batch should increase at a rate higher than the increase in the number of particles. As shown in Fig. 4.7, this problem exhibits order 1.37 time increase.

As the exact solution to this problem is not known, the ECMC solution was compared to an S_n solution. Fig. 4.10 compares the scalar flux at different refinement levels.

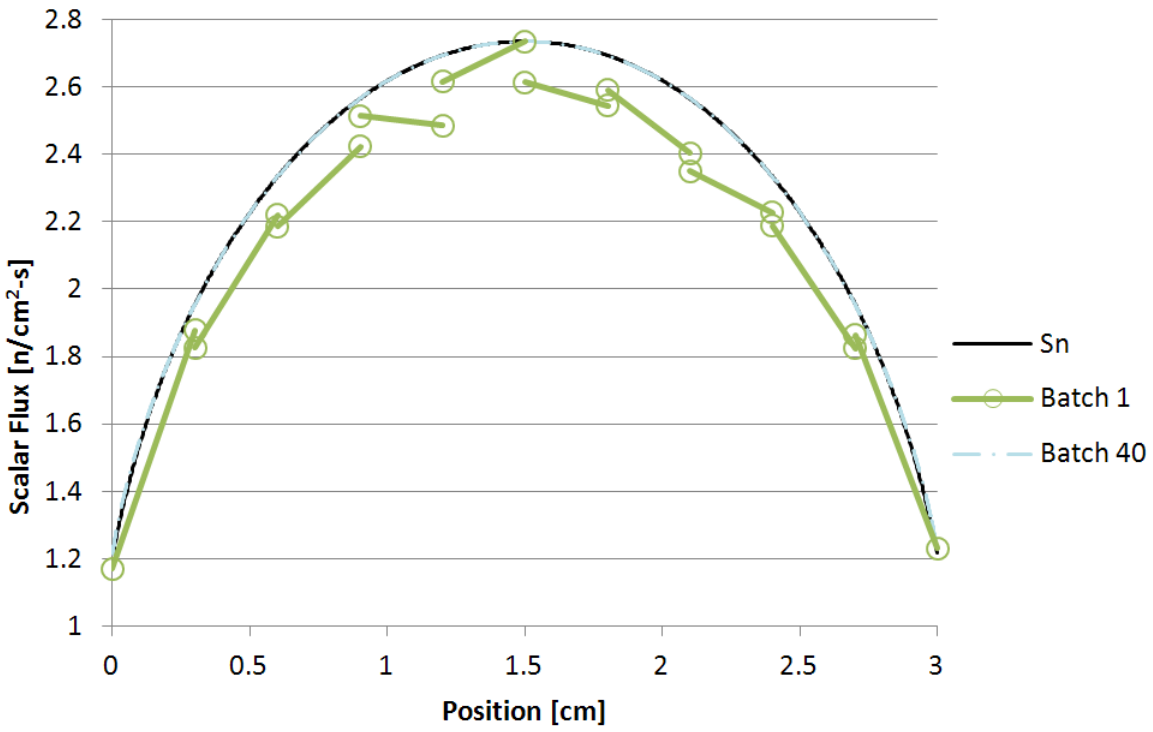


Figure 4.10: Scalar Flux for Internal Scattering Problem.

The solution closely matches the deterministic code after 40 batches. This verifies that our code works in high scattering environments and when the exact solution isn't known.

5. CONCLUDING REMARKS

The ECMC algorithms were studied utilizing finite-element trial spaces. The algorithms were applied to the one-dimension, one-speed, slab transport equation and utilized adaptive mesh algorithms and biased sampling. The results are encouraging and clearly indicate that further research relating to this approach is warranted. A next step would be to add further complexity to the equation studied. Anisotropic scattering would be a relatively simple next step, as the direction is already discretized. Energy could also be added to the equation, in the form of a multi-group discretization. Fission could also be taken into consideration. Future work should also extend the work into multiple dimensions, allowing for more realistic systems to be taken into account.

REFERENCES

[1] Nicholas Metropolis. "The Beginning of the Monte Carlo Method". *Los Alamo Science (Special Issue dedicated to Stanislaw Ulam)*, pages 125–130, 1987.

[2] Brian C. Franke, Ronald P. Kensek, Thomas W. Laub. *ITS Version 5.0: The Integrated TIGER Series of Coupled Electron/Photon Monte Carlo Transport Codes with CAD Geometry.* SAND2004-5172, Sandia National Laboratories, 2005.

[3] D.E. Cullen. *TART2005-A Coupled Neutron-Photon 3-D, Combinatorial Geometry, Time Dependent Monte Carlo Transport Code.* UCRL-SM-218009, Lawrence Livermore National Laboratory, 2005.

[4] B. L. Kirk. "Overview of Monte Carlo radiation transport codes". *Radiation Measurements*, 45(10):1318–1322, 2010.

[5] R.N. Blomquist. *VIM Monte Carlo Neutron/Photon Transport Code Users Guide Version 5.1.* Argonne National Laboratory, 2009.

[6] O. Petit, F.-X. Hugot, Y.-K. Lee, C. Jouanne, A. Mazzolo. *TRIPOLI-4 Version 4 User Guide.* Rapport CEA-R-6169, Commissariat à l'énergie atomique, CEA/SACLAY, Cedex, France, 2008.

[7] A.V. Dementyev, N.M. Sobolevsky. "SHIELD - A Monte Carlo hadron transport code". In *Proc. of a Specialists Meeting "Intermediate Energy Nuclear Data: Models and Codes"*, page 237. Issy-les-Moulineaux, France, 1994.

[8] S.D. Richards, C.M.J. Baker, A.J. Bird, P. Cowan, N. Davies, G.P. Dobson, T.C. Fry, A. Kyrieleis., P.N. Smith. "MONK and MCBEND: Current Status and

Recent Developments". In *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo*. Paris, France, 2013.

[9] Geant4 Collaboration. *Introduction to Geant4*. 2006. `http://geant4.web.cern.ch/geant4/UserDocumentation/Welcome/IntroductionToGeant4/html/index.html`.

[10] Paul K. Romano and Benoit Forget. "The OpenMC Monte Carlo Particle Transport Code". *Annals of Nuclear Energy*, 51:274–281, 2013.

[11] X-5 Monte Carlo Team. *MCNP - A General Monte Carlo N-Particle Transport Code, Version 5. Volume I: Overview and Theory*. Los Alamos National Laboratory, 2003.

[12] R. Kong and J. Spanier. "A New Proof of Geometric Convergence for General Transport Problems based on Sequential Correlated Sampling Methods". *Journal of Computational Physics*, 227:9762–9777, 2008.

[13] T.M. Evans, T.J. Urbatch, H. Lichtenstein, J.E. Morel. "A Residual Monte Carlo Method for Discrete Thermal Radiation Diffusion". *Journal of Computational Physics*, 189:539–556, 2003.

[14] T. Booth. "Exponential Convergence on a Continuous Monte Carlo Transport Problem". *Nuclear Science and Engineering*, 127:338–345, 1997.

[15] H. Lichtenstein. "Exponential Convergence Rates for Reduced-Source Monte Carlo Transport in [X,Y] Geometry". *Nuclear Science and Engineering*, 133:258–268, 1999.

[16] Jeffrey A. Favorite and Henry Lichtenstein. "Exponential Monte Carlo Convergence of a Three-Dimensional Discrete Ordinates Solution". *Transactions of the American Nuclear Society*, 81:147–148, 1999.

[17] Jim E. Morel, Jared P. Tooley, Brandon J. Blamer. "Exponentially-Convergent Monte Carlo via Finite-Element Trial Spaces". In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. Rio de Janeiro, RJ, Brazil, May 8-12 2011.

[18] Yaqi Wang and Jean C. Ragusa. "Standard and Goal-Oriented Adaptive Mesh Refinement Applied to Radiation Transport on 2D Unstructured Triangular Meshes". *Journal of Computational Physics*, 230:763–788, 2011.

APPENDIX A

RESIDUAL INTEGRATION

The residual is defined in Eq. (2.11) and Eq. (2.12). The interior integral is defined as

$$\begin{aligned} I_{i,m}^{int} &= \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{\mu_{m-1/2}}^{\mu_{m+1/2}} |\tilde{r}^{int}| d\mu \, dx \\ &= \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{\mu_{m-1/2}}^{\mu_{m+1/2}} \left| r_{i,m}^{a} + r_{i,m}^{x} \frac{2}{h_i}(x - x_i) + r_{i,m}^{\mu} \frac{2}{h_m}(\mu - \mu_m) \right| d\mu \, dx. \end{aligned} \tag{A.1}$$

Since the residual is linear, there can only be a maximum of one sign-change per face. As well, if a sign change occurs within a cell, then a sign change will occur on exactly 2 faces. Given this, there are seven possible results for the integral:

1. No sign change

$$I_{i,m}^{int} = h_i h_m |r_{i,m}^{a}|; \tag{A.2}$$

2. Sign change on left and right faces

$$I_{i,m}^{int} = \frac{h_i h_m}{2|r_{i,m}^{\mu}|} \left(|r_{i,m}^{a}|^2 + \frac{|r_{i,m}^{x}|^2}{3} + |r_{i,m}^{\mu}|^2 \right); \tag{A.3}$$

3. Sign change on top and bottom faces

$$I_{i,m}^{int} = \frac{h_i h_m}{2|r_{i,m}^{x}|} \left(|r_{i,m}^{a}|^2 + |r_{i,m}^{x}|^2 + \frac{|r_{i,m}^{\mu}|^2}{3} \right); \tag{A.4}$$

4. Sign change on top and left faces

$$I_{i,m}^{int} = h_i h_m \left| r_{i,m}^a + \frac{(r_{i,m}^a - r_{i,m}^x + r_{i,m}^\mu)^3}{12 r_{i,m}^x r_{i,m}^\mu} \right|; \tag{A.5}$$

5. Sign change on bottom and left faces

$$I_{i,m}^{int} = h_i h_m \left| r_{i,m}^a + \frac{(r_{i,m}^a - r_{i,m}^x - r_{i,m}^\mu)^3}{12 r_{i,m}^x r_{i,m}^\mu} \right|; \tag{A.6}$$

6. Sign change on top and right faces

$$I_{i,m}^{int} = h_i h_m \left| r_{i,m}^a + \frac{(r_{i,m}^a + r_{i,m}^x + r_{i,m}^\mu)^3}{12 r_{i,m}^x r_{i,m}^\mu} \right|; \tag{A.7}$$

7. Sign change on bottom and right faces

$$I_{i,m}^{int} = h_i h_m \left| r_{i,m}^a + \frac{(r_{i,m}^a + r_{i,m}^x - r_{i,m}^\mu)^3}{12 r_{i,m}^x r_{i,m}^\mu} \right|. \tag{A.8}$$

The face integral is defined as

$$\begin{aligned} I_{i,m}^{fac} &= \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{\mu_{m-1/2}}^{\mu_{m+1/2}} |\tilde{r}^{fac}| d\mu \, dx \\ &= \int_{\mu_{m-1/2}}^{\mu_{m+1/2}} |\mu| \left| r_{i,m}^{c1} + r_{i,m}^{c2} \frac{2}{h_m} (\mu - \mu_m) \right| d\mu. \end{aligned} \tag{A.9}$$

There is always an even number of cells in the initial mesh, and each cell has the same width. Thus μ does not change sign within a cell in the initial grid. This fact can not change with our refinement method. This makes this integral similar to the interior integral. The linear part can change sign at most once within a cell. There

are thus two possibilities:

1. No sign change

$$I_{i,m}^{fac} = h_m \left| \mu_m r_{i,m}^{c1} + \frac{h_m}{6} r_{i,m}^{c2} \right|; \tag{A.10}$$

2. Sign change

$$I_{i,m}^{fac} = \frac{h_m}{(r_{i,m}^{c2})^2} \left| \frac{\mu_m}{2} (r_{i,m}^{c2})^3 - \frac{h_m}{12} (r_{i,m}^{c1})^3 + \frac{\mu_m}{2} (r_{i,m}^{c1})^2 r_{i,m}^{c2} + \frac{h_m}{4} r_{i,m}^{c1} (r_{i,m}^{c2})^2 \right|. \tag{A.11}$$

APPENDIX B

JUMP ERROR INDICATORS

As discussed in section 2.4, there are a total of twelve possible formulations for the jump error indicators, three for each of the four boundaries. For the left boundary, these are as follows:

1. Adjacent cell and current cell at the same refinement level

$$\xi_i^{left} = \int_{\mu_i^B}^{\mu_i^T} \left| \left(\psi_j^a + \psi_j^x + \psi_j^\mu \frac{2}{h_j^\mu}(\mu - \mu_j^C) \right) - \left(\psi_i^a - \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C) \right) \right| d\mu; \tag{B.1}$$

2. Adjacent cell is more refined

$$\begin{aligned} \xi_i^{left} = \int_{\mu_i^B}^{\mu_i^T} \Bigg| & \frac{1}{2}\left(\psi_j^a + \psi_j^x + \psi_k^a + \psi_k^x\right) \\ & + \frac{1}{4}\left(\psi_j^\mu + \psi_k^\mu + 3\left(\psi_j^a + \psi_j^x - \psi_k^a - \psi_k^x\right)\right)\frac{2}{h_j^\mu}(\mu - \mu_j^C) \\ & - \left(\psi_i^a - \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C) \right) \Bigg| \, d\mu; \end{aligned} \tag{B.2}$$

3. Current cell is more refined, two options:

   (a) Current cell is relatively on top

$$\begin{aligned} \xi_i^{left} = \int_{\mu_i^B}^{\mu_i^T} \Bigg| & \left(\psi_j^a + \psi_j^x + \frac{1}{2}\psi_j^\mu + \frac{1}{2}\psi_j^\mu \frac{2}{h_j^\mu}(\mu - \mu_j^C) \right) \\ & - \left(\psi_i^a - \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C) \right) \Bigg| \, d\mu; \end{aligned} \tag{B.3a}$$

(b) Current cell is relatively on bottom

$$\xi_i^{left} = \int_{\mu_i^B}^{\mu_i^T} \left| \left(\psi_j^a + \psi_j^x - \frac{1}{2}\psi_j^\mu + \frac{1}{2}\psi_j^\mu \frac{2}{h_j^\mu}(\mu - \mu_j^C) \right) - \left(\psi_i^a - \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C) \right) \right| d\mu. \tag{B.3b}$$

For the right boundary, these are as follows:

1. Adjacent cell and current cell at the same refinement level

$$\xi_i^{right} = \int_{\mu_i^B}^{\mu_i^T} \left| \left(\psi_j^a - \psi_j^x + \psi_j^\mu \frac{2}{h_j^\mu}(\mu - \mu_j^C) \right) - \left(\psi_i^a + \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C) \right) \right| d\mu; \tag{B.4}$$

2. Adjacent cell is more refined

$$\xi_i^{right} = \int_{\mu_i^B}^{\mu_i^T} \left| \frac{1}{2}\left(\psi_j^a - \psi_j^x + \psi_k^a - \psi_k^x\right) + \frac{1}{4}\left(\psi_j^\mu + \psi_k^\mu + 3\left(\psi_j^a - \psi_j^x - \psi_k^a + \psi_k^x\right)\right)\frac{2}{h_j^\mu}(\mu - \mu_j^C) - \left(\psi_i^a + \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C) \right) \right| d\mu; \tag{B.5}$$

3. Current cell is more refined, two options:

(a) Current cell is relatively on top

$$\xi_i^{right} = \int_{\mu_i^B}^{\mu_i^T} \left| \left(\psi_j^a - \psi_j^x + \frac{1}{2}\psi_j^\mu + \frac{1}{2}\psi_j^\mu \frac{2}{h_j^\mu}(\mu - \mu_j^C) \right) - \left(\psi_i^a + \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C) \right) \right| d\mu; \tag{B.6a}$$

(b) Current cell is relatively on bottom

$$\xi_i^{right} = \int_{\mu_i^B}^{\mu_i^T} \left| \left(\psi_j^a - \psi_j^x - \frac{1}{2}\psi_j^\mu + \frac{1}{2}\psi_j^\mu \frac{2}{h_j^\mu}(\mu - \mu_j^C) \right) - \left(\psi_i^a + \psi_i^x + \psi_i^\mu \frac{2}{h_i^\mu}(\mu - \mu_i^C) \right) \right| d\mu. \tag{B.6b}$$

For the top boundary, these are as follows:

1. Adjacent cell and current cell at the same refinement level

$$\xi_i^{top} = \int_{x_i^L}^{x_i^R} \left| \left(\psi_j^a - \psi_j^\mu + \psi_j^x \frac{2}{h_j^x}(x - x_j^C) \right) - \left(\psi_i^a + \psi_i^\mu + \psi_i^x \frac{2}{h_i^x}(x - x_i^C) \right) \right| dx; \tag{B.7}$$

2. Adjacent cell is more refined

$$\xi_i^{top} = \int_{x_i^L}^{x_i^R} \left| \frac{1}{2}\left(\psi_j^a - \psi_j^\mu + \psi_k^a - \psi_k^\mu\right) + \frac{1}{4}\left(\psi_j^x + \psi_k^x + 3\left(\psi_j^a - \psi_j^\mu - \psi_k^a + \psi_k^\mu\right)\right)\frac{2}{h_j^x}(x - x_j^C) - \left(\psi_i^a + \psi_i^\mu + \psi_i^x \frac{2}{h_i^x}(x - x_i^C) \right) \right| dx; \tag{B.8}$$

3. Current cell is more refined, two options:

(a) Current cell is relatively on right

$$\xi_i^{top} = \int_{x_i^L}^{x_i^R} \left| \left(\psi_j^a - \psi_j^\mu + \frac{1}{2}\psi_j^x + \frac{1}{2}\psi_j^x \frac{2}{h_j^x}(x - x_j^C) \right) - \left(\psi_i^a + \psi_i^\mu + \psi_i^x \frac{2}{h_i^x}(x - x_i^C) \right) \right| dx; \tag{B.9a}$$

(b) Current cell is relatively on left

$$\xi_i^{top} = \int_{x_i^L}^{x_i^R} \left| \left(\psi_j^a - \psi_j^\mu - \frac{1}{2}\psi_j^x + \frac{1}{2}\psi_j^x \frac{2}{h_j^x}(x - x_j^C) \right) - \left(\psi_i^a + \psi_i^\mu + \psi_i^x \frac{2}{h_i^x}(x - x_i^C) \right) \right| dx. \tag{B.9b}$$

For the bottom boundary, these are as follows:

1. Adjacent cell and current cell at the same refinement level

$$\xi_i^{bottom} = \int_{x_i^L}^{x_i^R} \left| \left(\psi_j^a - +\psi_j^\mu + \psi_j^x \frac{2}{h_j^x}(x - x_j^C) \right) - \left(\psi_i^a - \psi_i^\mu + \psi_i^x \frac{2}{h_i^x}(x - x_i^C) \right) \right| dx; \tag{B.10}$$

2. Adjacent cell is more refined

$$\xi_i^{bottom} = \int_{x_i^L}^{x_i^R} \left| \frac{1}{2}\left(\psi_j^a + \psi_j^\mu + \psi_k^a + \psi_k^\mu\right) + \frac{1}{4}\left(\psi_j^x + \psi_k^x + 3\left(\psi_j^a + \psi_j^\mu - \psi_k^a - \psi_k^\mu\right)\right) \frac{2}{h_j^x}(x - x_j^C) - \left(\psi_i^a - \psi_i^\mu + \psi_i^x \frac{2}{h_i^x}(x - x_i^C) \right) \right| dx; \tag{B.11}$$

3. Current cell is more refined, two options:

(a) Current cell is relatively on right

$$\xi_i^{bottom} = \int_{x_i^L}^{x_i^R} \left| \left(\psi_j^a + \psi_j^\mu + \frac{1}{2}\psi_j^x + \frac{1}{2}\psi_j^x \frac{2}{h_j^x}(x - x_j^C) \right) - \left(\psi_i^a - \psi_i^\mu + \psi_i^x \frac{2}{h_i^x}(x - x_i^C) \right) \right| dx; \tag{B.12a}$$

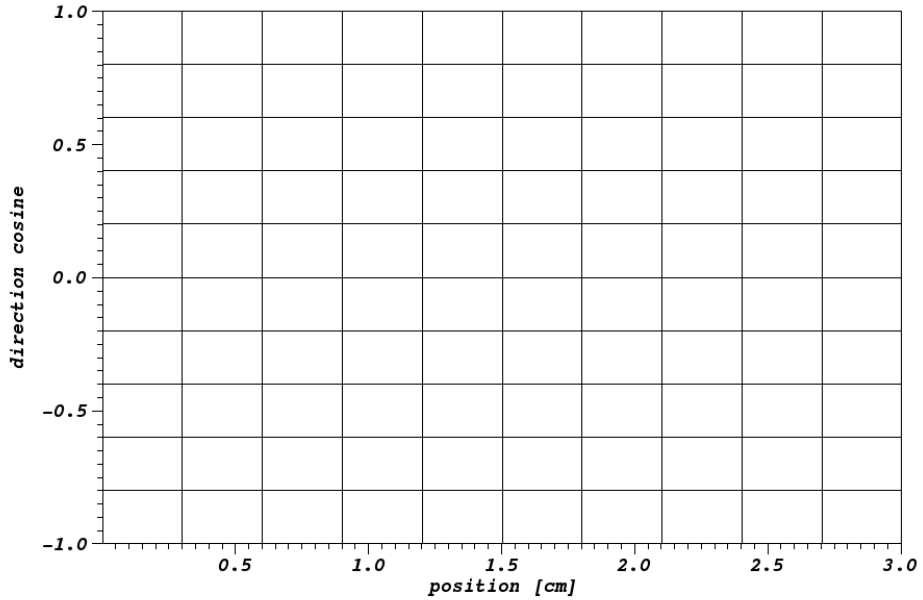
(b) Current cell is relatively on left

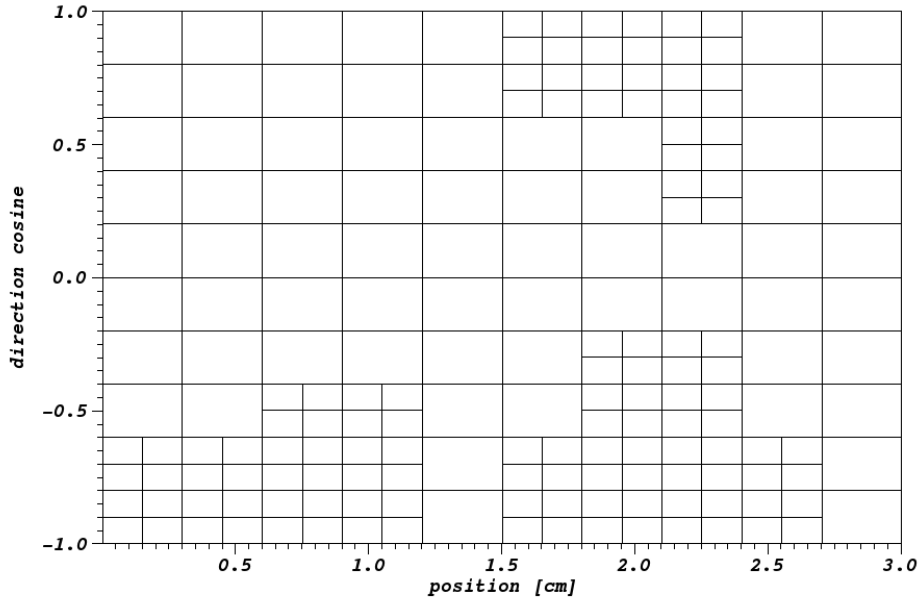
$$\xi_i^{bottom} = \int_{x_i^L}^{x_i^R} \left| \left(\psi_j^a + \psi_j^\mu - \frac{1}{2}\psi_j^x + \frac{1}{2}\psi_j^x \frac{2}{h_j^x}(x - x_j^C) \right) - \left(\psi_i^a - \psi_i^\mu + \psi_i^x \frac{2}{h_i^x}(x - x_i^C) \right) \right| dx. \tag{B.12b}$$

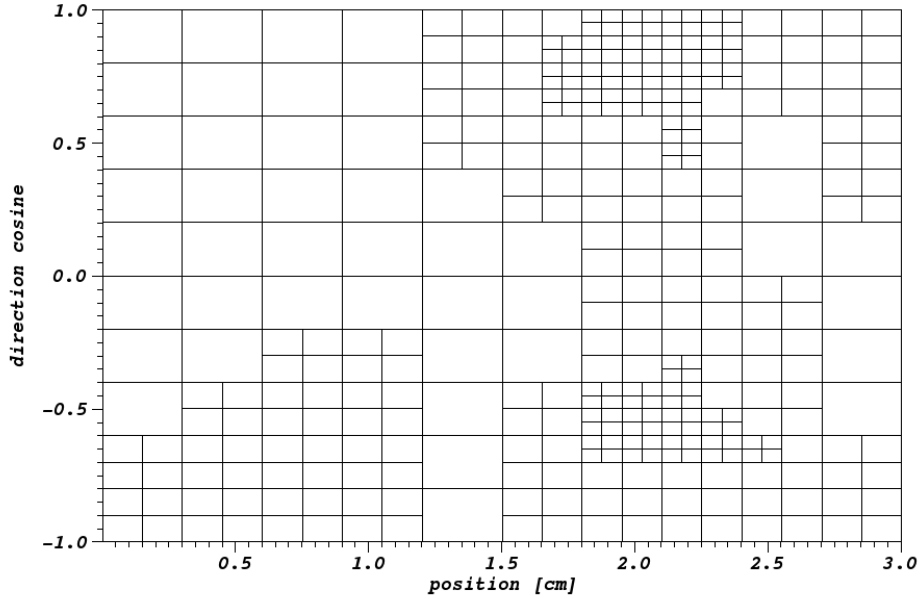
Note that each of these is a one-dimension absolute integral of a linear function. Thus the exact integral can be calculated, similar to Appendix A.

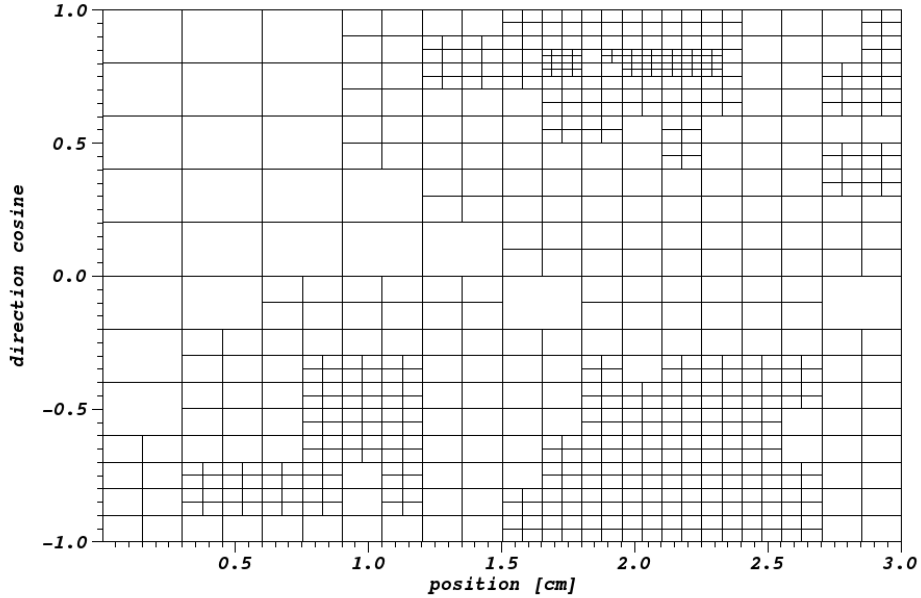
APPENDIX C

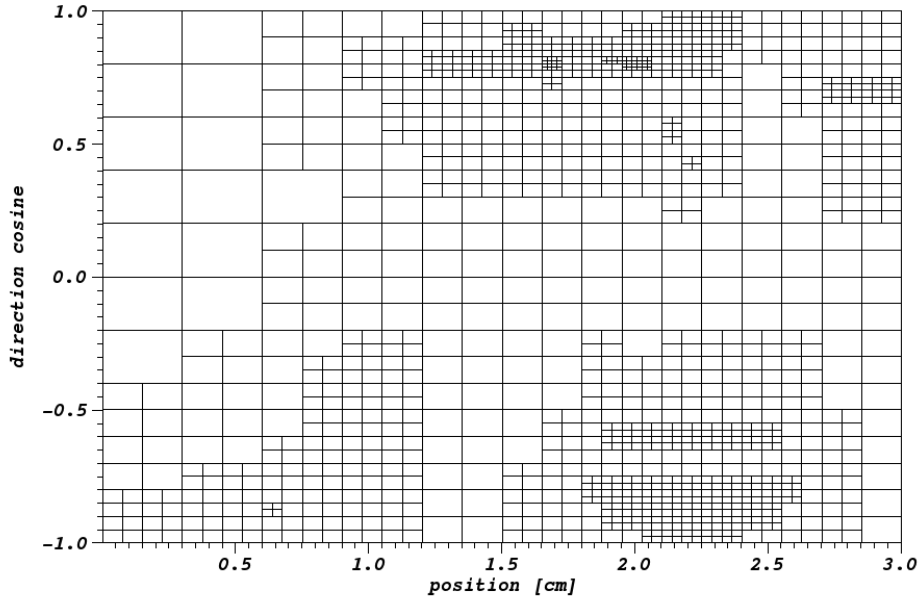
MANUFACTURED SOLUTION MESHES

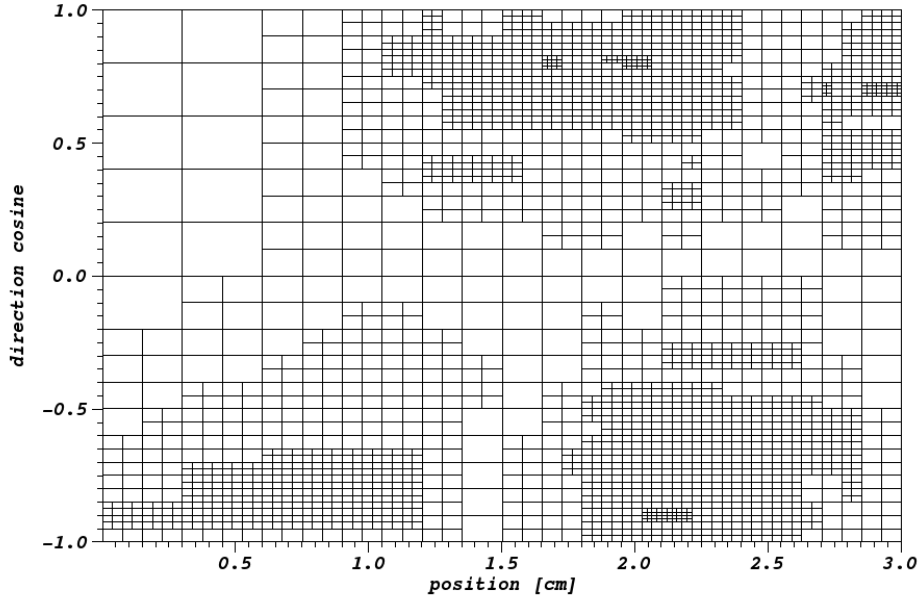


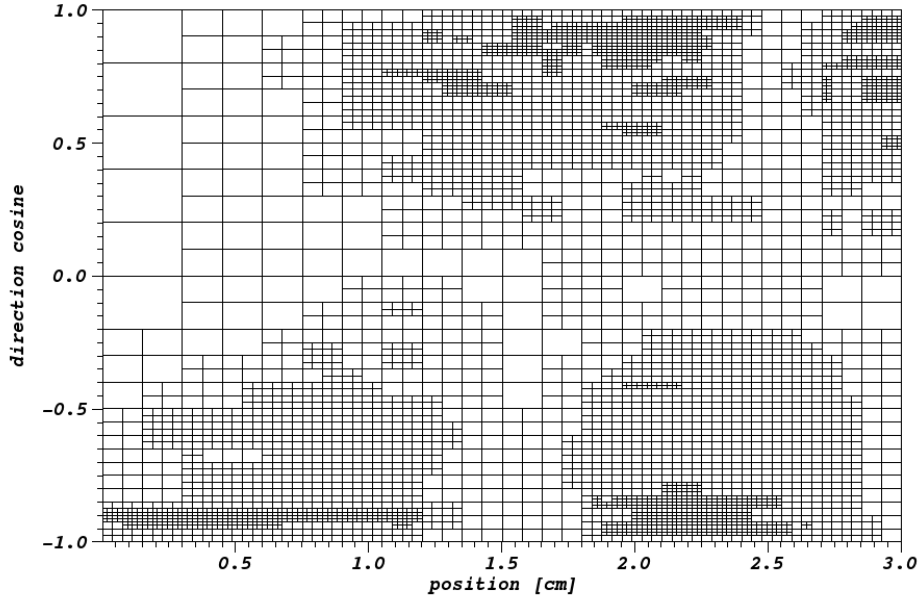


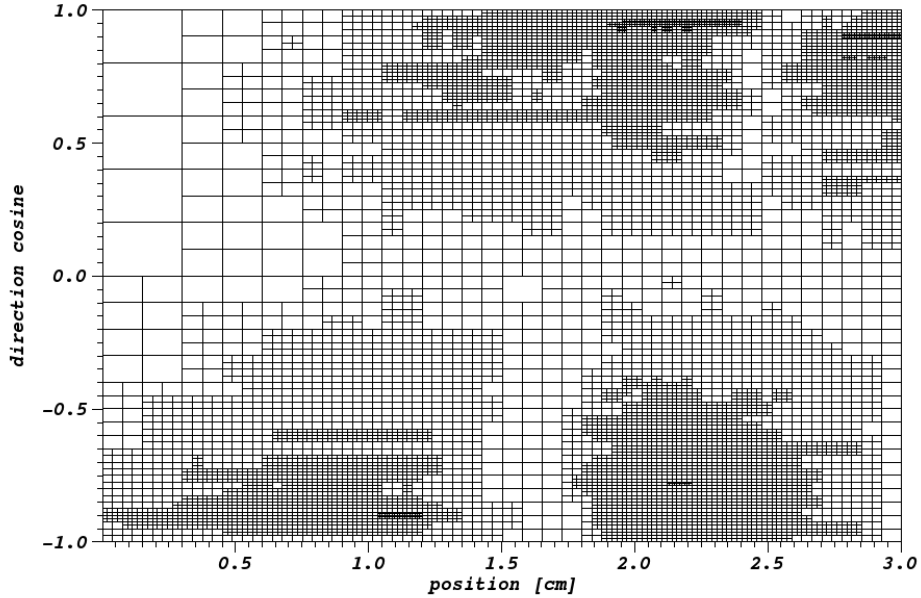

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

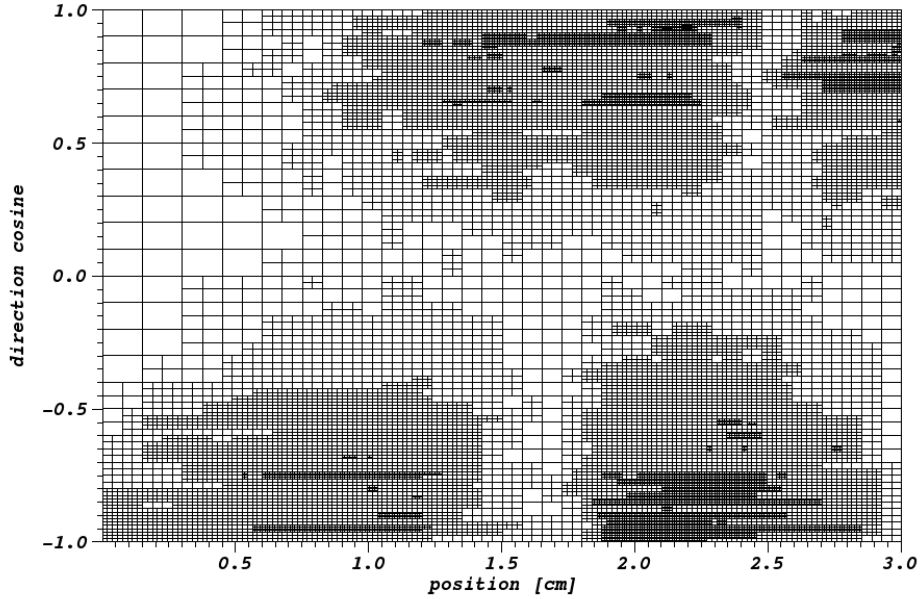

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

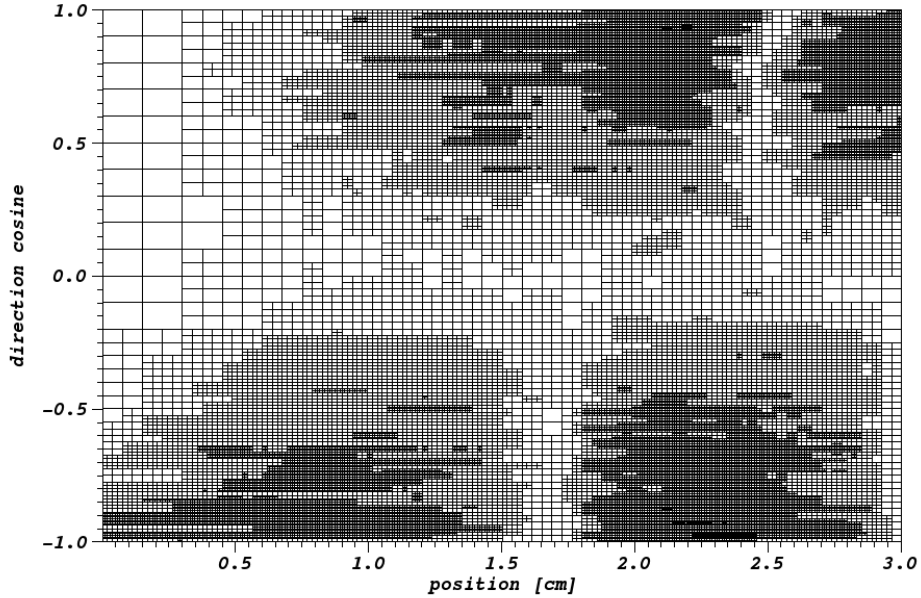

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

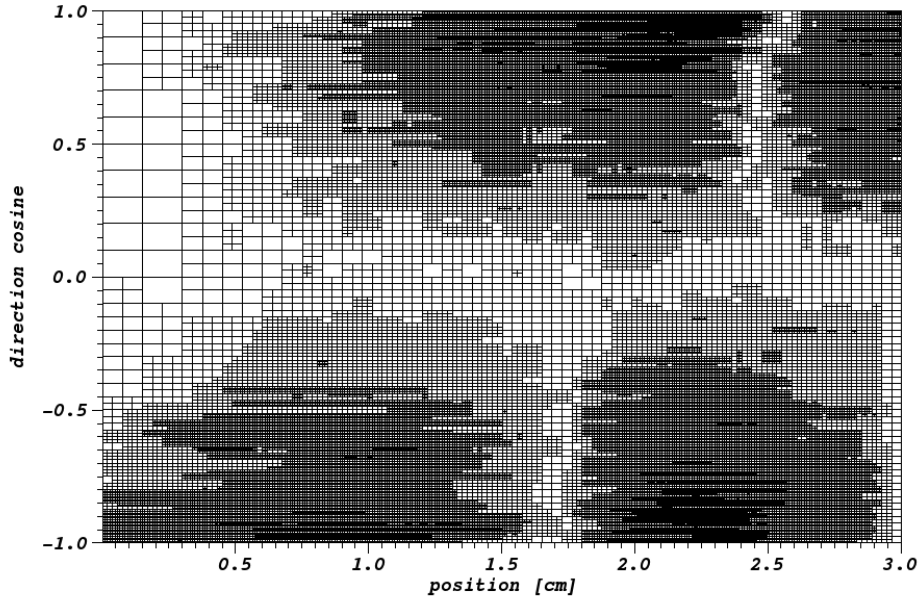

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

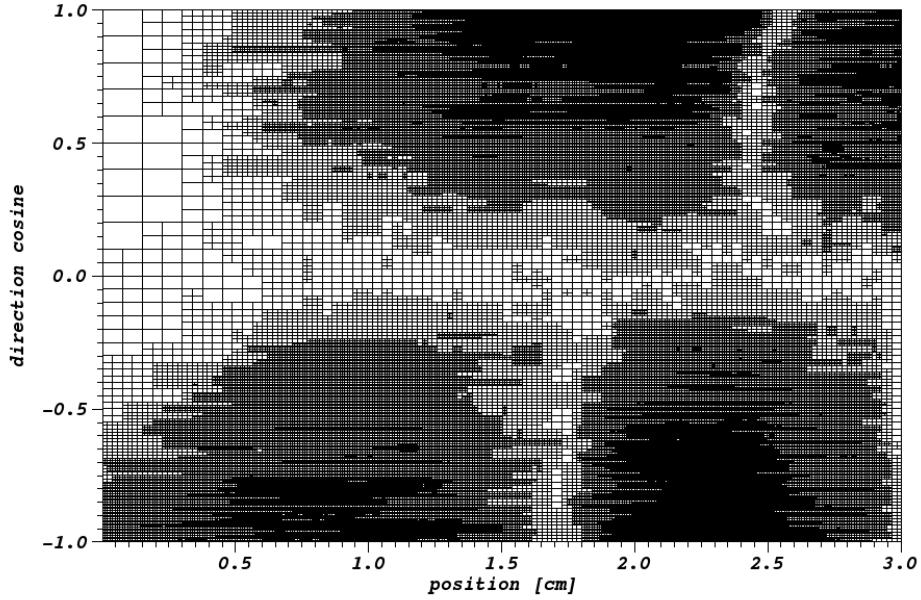

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

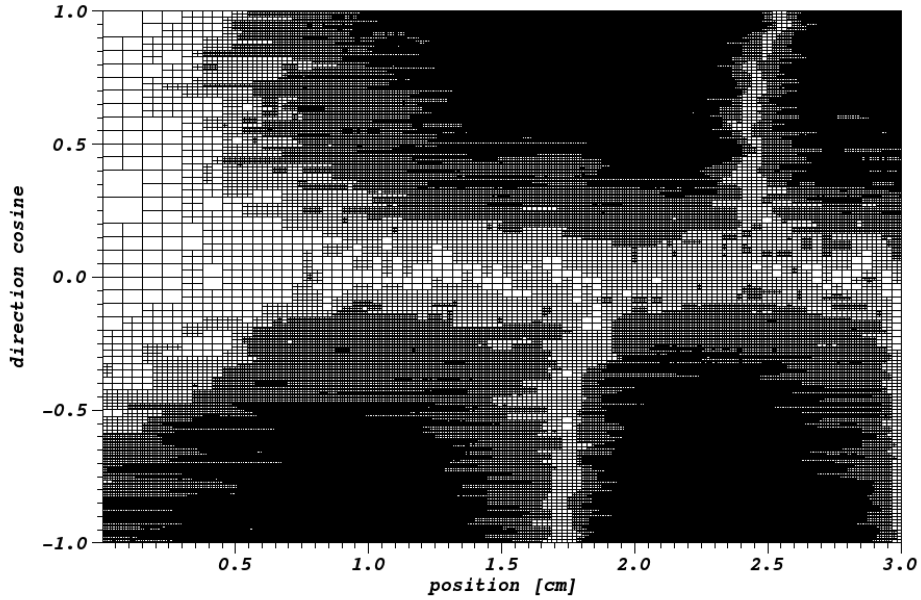

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

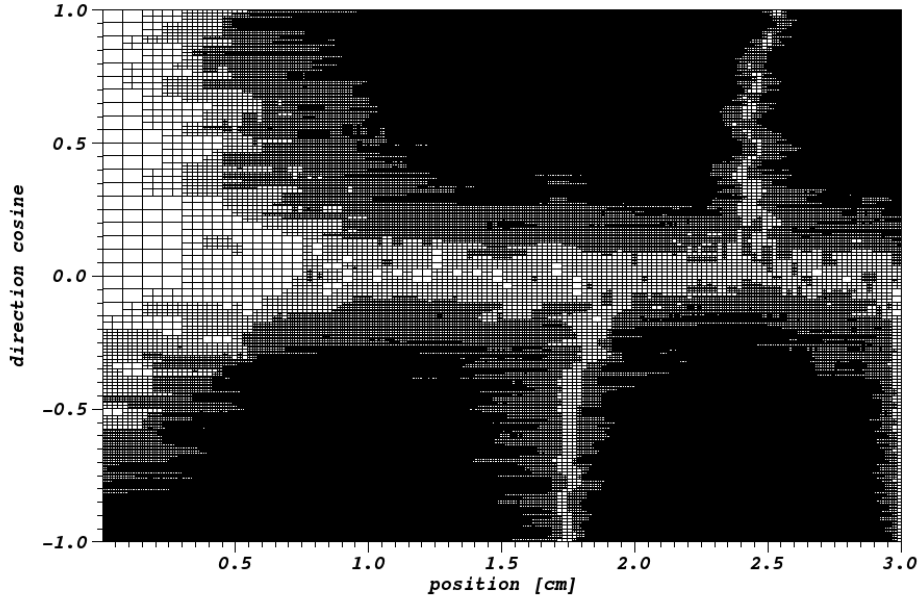

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

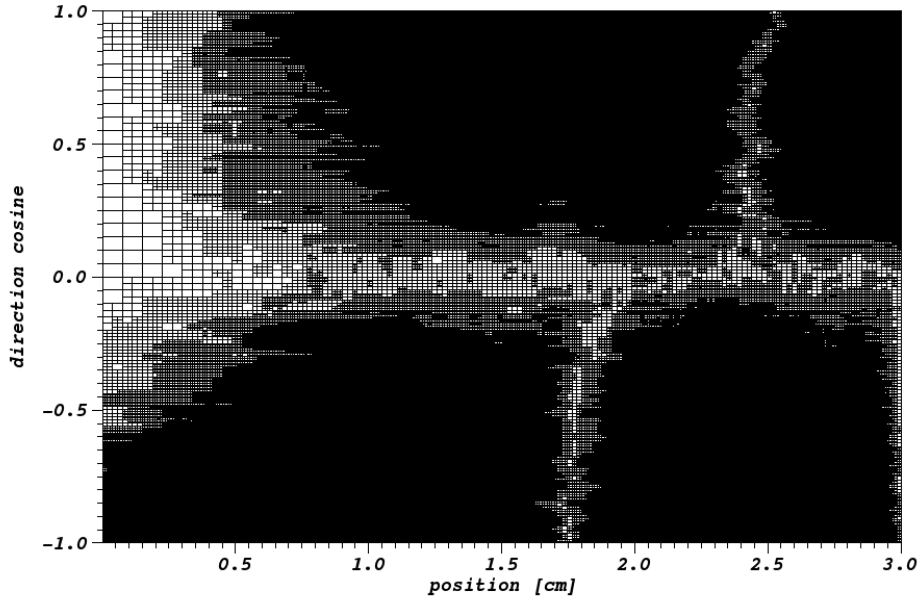

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

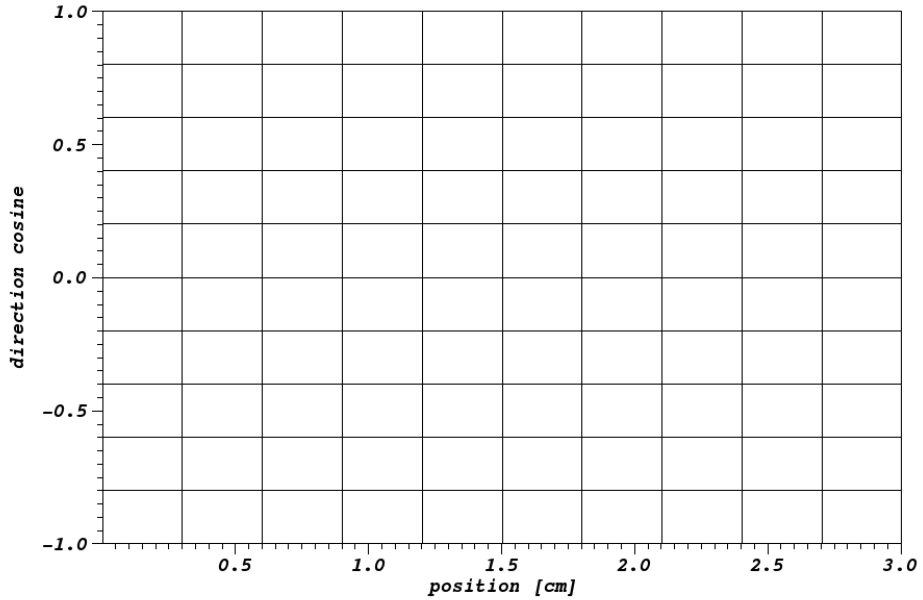

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

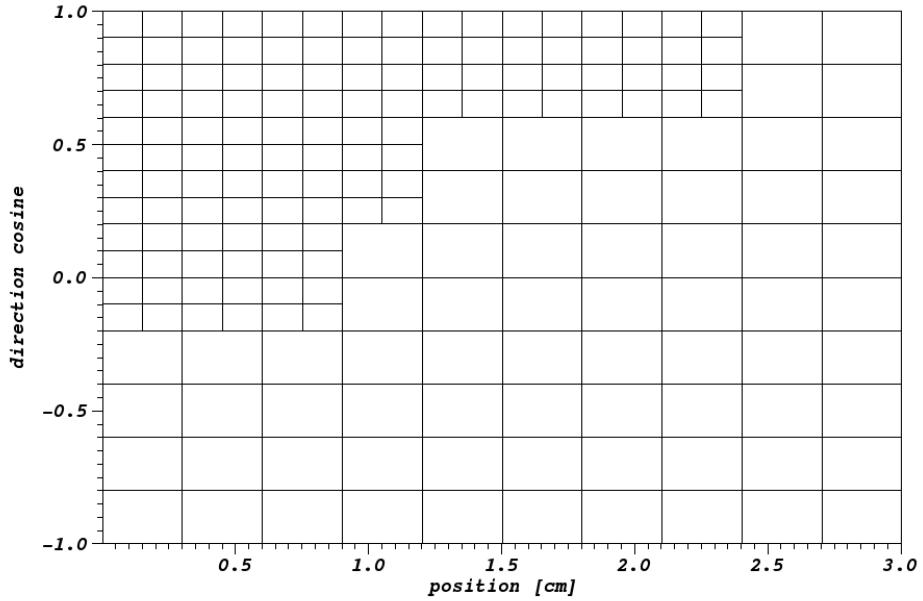

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

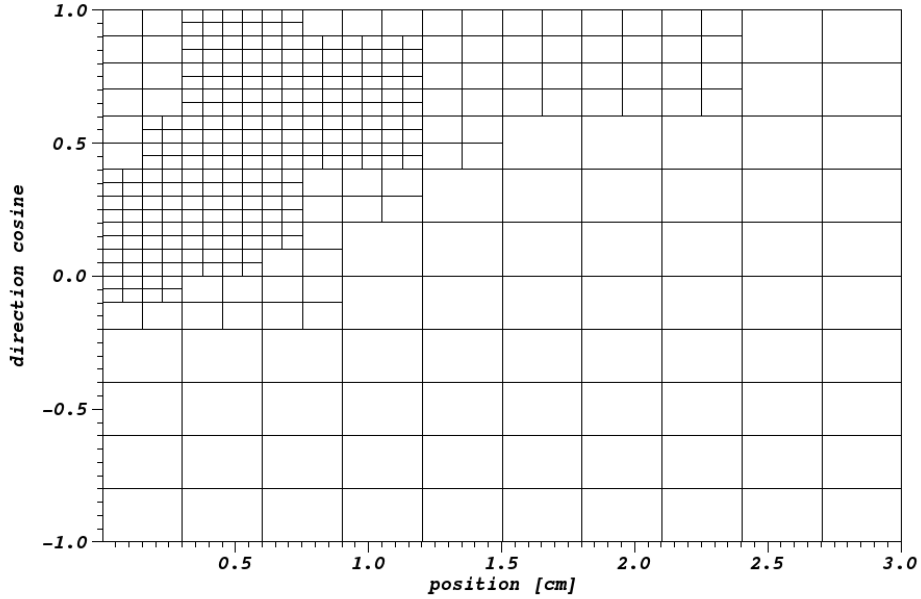

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

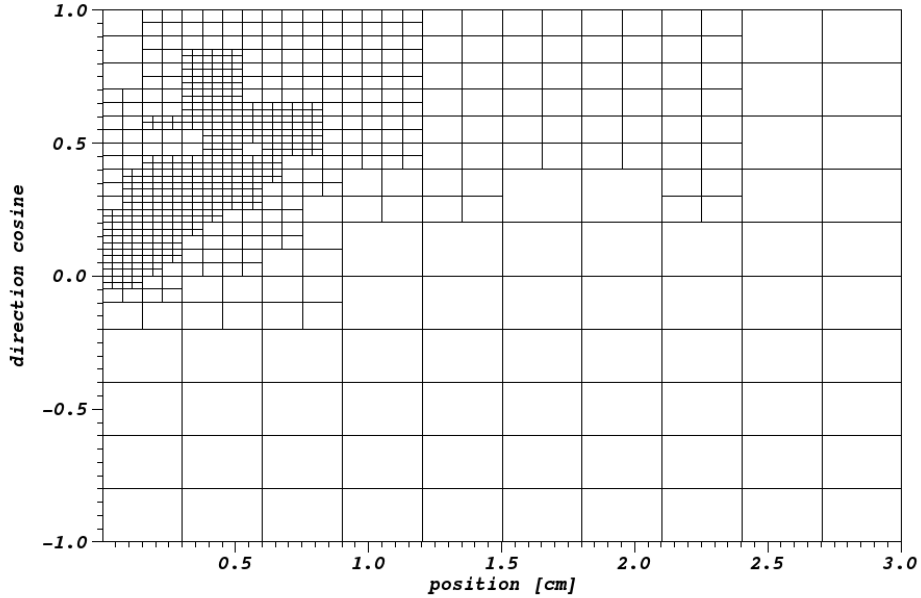
APPENDIX D

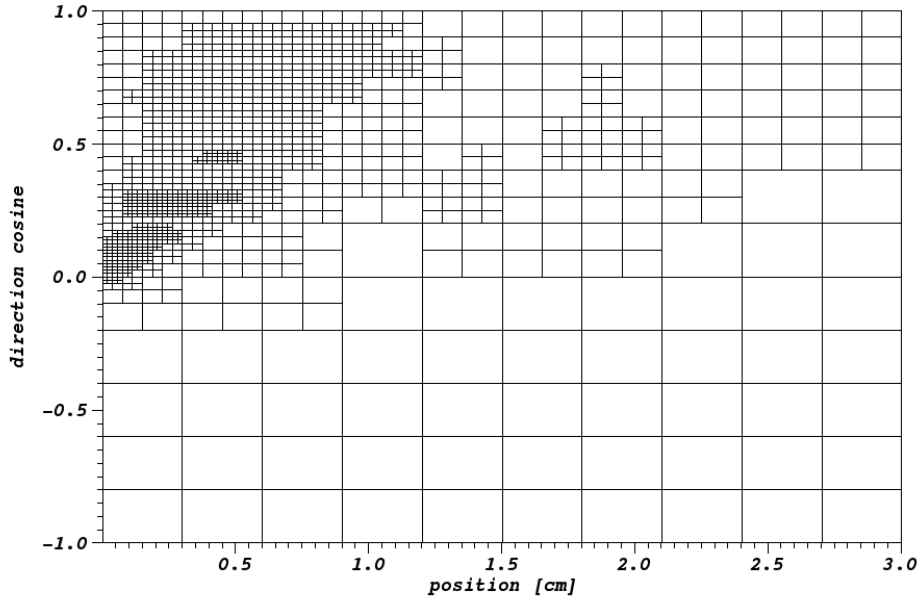
ATTENUATION SOLUTION MESHES

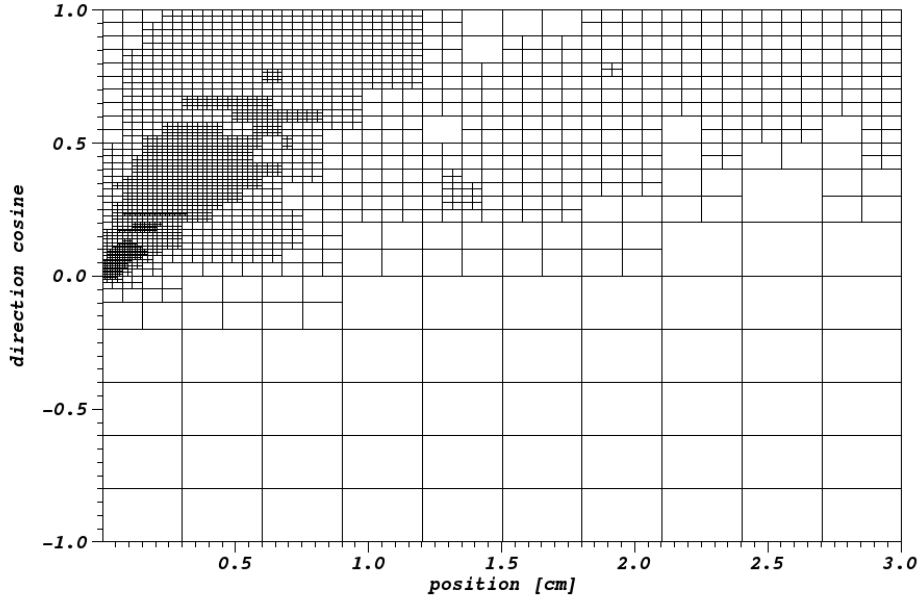


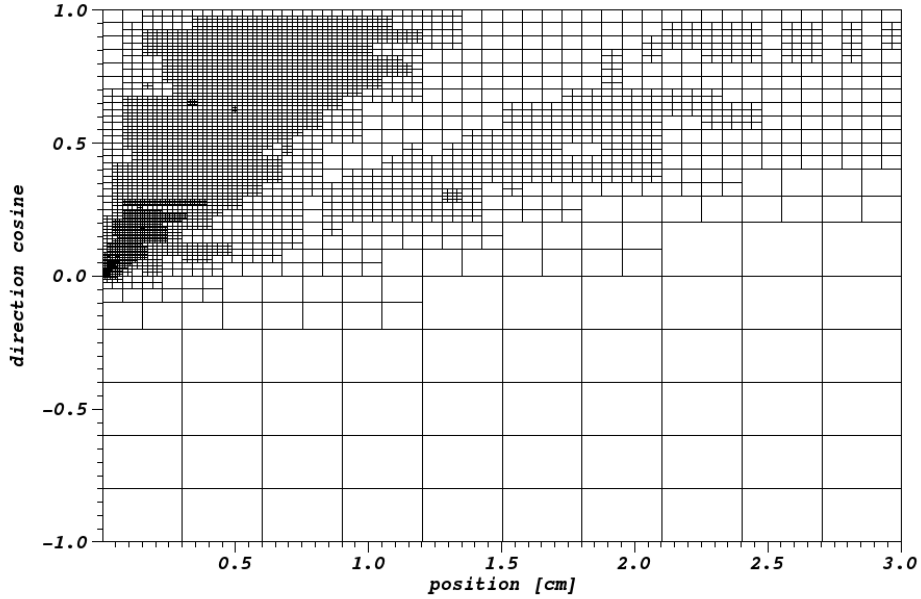


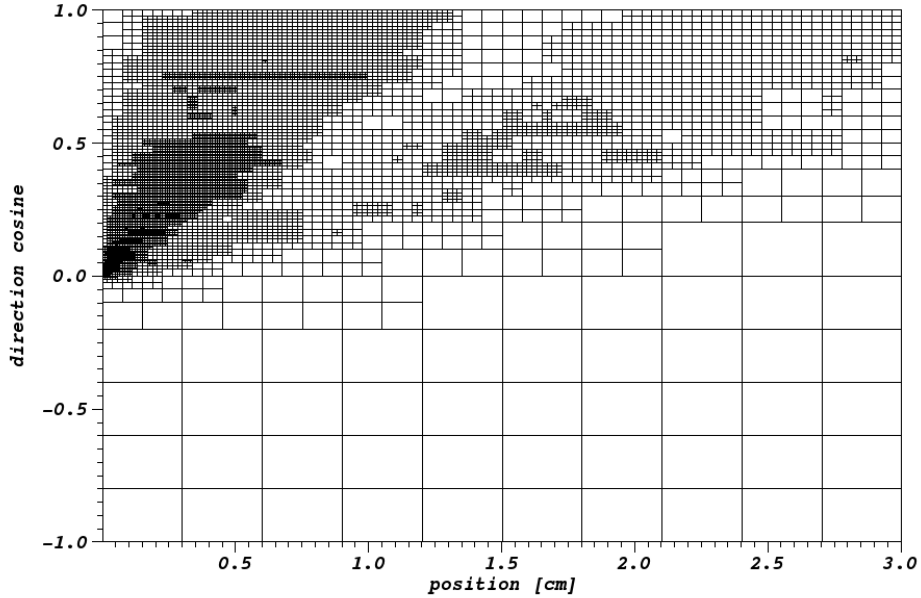

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

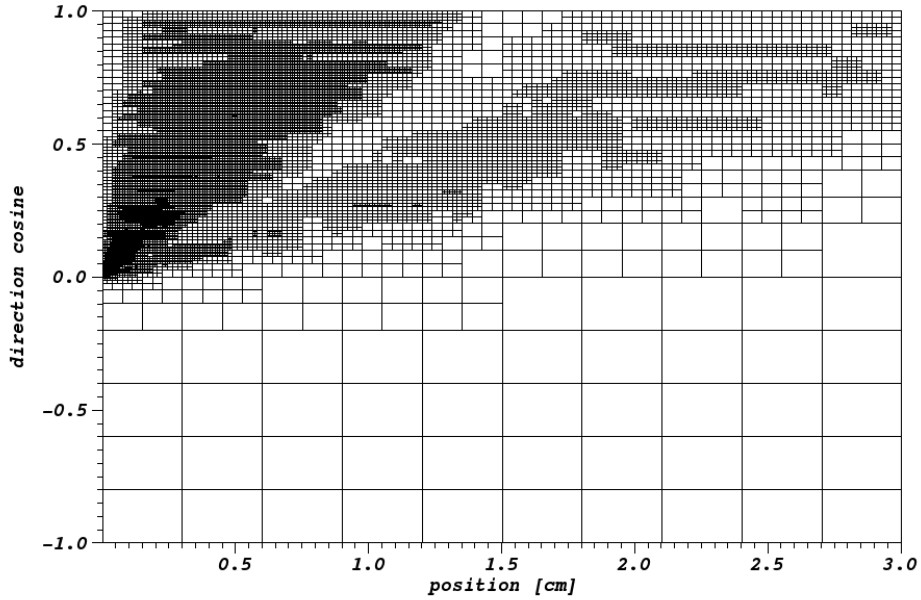

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

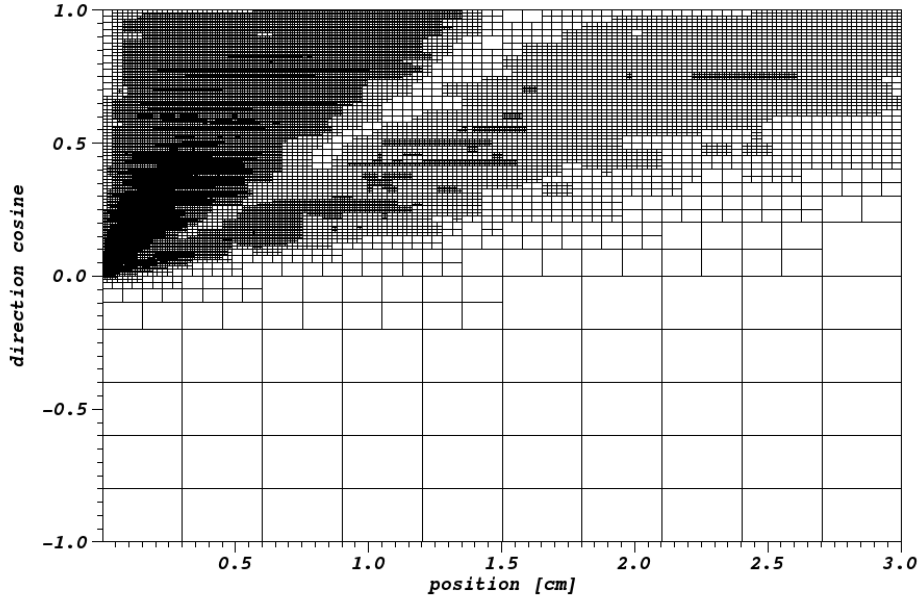

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

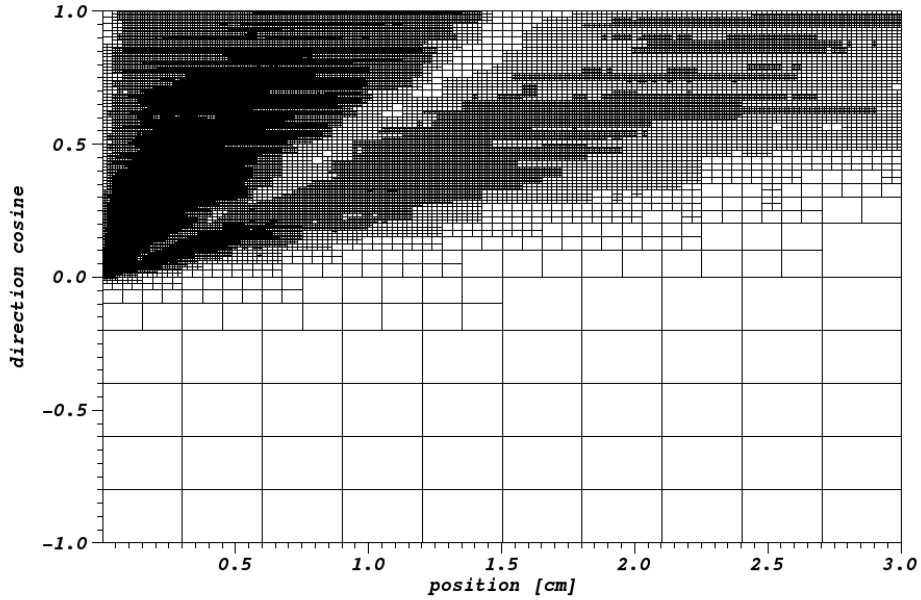

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

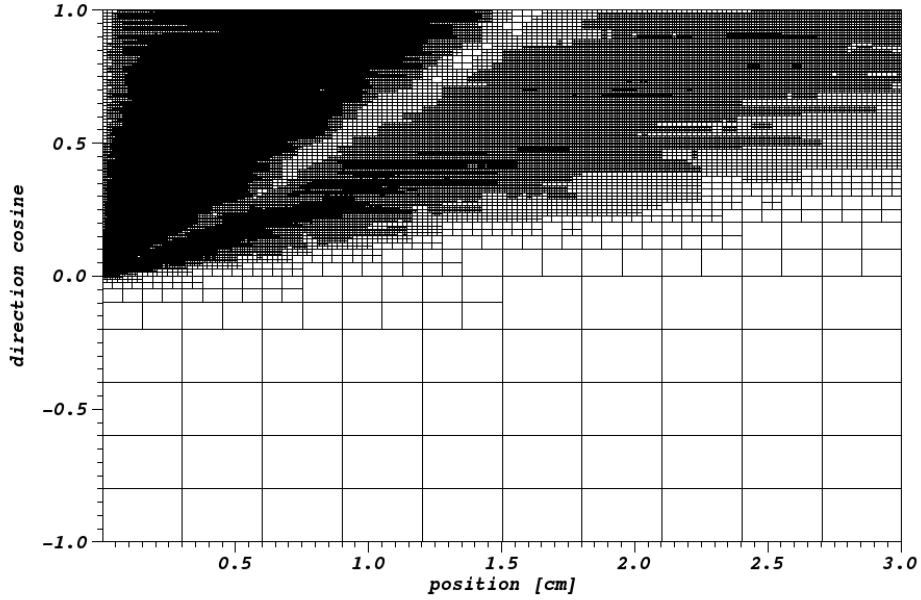

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

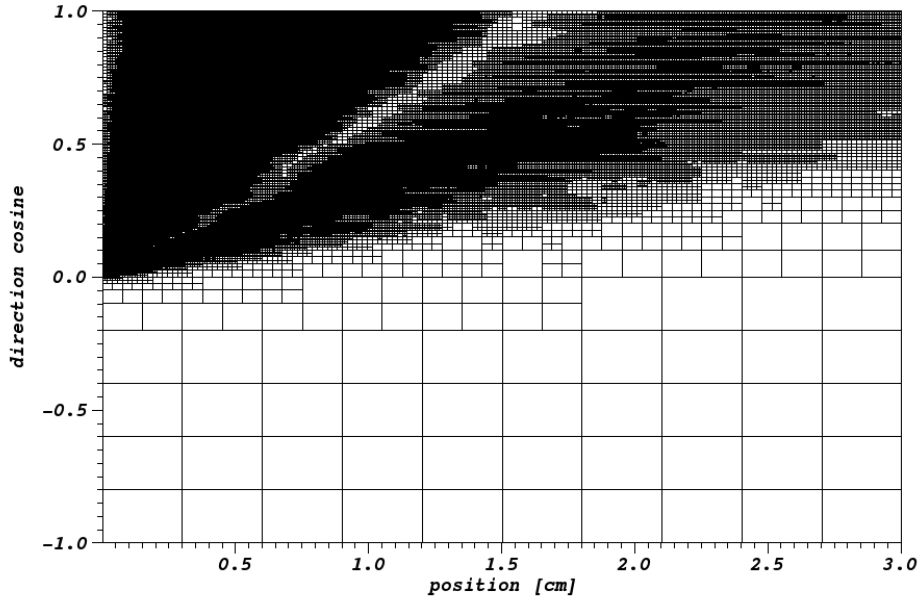

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

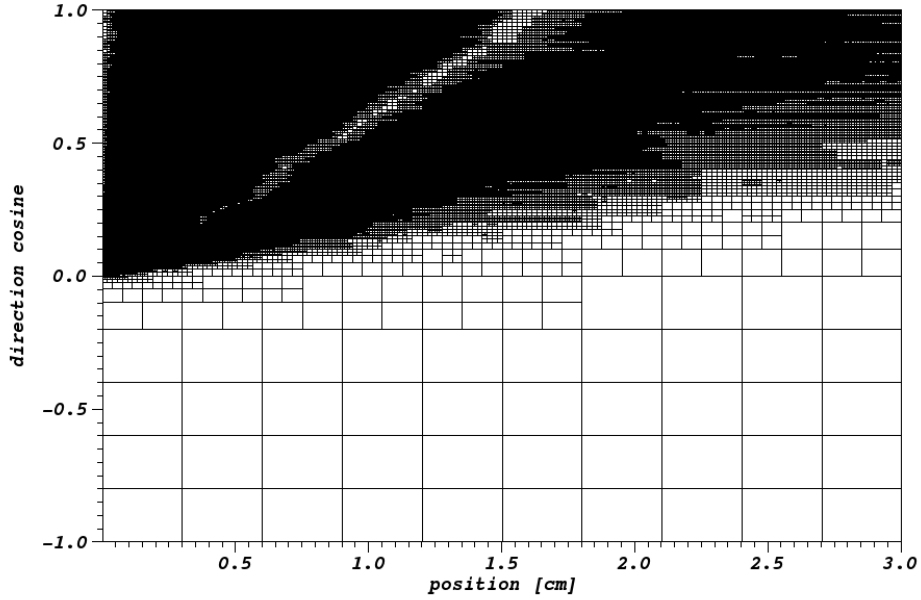

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

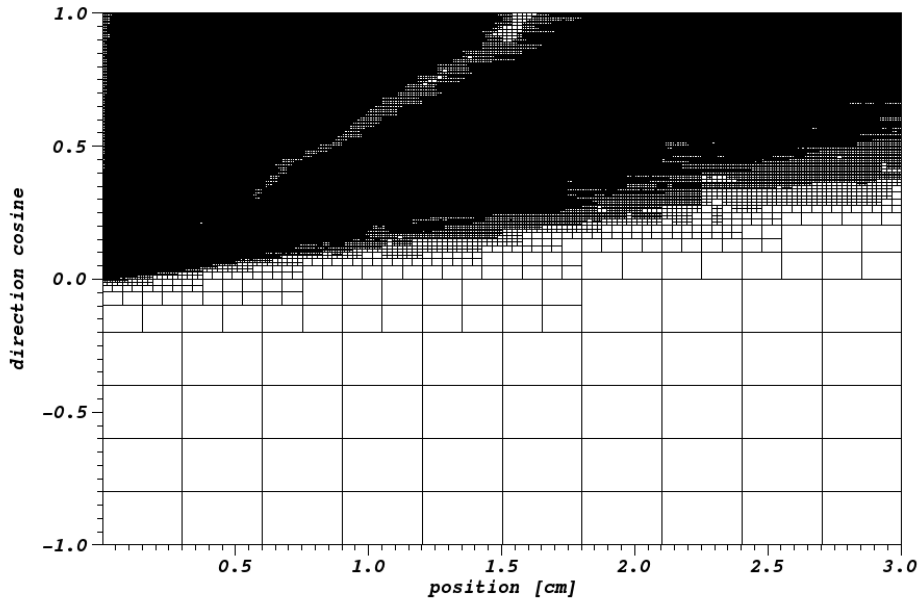

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

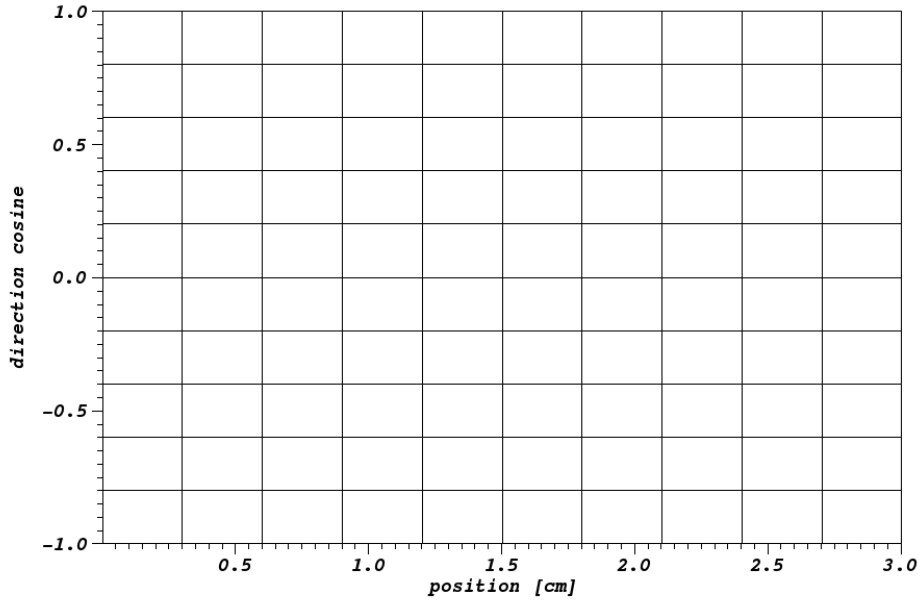

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

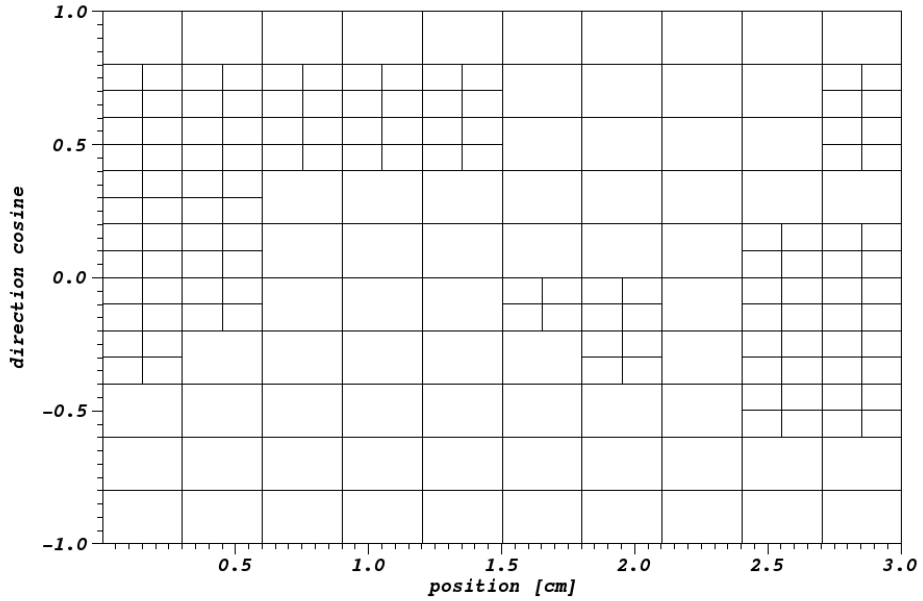

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

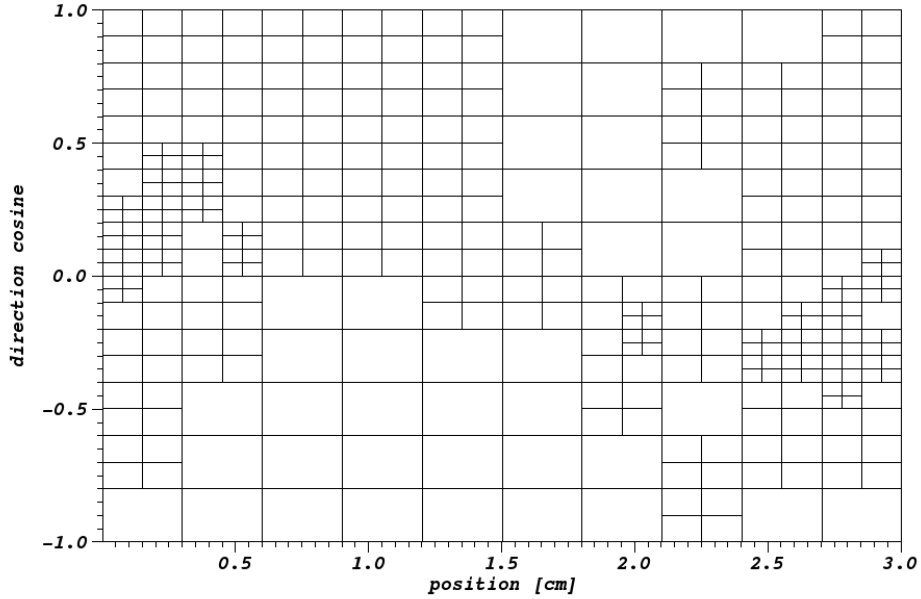

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

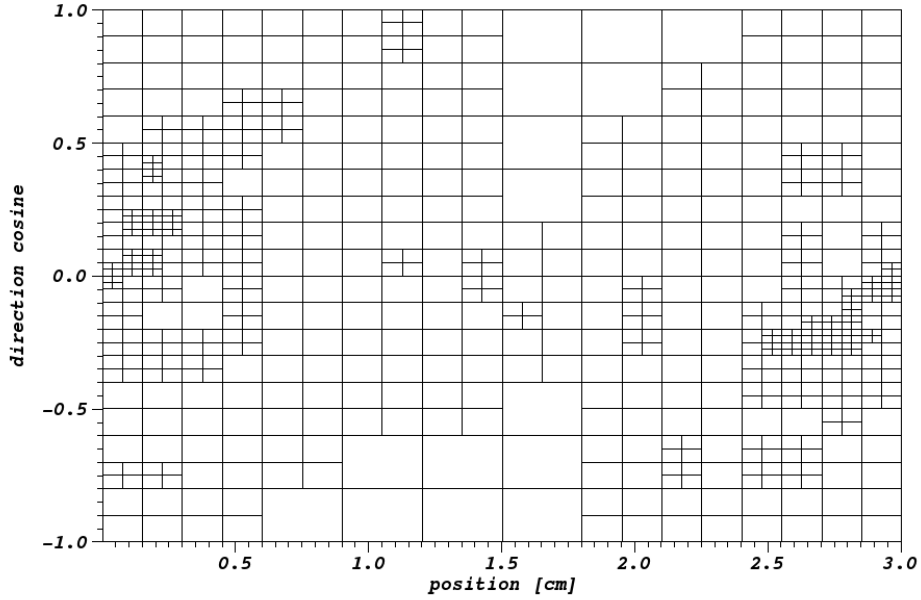
APPENDIX E

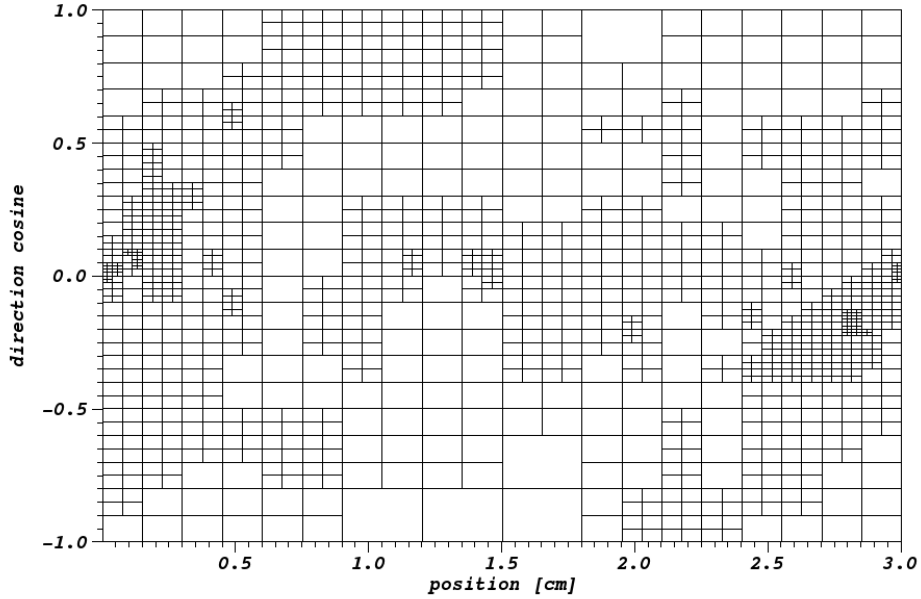
INTERNAL SOURCE MESHES

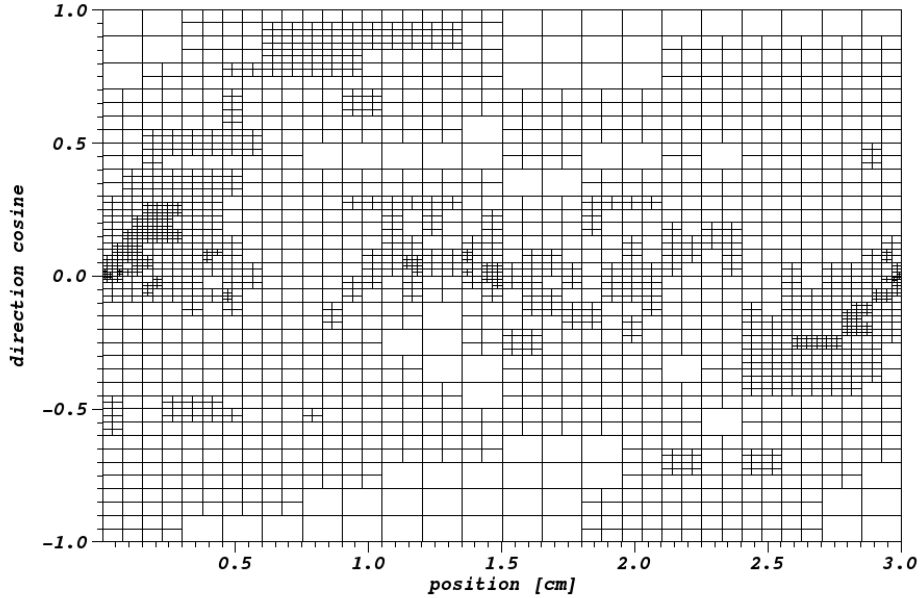


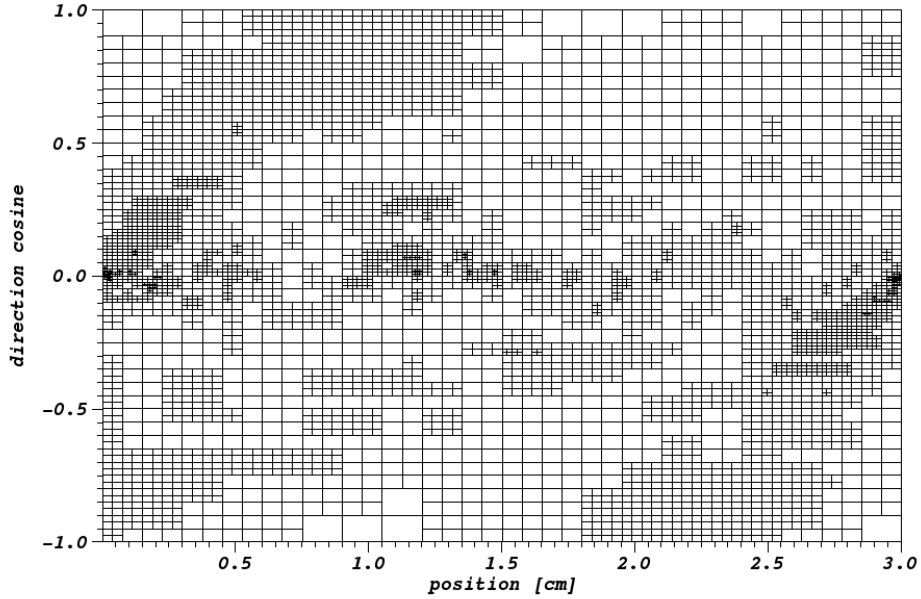


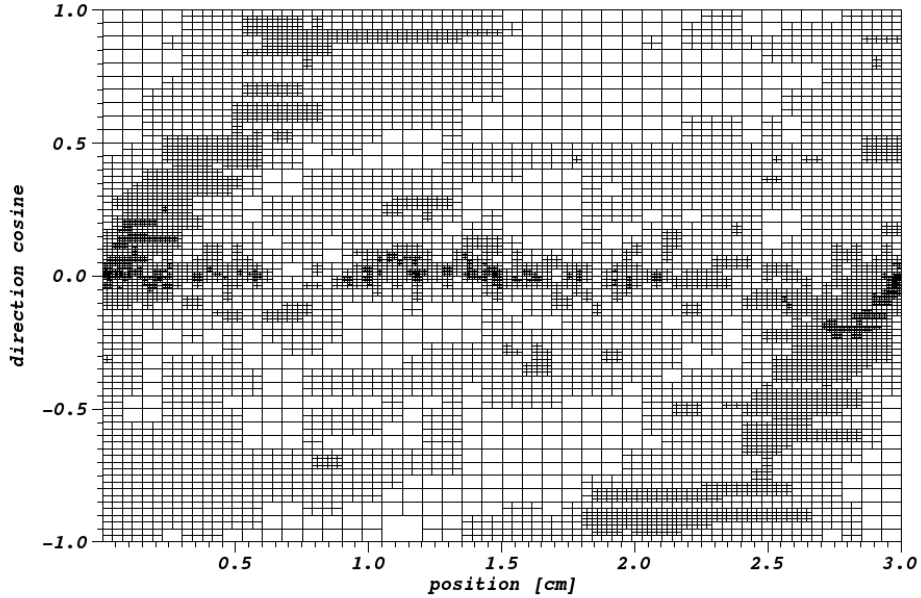

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

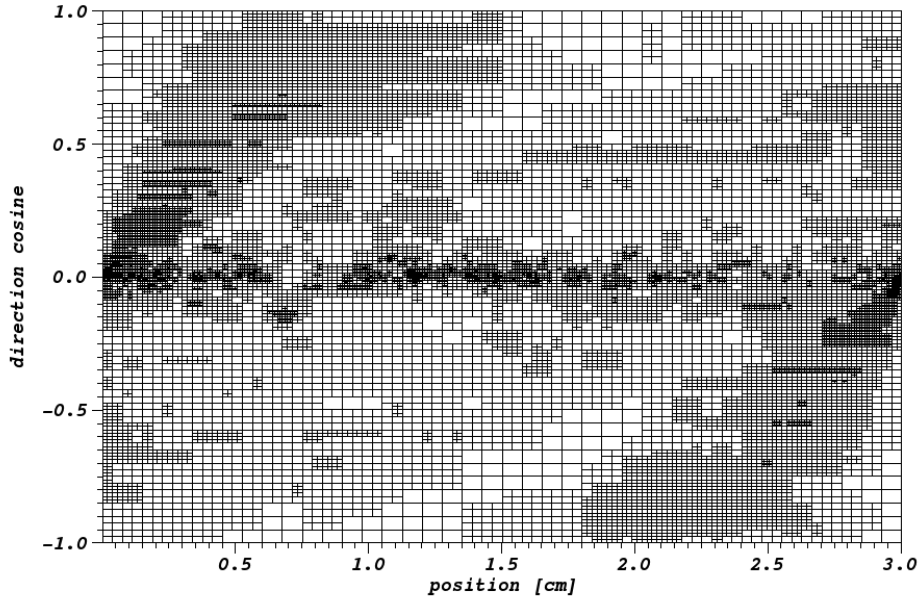

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

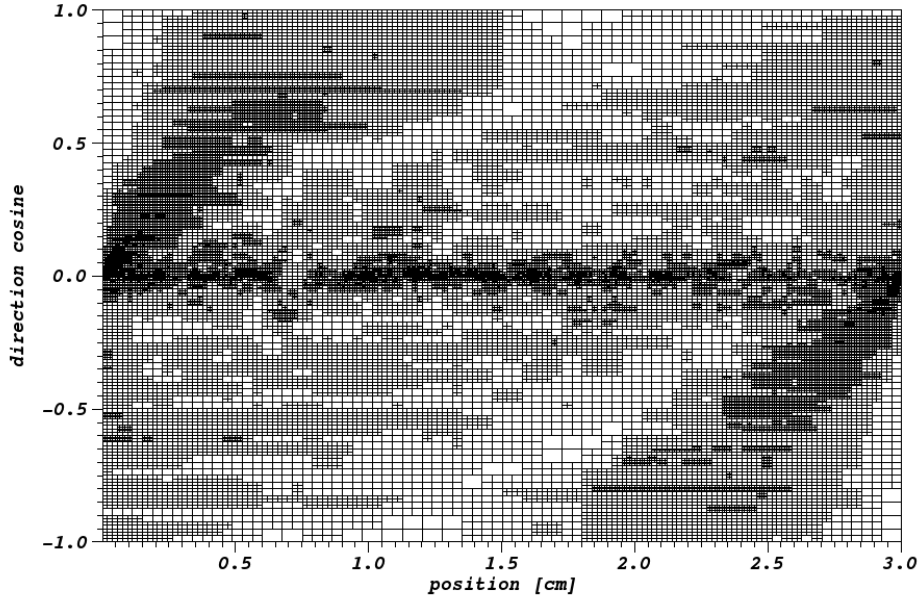

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

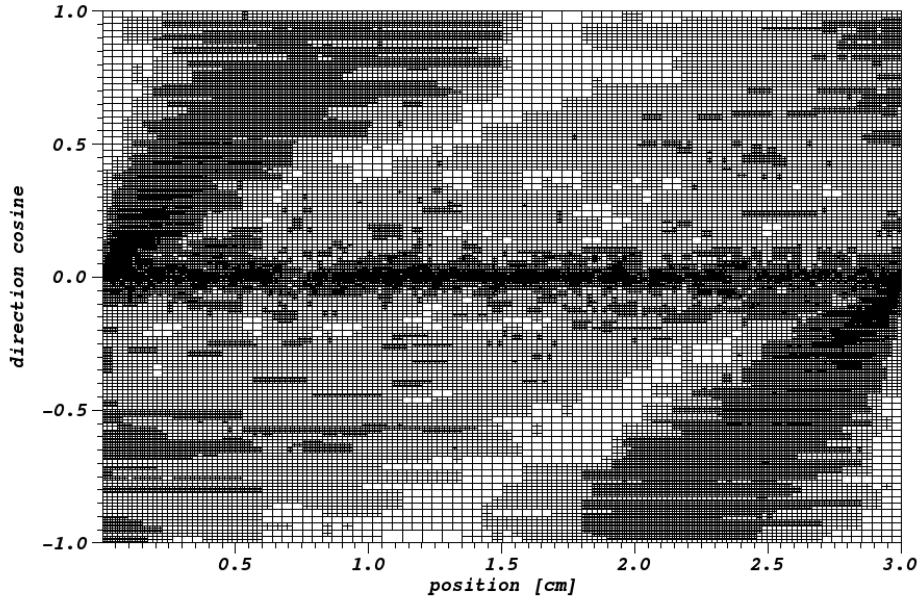

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

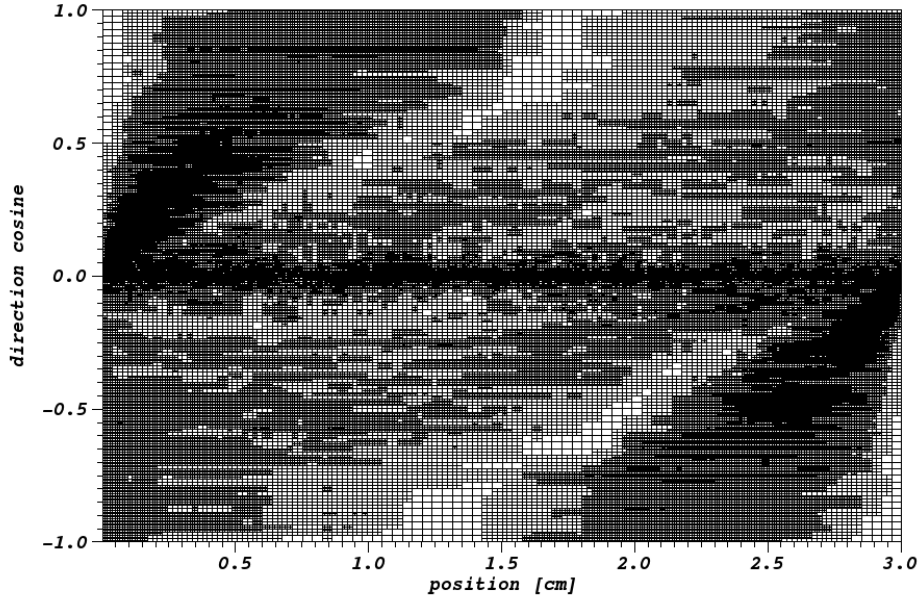

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

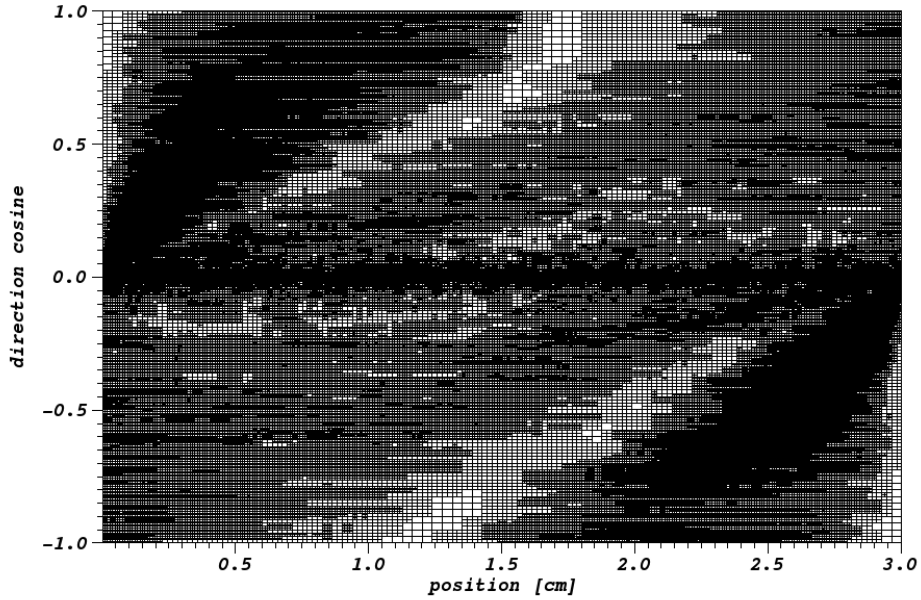

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]

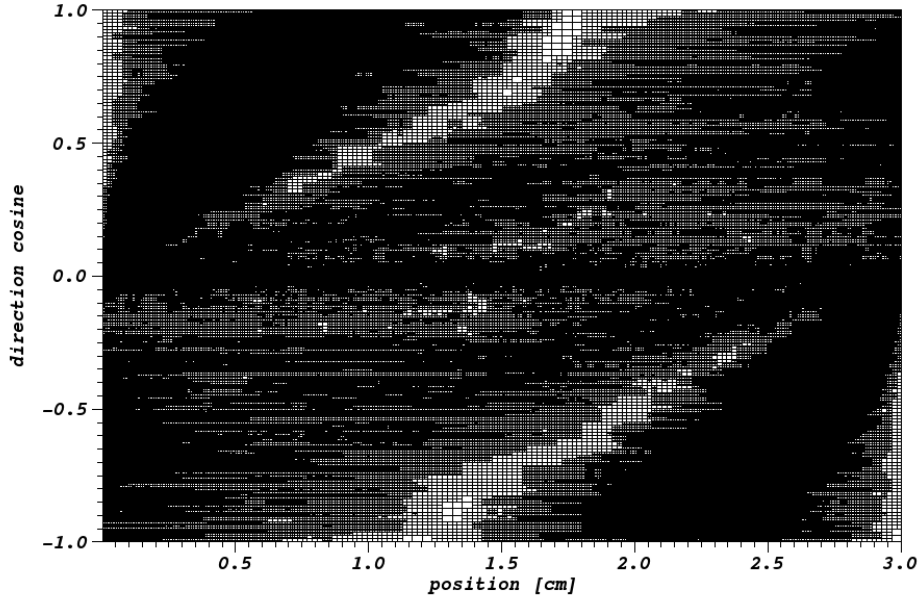

1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]


1.0
0.5
0.0
-0.5
-1.0
direction cosine
0.5
1.0
1.5
2.0
2.5
3.0
position [cm]