

Matrix Method for Modeling Fluence Distributions in Anisotropic Scattering Media

by

Thomas James McIntyre

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Biomedical Engineering

Department of Electrical and Computer Engineering  
University of Alberta

© Thomas James McIntyre, 2014

## **Abstract**

Determining the optical fluence from any source in highly scattering tissue can be a time consuming process. Previous methods have either been shown to be costly in terms of time, or produce erroneous results. Herein, the Matrix Method is proposed. This method attempts to use matrices to propagate the radiance through a highly scattering medium, returning the fluence distribution. This method is compared to the gold standard in photon transport, the Monte Carlo Method. The isotropic point source and the pencil beam simulations are used as tests. It is shown that the results from Matrix Method are similar to Monte Carlo far from the source, and that the times required for it to run are short. As well, no stochastic errors common to Monte Carlo are present in the Matrix Method.

# Acknowledgements

“What we do now echoes in eternity.”

Marcus Aurelius, Meditations

This thesis would not have been possible without the help of my parents, Tom & Mary McIntyre. I would like to thank the Electrical Engineering graduate students, faculty and staff who made this journey possible. A special thanks goes to the Graduate Students Association who lent me support when I needed it most. The memories I have from being a councillor will always be with me, as will the dinners (thanks Ellen – they were delicious!).

# Table of Contents

<b>1 Introduction</b> .....	<b>1</b>
1.1 Light Transport in Turbid Media .....	2
1.2 The Cell .....	2
1.3 Mie Scattering Theory .....	3
1.4 Near-Infrared Light .....	4
1.5 Project Motivation .....	5
1.6 Thesis Overview .....	6
<b>2 Background and Literature Review</b> .....	<b>7</b>
2.1 Introduction to Light Propagation in Scattering Media .....	8
2.2 Definitions .....	8
2.2.1 Fluence Rate and Fluence .....	8
2.2.2 Radiance .....	9
2.2.3 Absorption Coefficient .....	9
2.2.4 Scattering Coefficient .....	10
2.2.5 Extinction Coefficient .....	10

2.2.6 Anisotropic Scattering .....	10
2.2.7 Anisotropy .....	11
2.2.8 Transport Scattering Coefficient .....	12
2.3 Radiative Transfer Equation .....	12
2.4 Diffusion Theory .....	13
2.4.1 Diffusion Equation .....	13
2.5 Methods for Light Propagation in Tissue .....	14
2.6 Solving the Full Radiative Transfer Equation .....	15
2.7 Diffusion Approximation and Green's Function .....	15
2.8 Computational Methods for RTE and Diffusion .....	17
2.9 P3 Approximations .....	19
2.10 Monte Carlo .....	20
2.10.1 Monte Carlo Method .....	20
2.11 Hybrid Method of Monte Carlo Diffusion .....	23
2.12 Discrete Ordinates .....	24
<b>3 Matrix Method Theory .....</b>	<b>25</b>
3.1 How the Matrix Method is Equivalent to Monte Carlo .....	26
3.2 Matrix Method Fundamentals .....	26
3.2.1 Grid Creation .....	26

3.2.2 Approximations .....	27
3.3 Projection Step of Matrix Method .....	28
3.3.1 Radiance .....	28
3.3.2 Projection Matrix .....	29
3.3.3 Projection Elements Derivation .....	30
3.4 Limit Analysis of Projection Elements .....	34
3.4.1 Scattering Limit Analysis .....	34
3.4.2 Projection Limit Analysis .....	36
3.5 Reducing the Number of Coefficients Due to Beer's Law .....	37
3.6 Scattering Step of Matrix Method .....	38
3.6.1 Scattering Matrix Elements .....	39
3.7 Overall Matrix Equation .....	42
3.8 Toeplitz Matrices .....	42
3.9 Projection with Convolution .....	44
3.10 Absorption .....	45
3.11 Boundaries .....	46
3.11.1 Index-Matched Boundaries .....	47
3.11.2 Index-Mismatched Boundaries .....	47
3.12 Addressing Boundaries in Pencil Beam and Point Source.....	48
3.13 Memory .....	49

3.14 Computational Complexity .....	50
3.14.1 Scattering Operations .....	50
3.14.2 Projection Operations .....	50
3.14.3 Overall Operations .....	51
3.15 Potential for Error Reduction .....	51
3.15.1 Fluence Analysis .....	52
3.15.2 Time & Error Analysis .....	53
3.15.3 Error Comparison .....	55
3.16 Overall Block Diagram .....	55
<b>4 Matrix Method Results .....</b>	<b>59</b>
4.1 Introduction .....	60
4.1.1 Mean Square Error .....	60
4.1.2 Mean Absolute Percent Error .....	61
4.1.3 Time Analysis .....	61
4.1.4 Simulation Parameters .....	62
4.1.5 Number of Solid Angle Segments .....	64
4.2 2D Isotropic Point Source Results .....	65
4.2.1 Monte Carlo Results .....	65
4.2.2 Matrix Method Results .....	66

4.2.3 Cross Section Results .....	68
4.2.4 MAPE & MSE Results .....	73
4.2.5 MAPE Plots .....	75
4.3 2D Isotropic Point Source Results .....	77
4.3.1 Monte Carlo Results .....	77
4.3.2 Matrix Method Results .....	78
4.3.3 Cross Section Results .....	80
4.3.4 MAPE & MSE Results .....	88
4.3.5 MAPE Plots .....	89
4.4 Possible Matrix Method Errors .....	91
4.4.1 Bottom Boundary Cross Section per Iteration .....	91
4.4.2 Small Masks .....	93
4.4.3 Number of Solid Angle Segments .....	93
4.4.4 Number of Iterations .....	93
4.4.5 Scattering Location .....	94
4.4.6 Multiple Direction Averaging .....	95
4.4.7 Pixel Size .....	96
4.4.9 Diagonal Pixels .....	97
4.5 Clover, Bow Tie and Cross Discussion .....	98
4.5.1 Point Source: Source Correction Factor .....	99



4.6 Time Analysis .....	100
4.6.1 Simulation Time .....	100
<b>5 Conclusions .....</b>	<b>102</b>
5.1 Chapter Summary .....	103
5.2 Future Directions .....	104
<b>References .....</b>	<b>105</b>
<b>Appendix .....</b>	<b>120</b>

# List of Tables

3.1	Sizes of the convolution mask determined by using an extinction coefficient of $100 \text{ cm}^{-1}$ and a grid with square elements of size of $50 \mu\text{m}$ by $50 \mu\text{m}$ . The threshold intensity is the ratio of final intensity to initial intensity. ....	44
4.1	Sample times taken for the simulations to run. Here, the programs were both written so they could benefit from MATLAB parallel processing toolbox, which can use multiple processors to run simulations. ....	99

# List of Figures

1.1	The animal cell with labeled organelles [15].	3
1.2	Mie Theory prediction of scattering efficiency from a plane wave and a spherical particle [20].	4
1.3	The optical window in biological tissue [21].	5
2.1	A 3D solid angle segment $\Omega$ encompassing the subsection A (the dark blue circle).	9
2.2	Examples of a finite element represented with a triangle. The values within the triangle can be represented by the sum of the values at each node, scaled by a factor proportional to their distance from a point of interest [45].	18
2.3	Example simulation medium using both MC and Diffusion, beginning with an incident beam. Here, the final step (outlined with $l'_t$ ) converts the photon to a point source [75].	23
3.1	Examples of photon scattering in a typical simulation using a) MC and b) MM. The blue arrows represent inward photons while the red arrows represent scattered photons.	27
3.2	The discrete representation of the radiance with 8 different solid angle segments, each 45 degrees wide.	27
3.3	A pictorial representation of the propagation of radiance from pixel I, solid angle segment L to pixel F, solid angle segment K using the propagation matrix element $p_{if}^{lk}$ .	28

3.4	The radiance ( $W/m^2 \cdot sr$ ) column vector $\underline{b}_{\{ji\}}$ , used in the projection step. It is sorted such that the index of each element is given by $i + M * (j - 1)$ , where $i$ is the pixel and $j$ is the solid angle segment. ....	29
3.5	The projection matrix with M pixels and J solid angle segments. ....	29
3.6	The projection matrix accounting for no propagation to different solid angle segments. ....	30
3.7	A plot of the interpolation function. This function is used as a linear fit for the radiance, and will be represented by the weighing function $w(\theta, \theta_j)$ . Here index j represent any single solid angle segment from 1 to J. ....	31
3.8	An example of a green ray projected through a medium and through a specific pixel. Rays 1 and 2 – black, centered at $\theta_1$ and $\theta_2$ – are the boundaries of the solid angle segment. For the destination pixel, $r_i, r_f(\theta)$ , and $d(\theta)$ are illustrated. Here, one can see the angular span for each value of $p_{if}^k$ will change, depending on the pixel. In the red pixel, although the solid angle segment spans from $\theta_1$ to $\theta_2$ , the span from $\theta_1$ to $\theta_i$ is used. $\theta_i$ is the angle which intersects the upper left corner, and is outlined as a single dotted red ray. ....	33
3.9	An example of the projection step acting on $\underline{b}_{\{ji\}}$ . The index r is used as an indicator. ....	37
3.10	This is the radiance column vector $\underline{b}_{\{ij\}}$ , used in the scattering step. It is sorted such that the index of each element is given by $j + J * (i - 1)$ , where $i$ is the pixel and $j$ is the solid angle segment. ....	38
3.11	The (blue) PDF and (red) modified PDF of the Henyey-Greenstein function. ....	39
3.12	The scattering matrix which operates on $\underline{b}_{\{ij\}}$ . ....	41

3.13	Transition from a typical sub matrix to one which is only dependent on distance. ....	42
3.14	Projection sub matrix within a homogeneous medium. This is an example of a Toeplitz matrix. Each row only has non-zero values in the entries ranging from $p_{t}^k$ to $p_{t}^k$ . These values represent the threshold where any radiance propagating past it can be considered negligible to the simulation. ....	43
3.15	Examples of a 51 x 51 convolution mask. Here, the mask covers the 8 <sup>th</sup> solid angle segment out of 60 total solid angle segments. This corresponds to a central angle of 42°. The scale is the log10 values of $p_{if}^k$ . ....	45
3.16	An example of a boundary crossing. The red dots with the overlaid radiance segments represent the normal neighbouring pixels while the green dots are the boundary points. Each of these green dots has the same number of solid angle segments. ....	46
3.17	Block diagram for Declare Variables section. ....	56
3.18	Block diagram for Create Projection Masks section. ....	57
3.19	Block diagram for Create Scattering Matrix section. ....	58
3.20	Block diagram for Radiance Propagation section. ....	58
4.1	The Mean Square Error between the modified HG function and it's linearly interpolated version. ....	62
4.2	The modified HG function and it's linearly interpolated version over 360°. Insert ranges from -15° to 15°. ....	63

4.3 The fluence ( $J/m^2$ ) distribution result of the MC simulation with the isotropic point source in an infinite turbid media. Results are in logarithmic (base 10) scale. ....65

4.4 The fluence distributions from MM involving an isotropic point source in an infinite turbid media. These are displayed logarithmically (base 10) to ease comparisons. From top left to bottom right we have 100, 200, 300 and 400 iterations. The axes outline the distances of the medium, in cm. ....66

4.5 The fluence distributions from MM involving an isotropic point source in an infinite turbid media. These are displayed logarithmically (base 10) to ease comparisons. From top left to bottom right we have 500, 1000, 2000 and 3000 iterations. The axes outline the distances of the medium, in cm. ....67

4.6 Numbered cross sections used in analysis. The axes outline the distance, in cm. ....68

4.7 The vertical cross section of the fluence through the source term. Blue is MM and green is MC. The upper plot represents the entire range, while the lower plot is the region within 1 mm of source term. ....69

4.8 The diagonal cross section of the fluence through the source term. Blue is MM and green is MC. The upper plot represents the entire range, while the lower plot is the region within 1 mm of source term. ....70

4.9 The cross section midway between the source and the edge. Blue is MM and green is MC. The upper plot represents the entire range, while the lower plot is the region within 1 mm of the cross sections' center. ....71

4.10 The cross section of the edge of the medium. Blue is MM and green is MC. The upper plot represents the entire range, while the lower plot is the region within 1 mm of the cross sections' center. ....72

4.11 MSE (blue) and MAPE (green) of the MM isotropic point source simulation with respect to MC. The insert ranges from 1000 to 3000 iterations. ....74

4.12 The absolute percent error in the isotropic point source between MM and MC. ....75

4.13 The absolute percent error in the isotropic point source between MM and MC. Errors larger than 2% are shown. A scale beginning at 2 % (black) to 4 % (red) and to 6 % (white) is included for better contrast. ....76

4.14 The fluence ( $J/m^2$ ) distribution result of the MC simulation results of MC simulation of a pencil beam incident on a surface of a semi-infinite turbid media. Results are in logarithmic (base 10) scale. ....77

4.15 The fluence distributions from MM involving a pencil beam source incident to a semi-infinite turbid media. These are displayed logarithmically (base 10) to ease comparisons. From top left to bottom right we have 100, 200, 300 and 400 iterations. The axes outline the distances of the medium, in cm. ....78

4.16 The fluence distributions from MM involving a Pencil beam source incident to a semi-infinite turbid media. These are displayed logarithmically (base 10) to ease comparisons. From top left to bottom right we have 500, 1000, 2000 and 3000 iterations. The axes outline the distances of the medium, in cm. ....79

4.17 Numbered cross sections used in analysis. The axes show the distance, in cm. ....80

4.18 The cross section along the center of the medium, through the source. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region up to 2 mm from the source. ....81

4.19 The cross section along the 45° diagonal through the source to the edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region up to 2 mm from the source. ....82

4.20 Horizontal cross section midway between the source and the upper edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region within 1 mm of the center. ....83

4.21 The bottom edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region within 1 mm of the center. ....84

4.22 The side edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region within 1 mm of the largest value along this edge. ....85

4.23 The upper edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region within 1 mm of the largest value along this edge. ....86

4.24 MSE (blue) and MAPE (green) of the MM pencil beam simulation. ....88

4.25 The absolute percent error in the pencil beam simulation in a semi-infinite turbid media between MM and MC. ....89

4.26 The absolute percent error in the isotropic point source between MM and MC. Errors larger than 2% are shown. A scale beginning at 2 % (black) to 4 % (red) and to 6 % (white) is included for better contrast. ....90



4.27 Pencil Beam. Horizontal cross section of the lower boundary for several iterations. The upper plot shows the percent error after the (cyan) first, (red) second, (green) third and (blue) fourth iteration. The lower plot shows the percent error after the (blue) fourth, (green) fifth, (red) tenth, (cyan) fifteenth and (black) 3000th iteration. ....92

4.28 The location of mean absorbance and scattering within a pixel from (blue) a linear model and (green) Beer’s Law. The location of mean absorbance and scattering using weighted average was found to be 20  $\mu\text{m}$  with linear and 18.7  $\mu\text{m}$  with Beer’s Law within a pixel 20  $\mu\text{m}$  long. ....94

4.29 Above is 1) initial first order radiance, 2) propagation of first order radiance, 3) outline of the location of mean scattering within the initial pixel (red) and 4) second order radiance. In MM second order radiance is propagated from the center as opposed to the location of mean scattering. ....97

# Chapter 1

## Introduction

This study was conducted to design, develop and evaluate a novel method of predicting the fluence distribution from a light source incident or within anisotropic scattering media. Methods which can be used with multiple types of sources, address multiple directions, yield low standard errors in the results and short simulation time is important in several fields. These include Photodynamic therapy dosimetry [1, 2], imaging inverse problems [3, 4] and sensing of optical properties [5, 6]. This thesis will focus on human tissue as a medium with red and near-infrared (NIR) light as the source wavelengths. The projects scope involves analyzing current methods of light propagation in turbid media, developing the theoretical foundations for the novel method, and comparing its results and computational efficiency to Monte Carlo.

## 1.1 Light Transport in Turbid Media

Light transport in turbid media is a field of research which involves analyzing the fluence from a light source within or impinging upon a scattering media [7]. Tissues are media which scatter light to a high degree whenever it attempts to propagate through it [8]. Modelling the light distribution within these tissues is critical as it would allow for the improved efficiency of dose delivery for brachytherapy and other related treatments [9]. One of the major reasons for the high degree of scattering comes from the internal structure of the cell.

## 1.2 The Cell

When photons scatter, the photon changes its direction due to an elastic collision with some object or molecule within a medium [10]. The cell has a number of different organelles within it (Fig. 1.1), however only some of these contribute to photon scattering. The organelles which significantly contribute to photon scattering are the mitochondria and the cell nucleus [11]. The main reason these two organelles contribute is they are large relative to other organelles.

Other organelles, such as ribosomes, endoplasmic reticulum and centrioles, simply aren't big enough to induce the significant proportion of scattering which is seen with the mitochondria and nucleus. Although higher energy particles (i.e. electrons) can be scattered by these smaller organelles – yielding higher resolution [12], this thesis will focus on photons from the red and NIR range. Photons are not only scattered by tissue, they are also absorbed within it. Some of the largest absorbers of red and NIR light in human tissue include melanin, oxy- and deoxyhemoglobin [13]. As photons are absorbed, their energy is converted to heat [14].

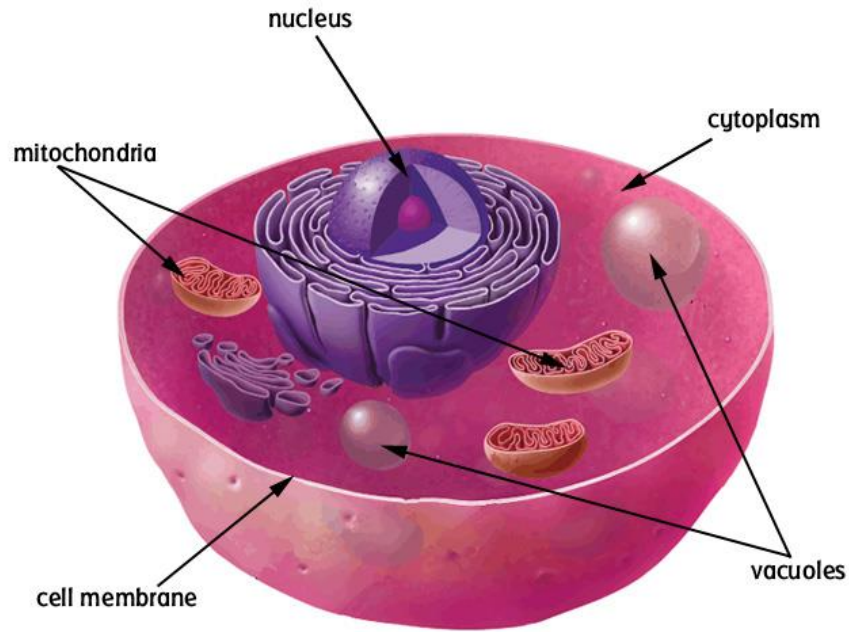


Figure 1.1. The animal cell with labeled organelles [15].

### 1.3 Mie Scattering Theory

The theory behind why a cell nucleus and mitochondria can scatter these photons, while other organelles cannot, has been developed in Mie scattering theory. Mie scattering theory predicts that an object's scattering efficiency can be related to a size parameter (eq. 1-1) [16, 17].

$$Size\ Parameter = \frac{2\pi nr}{\lambda} \tag{1-1}$$

The size parameter depends on the radius of the object  $r$ , the index of refraction of the medium  $n$ , and the wavelength of the incident light source  $\lambda$ . The scattering efficiency is the ratio of the effective scattering cross section to the object's geometric cross section. It is very low when the radius of the object is smaller than the photon's wavelength (Fig. 1.2). As the object's radius increases, the scattering efficiency dramatically increases, and eventually levels off at a value of 2 (Fig. 1.2).

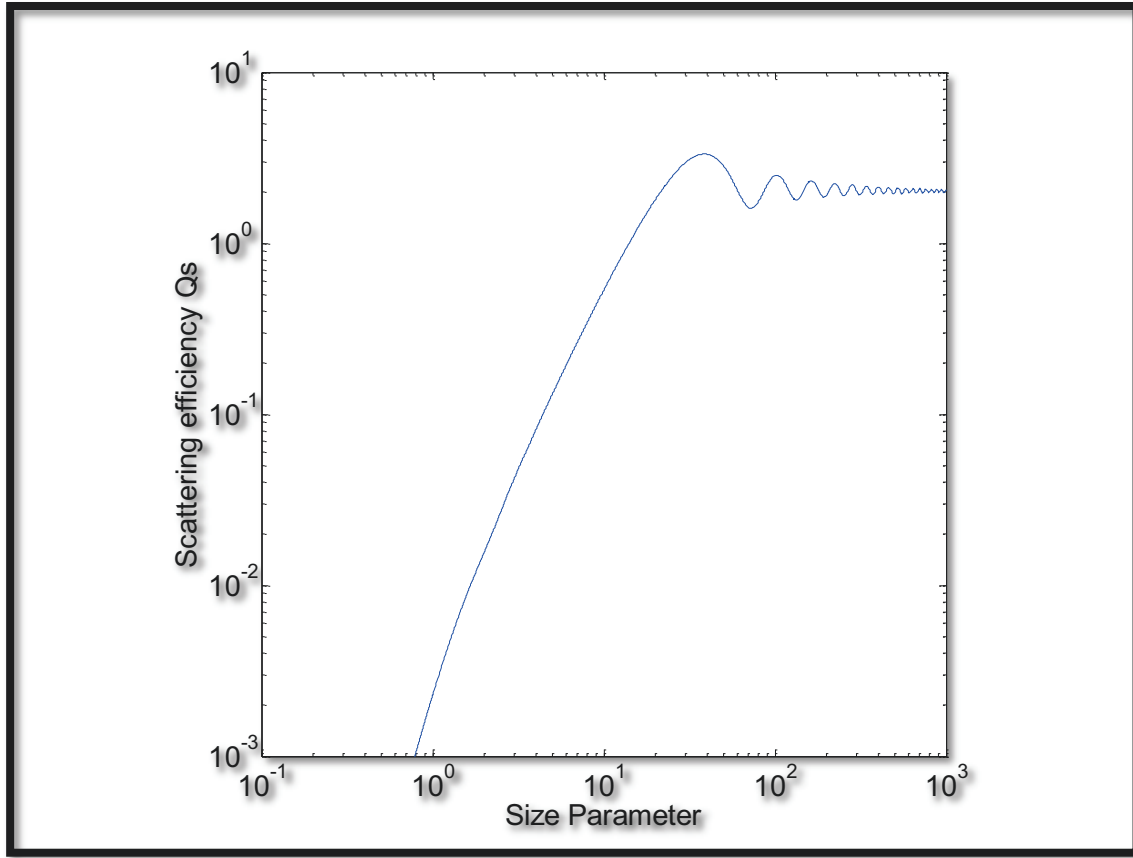


Figure 1.2. Mie Theory prediction of scattering efficiency from a plane wave and a spherical particle [20].

The size parameter for ribosomes, endoplasmic reticulum and centrioles is approximately 0.5 for red and NIR light; these organelles are too small to cause significant scattering. The mitochondria and nucleus, whose diameter are 1-10  $\mu\text{m}$  however [18, 19], would have a size parameter between 5 and 75. This leads to significant scattering of red and NIR photons.

## 1.4 Near-Infrared Light

As previously mentioned, red and NIR photons will be used in these simulations since photon penetration depth in human tissue is greater within this wavelength range [21]. This is largely due to the absorption of biological absorbers commonly found in human tissues (Fig. 1.3) [21, 22].

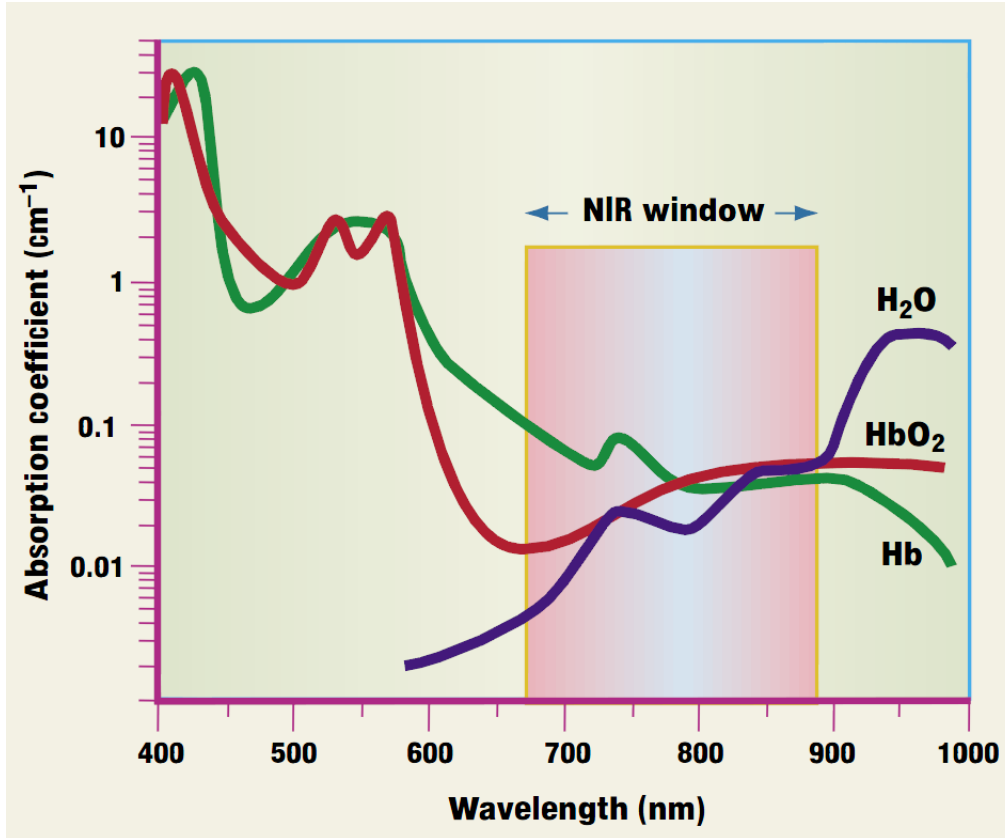


Figure 1.3. The optical window in biological tissue [21].

Here, hemoglobin is shown to absorb photons with wavelengths less than 600 nm, while water absorbs photons with wavelengths greater than 1050 nm. As such, an optical window exists between 650 nm and 900 nm where photons can penetrate through tissue [21 – 23]. The significant interest in this photon range is due to the optical window.

## 1.5 Project Motivation

Although the forward problem has been well examined (Chapter 2), the inverse problem of determining tissue parameters has not. Currently, this is found using the inverse of a Jacobian Matrix [24]. This inverse is ill-conditioned however, with small errors leading to large deviations. A simulation method with very small errors would be useful in the inverse problem.

## 1.6 Thesis Overview

This thesis is written to evaluate a novel method which uses convolution masks and matrices to model photon propagation in highly scattering tissue. Chapter 2 focusses on the background of light propagation in anisotropic scattering media and encompasses a literature review of the field. It includes an explanation of the parameters used in fluence modelling and a summary of other modelling methods. Briefly, the main methods are Diffusion Theory and Monte Carlo. Diffusion Theory changes anisotropic scattering to isotropic scattering, simplifying the computations but being inaccurate near sources. Monte Carlo uses photon packets to stochastically model fluence. It is accurate near sources but has large standard errors far away. Hybrids of these two have been made, which decrease computation time and increase accuracy.

In Chapter 3, the Matrix Method is developed from fundamentals and its process is explained. The Matrix Method calculates photon propagation as an ensemble of all photons in each iteration. By combining them as one set, it is expected that no stochastic issues common to Monte Carlo should be present. The Matrix Method includes direction with all of its calculation by using the radiance. This is in contrast to Diffusion Theory, which calculates the fluence directly. Chapter 4 shows the results of two different 2D simulations, one of a pencil beam source in an index-matched semi-infinite turbid media while the other an isotropic point source in an infinite turbid media. Both of these media are not infinite due to memory limitations, and so are anisotropic turbid media within an index matched non-scattering highly absorbing ambient media. The results are then compared and contrasted with Monte Carlo in the same conditions.

## Chapter 2

# Background and Literature Review

The field of light transport in turbid media has been examined by numerous authors, with many different types of simulations being developed to accurately predict the fluence distribution from a source. The nomenclature, commonly used equations and expressions will be listed and discussed. More common terms, including fluence ( $J/m^2$ ), radiance ( $W/(m^2 \cdot sr)$ ), anisotropy, scattering and absorption coefficient ( $cm^{-1}$ ) will be defined. In this field there are several methods which are much more common in current literature, and they will be examined and discussed. These methods include the Monte Carlo Method (MC), the Diffusion Approximation and Green's Function, the Hybrid version of Monte Carlo and Diffusion Theory, the Radiative Transport Equation and Discrete Ordinance.



## 2.1 Introduction to Light Propagation in Scattering Media

Light propagation in anisotropically scattering media has been a research field which has had significant interest in the last several years [56 – 91]. Although simulation methods may differ fundamentally, each method relies on a series of fundamental facts associated with light transport. These facts form the basis for the field of light propagation in scattering tissue.

## 2.2 Definitions

### 2.2.1 Fluence Rate and Fluence

Fluence rate ( $W/m^2$ ) is “the energy flow per unit area per unit time, regardless of the flow direction” [25]. In this thesis, the term energy from the definition of fluence rate will refer to the photons’ energy. An absorbing substance will absorb photons regardless of their direction [26], so the fluence rate can be used to determine the absorbance.

$$\textit{Fluence rate} = \Phi(\vec{r}, t) \quad (W/m^2) \quad (2-1)$$

Here,  $\vec{r}$  denotes position and  $t$  denotes time. One can derive the fluence ( $J/m^2$ ) from the fluence rate. Fluence is “the time-integrated fluence rate” [25]. It is the total energy per unit area. Fluence is the value calculated by each simulation – it’s also directionless. It depends solely on the total number of photons which arrive at a location.

$$\textit{Fluence} = F(\vec{r}) = \int_{-\infty}^{\infty} \Phi(\vec{r}, t) dt \quad (J/m^2) \quad (2-2)$$

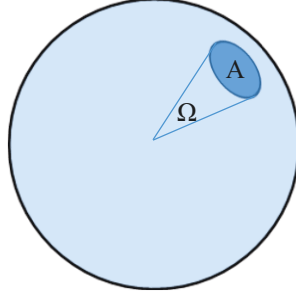


Figure 2.1. A 3D solid angle segment  $\Omega$  encompassing the subsection A (the dark blue circle).

## 2.2.2 Radiance

Radiance ( $W/(m^2 \cdot sr)$ ) is effectively the fluence rate per unit solid angle  $\hat{s}$  [25, 26].

$$Radiance = L(\vec{r}, \hat{s}, t) \quad (W/(m^2 \cdot sr)) \quad (2-3)$$

$$\Phi(\vec{r}, t) = \int_{4\pi} L(\vec{r}, \hat{s}, t) d\hat{s} \quad (W/m^2) \quad (2-4)$$

A solid angle segment  $\hat{s}$  is simply the angle which covers a subsection of the area (3D) or circumference (2D) from the objects center [27]. Although the definition of a solid angle can extend to many geometric shapes, this thesis will only consider circles and spheres.

## 2.2.3 Absorption Coefficient

The absorption coefficient ( $cm^{-1}$ ) is “the probability of photon absorption in a medium per unit path length” [28]. This coefficient is important for heat production and fluorescence [29].

$$Absorption\ coefficient = \mu_a \quad (cm^{-1}) \quad (2-5)$$

## 2.2.4 Scattering Coefficient

Similar to the absorption coefficient, the scattering coefficient ( $cm^{-1}$ ) is “the probability of a photon scattering in a medium per unit path length” [28]. In tissue, a photon is likely to scatter many times prior to being absorbed [8], making it an important coefficient.

$$\textit{Scattering coefficient} = \mu_s \quad (cm^{-1}) \quad (2-6)$$

## 2.2.5 Extinction Coefficient

Typically the scattering and absorption coefficients are combined. This value, the extinction coefficient ( $cm^{-1}$ ), represents the probability of a photon either being absorbed or scattered in a medium per unit path length [28].

$$\textit{Extinction coefficient} = \mu_t = \mu_s + \mu_a \quad (cm^{-1}) \quad (2-7)$$

Its inverse is the mean free path ( $cm$ ) between interaction events for the photons [28]

$$l_t = \frac{1}{\mu_t} \quad (cm) \quad (2-8)$$

## 2.2.6 Anisotropic Scattering

Although the scattering coefficient is important in determining the likelihood of scattering, it gives no insight into how a photon scatters. It has been shown that the Henyey-Greenstein probability distribution can model the change in a photon’s propagation after scattering [30]. Other distributions, like  $\delta$ -Eddington, can also be used to model photon scattering [31].

$$P_{HG}(\theta) = \frac{1}{2} \frac{(1 - g^2) \sin \theta}{(1 + g^2 - 2g \cos \theta)^{3/2}} \quad (2-9)$$

Here, the angle  $\theta$  represents the angular deflection due to the photon's scattering. Scattering can occur isotropically or anisotropically [32]. The coefficient  $g$  is the anisotropy, described in section 2.2.7.

The  $\delta$ -Eddington distribution involves separating the scattered photons into two groups, one which is forward projected and another which can be scattered isotropically or in a linearly anisotropic manner [31, 33]. The weighing factor  $f$  is between 0 and 1 [31], and represents the fraction of photons which are forward projected. This linear anisotropic scattering  $\delta$ -Eddington distribution is shown in equation 2-10 [33]

$$P_{\delta-E}(\theta) = \left(\frac{1}{4\pi}\right) [2f\delta(\theta) + (1 - f)(1 + 3g \cos \theta)] \sin \theta \quad (2-10)$$

The isotropically scattering  $\delta$ -Eddington distribution is shown in equation 2-11 [33].

$$P_{\delta-E}(\theta) = \left(\frac{1}{4\pi}\right) [2f\delta(\theta) + (1 - f)] \sin \theta \quad (2-11)$$

## 2.2.7 Anisotropy

The anisotropy  $g$  gives the probability distribution its characteristic shape. Here,  $g$  is “defined as  $\langle \cos \theta \rangle$ , has a value between -1 and 1” [32]. If  $g$  is equal to zero, scattering is isotropic [32]. Other research shows setting  $g$  to be 0.9 is appropriate in biological tissues, as it models a very forward directed probability distribution [32].

## 2.2.8 Transport Scattering Coefficient

Although the scattering coefficient has been defined, another value allows the anisotropy to be addressed without using a distribution (i.e. isotropically). Here, the “transport scattering coefficient” ( $cm^{-1}$ ) is defined as [34]

$$\mu'_s = \mu_s(1 - g) \quad (cm^{-1}) \quad (2-12)$$

From this the “transport interaction coefficient” ( $cm^{-1}$ ) is [34]

$$\mu'_t = \mu_a + \mu'_s \quad (cm^{-1}) \quad (2-13)$$

With the transport mean free path ( $cm$ ) as the inverse of the transport interaction coefficient [34]

$$l'_t = \frac{1}{\mu'_t} \quad (cm) \quad (2-14)$$

## 2.3 Radiative Transfer Equation

The analytical way to model photon transport in anisotropically scattering media is through the radiative transfer equation (RTE) [25]. Here, several contributions, including conservation of energy, are taken into account [35]. These contributions include the beam’s divergence, energy loss due to absorption and scattering, other photons scattered from one direction into the same direction as the beam as well as the initial source of photons. The RTE (eq. 2-15) is a complex equation and is derived elsewhere [35].

$$\begin{aligned} \frac{\partial L(\vec{r}, \hat{s}, t)}{c \partial t} &= -\hat{s} \cdot \nabla L(\vec{r}, \hat{s}, t) - \mu_t L(\vec{r}, \hat{s}, t) \\ &+ \mu_s \int_{4\pi} L(\vec{r}, \hat{s}', t) P(\hat{s}' \cdot \hat{s}) d\Omega' + S(\vec{r}, \hat{s}, t) \end{aligned} \quad (2-15)$$

The term  $P(\hat{s}' \cdot \hat{s})$  ( $sr^{-1}$ ) represents the probability that a photon propagating in any direction  $\hat{s}'$  scatters into direction  $\hat{s}$  [35]. The integral surrounding it accounts for all initial propagation directions  $\hat{s}'$ . The divergence of the radiance into or out of the solid angle element is accounted for with the gradient. The source term in the RTE is  $S(\vec{r}, \hat{s}, t)$ .

## 2.4 Diffusion Theory

The RTE is very difficult to solve, and so other techniques and simplifications have been introduced. One of these simplifications is the diffusion approximation. Here, after enough scattering has occurred the radiance is assumed to be nearly isotropic [28]. This assumption is expected to hold in high scattering mediums ( $\mu'_s \gg \mu_a$ ) [28].

### 2.4.1 Diffusion Equation

By making this assumption it can be shown that the RTE can be rearranged such that the expression depends on fluence rate ( $W/m^2$ ) instead of radiance ( $W/(m^2 \cdot sr)$ ). A full derivation can be found in other literature [34, 35], yielding the diffusion equation (eq. 2-16) [34]

$$\frac{\partial \Phi(\vec{r}, t)}{c \partial t} + \mu_a \Phi(\vec{r}, t) - \nabla \cdot [D \nabla \Phi(\vec{r}, t)] = S(\vec{r}, t) \quad (2-16)$$

Where the source is assumed to be isotropic and  $D$  is the diffusion coefficient ( $cm$ ), given by [34]

$$D = \frac{1}{3(\mu_a + \mu'_s)} \quad (cm) \quad (2-17)$$

In the case where the diffusion coefficient represents a homogeneous medium (i.e. its space invariant) one can further simplify the diffusion equation [34], resulting in

$$\frac{\partial \Phi(\vec{r}, t)}{c \partial t} + \mu_a \Phi(\vec{r}, t) - D \nabla^2 \Phi(\vec{r}, t) = S(\vec{r}, t) \quad (2-18)$$

One of the key features of the diffusion equation is that it depends on the combination of  $\mu_s$  and  $g$ , but not on either of them individually. This is the similarity relation [34], which means the medium can be treated as an isotropically scattering medium with scattering coefficient  $\mu'_s$ .

The diffusion equation can be evaluated using techniques such as finite element and finite difference time domain [36]. It also has only four degrees of freedom, being position and time, making it simpler to evaluate than the RTE, which has 6 [34].

## 2.5 Methods for Light Propagation in Tissue

Many different methods have been developed attempting to evaluate the fluence within anisotropic scattering tissue. Each of them has benefits and drawbacks, and they will be outlined in sections 2.6 to 2.13. This section of the chapter will begin with solving the initial radiative transfer equation.

## 2.6 Solving the Full Radiative Transfer Equation

This is a method which is rarely used, mainly due to its complexity. The RTE has 6 degrees of freedom, which are position (x, y and z), time (t), and direction ( $\theta, \phi$ ) [34]. In this case, only very simple problems can be solved [37 – 39]. As an example, the case involving a time-independent response of a non-scattering media yields a new differential equation

$$\hat{s} \cdot \nabla L(\vec{r}, \hat{s}) = -\mu_a L(\vec{r}, \hat{s}) + S(\vec{r}, \hat{s}) \quad (2-19)$$

Which is considerably simpler to solve. Here, depending on the source term, techniques of partial differential equations can be used to solve the simplified equation.

In recent work [40], the radiative transfer equation has been solved for the case of the semi-infinite medium. Here, the authors separated the problem into three sections, involving the ballistic, homogeneous and particular portions.

## 2.7 Diffusion Approximation and Green's Function

Outside of very simple situations the RTE is difficult to solve without numerical methods. One common exception occurs when the diffusion approximation can be used. Depending on the source term, analytical solutions may exist. With an infinitely short-pulsed point source  $S(\vec{r}, t) = \delta(\vec{r}, t)$ , the solution is [34]

$$\Phi(\vec{r}, t) = \frac{c}{(4\pi Dct)^{3/2}} e^{(-\frac{r^2}{4Dct} - \mu_a ct)} \quad (W/m^2) \quad (2-20)$$

In the case of a time-independent point source  $S(\vec{r}) = \delta(\vec{r})$ , [34]



$$\Phi(\vec{r}) = \frac{1}{4\pi D r} e^{(-\sqrt{3\mu_a(\mu_a+\mu'_s)} r)} \quad (W/m^2) \quad (2-21)$$

Each of these two equations are the impulse response within an infinite homogenous scattering medium. An impulse response is also referred to as a Green function [34]. The benefit of modeling an impulse response is it can be used to model other sources by convolving the results of the impulse response so it matches the other source.

It can be shown that only a minor change occurs when translating the source terms to a new location  $\vec{r}'$  and initial time  $t'$ . The expression for the impulse responses are simply change from being dependent on  $r$  and  $t$  to  $|r - \vec{r}'|$  and  $(t - t')$  [34]. Essentially the solution stays constant, regardless of where the source or the detector are located. This is the principle of reciprocity [34].

The diffusion equation has the strength of determining analytical answers to problems in homogeneous medium. It can also be used to create differential equations when boundary conditions are available, and these equations can be solved with numerical means. One weakness of diffusion is that it requires the radiance to be isotropic after a number of scattering events [34]. This is not the case near the initial source, as the number of scattering events near to it is typically small. As such, it is accurate further away from the source than closer to it [34].

There have been recent attempts to address the errors from diffusion theory near the point of entry of a pencil beam incident on a semi-infinite turbid media. Vitkin et. al. [41] derived the phase function corrected diffusion approximation, which changes the anisotropic scattering into a delta-isotropic part and an anisotropic part. Substituting this into the RTE yields two terms, and their superposition was shown to yield results comparable to Monte Carlo.

The phase function corrected diffusion approximation was built upon by RJ Zemp [42]. Here, the author extended the solution found by Vitkin et. al. to include obliquely incident pencil beams. Simulations were able to show strong agreement in low absorption media with both normally incident and oblique incident pencil beams.

The diffusion approximation is typically used with other methods. They are more accurate closer to the source and boundaries, but take considerably longer to do computations far from a source – one example would be Monte Carlo simulations [43]. Both the RTE and diffusion approximation are normally too difficult to solve analytically, but computational methods exist to find solutions.

## 2.8 Computational Methods for RTE and Diffusion

In simulations where the mediums are complex, it is nearly impossible to generate an analytical solution to the RTE or the diffusion approximation. To solve these, techniques for solving partial differential equations will be used. Common techniques include the finite difference time domain and the finite element methods [36]. Each of these methods solve the differential equation within a medium. The finite difference time domain method is a technique that uses the Taylor series expansion around a point to replace a derivative [44]. Examining a function  $f(x)$ , one can find its value at  $(x_o + \Delta x)$  by generating a Taylor expansion around  $x_o$ .

$$\begin{aligned} f(x_o + \Delta x) = & f(x_o) + \Delta x f'(x_o) + \frac{(\Delta x)^2}{2} f''(x_o) \\ & + \frac{(\Delta x)^3}{3!} f'''(x_o) + \dots \end{aligned} \tag{2-22}$$

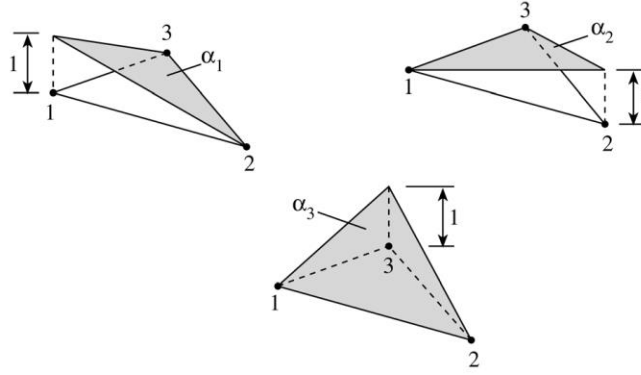


Figure 2.2. Examples of a finite element represented with a triangle. The values within the triangle can be represented by the sum of the values at each node, scaled by a factor proportional to their distance from a point of interest [45].

And similarly the value around  $(x_o - \Delta x)$  can be found. Subtracting them and solving for the first order derivative yields eq. 2-23. The error is proportional to the size of  $\Delta x$ .  $\Delta x$  can be set to be very small, minimizing the error. These points are uniformly spaced through the medium.

$$f'(x_o) = \frac{(f(x_o + \Delta x) - f(x_o - \Delta x))}{2\Delta x} + \frac{(\Delta x)^2}{3!} f'''(x_o) + \dots \quad (2-23)$$

$$f'(x_o) = \frac{(f(x_o + \Delta x) - f(x_o - \Delta x))}{2\Delta x} + O(\Delta x^2) \quad (2-24)$$

Finite element differs from finite difference time domain in that the results within any element are calculated by the values at the nodes [46]. These nodes outline the boundary of the elements, can be placed anywhere within a medium and their values are found by solving a system of equations based on the governing equations [47]. The finite element method has been examined in several articles [48 – 52]. Other ways have been introduced to solve the RTE, which can include replacing the terms within the RTE by expanding them with spherical harmonics, as described in section 2.9.

## 2.9 P<sub>3</sub> Approximations

The radiance and the source terms are expanded using spherical harmonics, as shown in eq. 2-25 and eq. 2-26.  $Y_{lm}(\hat{S})$  are the spherical harmonics while  $\Phi_{lm}$  and  $\sigma_{lm}$  are the moments of the radiance and the source distributions, respectively [53, 54].

$$L(\vec{r}, \hat{s}, t) = \sum_{l=0}^N \sum_{m=-l}^l \left( \frac{2l+1}{4\pi} \right)^{\frac{1}{2}} \Phi_{lm}(\vec{r}, t) Y_{lm}(\hat{s}) \quad (2-25)$$

$$S(\vec{r}, \hat{s}, t) = \sum_{l=0}^N \sum_{m=-l}^l \left( \frac{2l+1}{4\pi} \right)^{\frac{1}{2}} \sigma_{lm}(\vec{r}, t) Y_{lm}(\hat{s}) \quad (2-26)$$

Equations 2-25 and 2-26 outline the P<sub>N</sub> approximation, where the radiance and the source are calculated with N levels. In the P<sub>3</sub> approximations, the levels are truncated to  $l = 3$  which reduces the number of calculations significantly. After truncation, equations 2-25 and 2-26 can be substituted back into the RTE, and, depending on the medium, the resulting expression can be reduced to a simple equation [53].

An example of the P<sub>3</sub> approximation can be found in D. J. Dickey, et. al. [55]. Here, the authors extended some of their previous work with the P<sub>3</sub> approximation to incorporate it with spherical geometry. Although the plane wave geometry had been solved, it was shown to be impractical for clinical use. An Intralipid/methylene blue phantom was used to test the modification, and the results were shown to be reproducible.

## 2.10 Monte Carlo

MC is a very common technique in the field of anisotropic scattering, due to its capacity to handle a wide array of applications and its programming simplicity [56 – 72]. Although MC does not have an official definition, a simplified version relating to the field of light scattering can be made [73]. Here, MC is a modelling method which uses random variables to simulate multiple photon trajectories. These trajectories are then combined, generating a final result [73].

MC simulations can be used for a wide variety of different types light propagation experiments, common examples being the isotropic point source and the pencil beam. The isotropic point source is modeled as a point source in a scattering homogeneous medium [71]. Simulations with MC have been expanded to include inhomogeneous mediums, however that is outside of the scope of this thesis, and so will only be mentioned. The pencil beam is a case where a beam of photons is incident on a scattering media [72], here perpendicularly to the edge. In each case the source can be represented as a delta function. This means that the medium's response (i.e. final fluence) can be interpreted as the impulse responses or Green's functions [34].

### 2.10.1 Monte Carlo Method

MC, as it applies to photon transport, differs from the other methods in how the solution is found. While methods like Diffusion use the RTE, MC examines the propagation of photon packets as they pass through a medium [72]. As MC methods of photon propagation have been extensively published [56 – 72], this section will simply seek to give an overview of the main feature [32, 73]. For MC simulations, the term photon and photons packets will be used interchangeably with one another.

MC simulations of photon transport begin by defining the photon with a number of parameters. These include, but are not limited to location, direction, weight and step size [32]. The location, direction and step size are recorded to determine where the photon will undergo its next scattering and absorption event. The weight is used to measure how much of the photon will be absorbed upon arrival.

The step size is the distance the photon packet will travel prior to undergoing a scattering and absorption event. It is determined by generating a pseudorandom number  $\varepsilon$  (uniformly distributed between 0 and 1) and using an equation based on the free path [32].

$$s = -\frac{\ln \varepsilon}{\mu_t} \quad (cm) \quad (2-27)$$

In the case where the step size takes a photon to a section with a different index of refraction, the photon arrives at the boundary, the step size is reduced by the distance traveled, and it then undergoes reflection or refraction at that point [74]. The photon is redirected based on these principles, and continues until it reaches another boundary or has traveled the remaining step size [74].

Once the photon has traveled the step size, the photon undergoes a scattering and absorption event [32]. For absorption, the photon's weight (i.e. photon packet's energy) is reduced by a fraction proportional to the photons current weight [32]. This proportion is recorded and the new weight is saved. These values are later used to determine the fluence.

$$W_{new} = W_{old} - \frac{\mu_a}{\mu_t} W_{old} \quad (2-28)$$

Once the absorption has been calculated, the photon undergoes the scattering portion of the event. Here, another pseudorandom number  $\varepsilon$  (uniformly distributed between 0 and 1) is used in conjunction with the Henyey-Greenstein probability distribution [32]. This distribution will be discussed in more depth in Chapter 3. The probability distribution is arranged such that the cosine of the angle can be generated, with the desired probability density function,

$$\cos \theta = \begin{cases} \frac{1}{2g} \left[ 1 + g^2 - \left( \frac{1 - g^2}{1 - g + 2g\varepsilon} \right)^2 \right] & \text{if } g \neq 0 \\ 2\varepsilon - 1 & \text{if } g = 0 \end{cases} \quad (2-29)$$

As the photon propagates through the medium and undergoes many events the weight begins to decrease. Eventually the weight reaches a value where the information yielded by further propagating the photon is negligible. Here a roulette method is used to determine whether the photon should be propagated further [32].

Once the photon passes below a threshold weight they are subjected to a pseudorandom number test [32]. Here, photons which pass have their weights increased – those that don't are terminated. The weight increase is inversely proportional to the chance that the photon will terminate, allowing conservation of energy to be respected [32]. Once all the photons have propagated, the absorption can be converted into fluence. The fluence is [74]

$$F[x, y, z] = \frac{A[x, y, z]}{\mu_a} \quad (2-30)$$

It is also important to scale the fluence by the number of photons used.

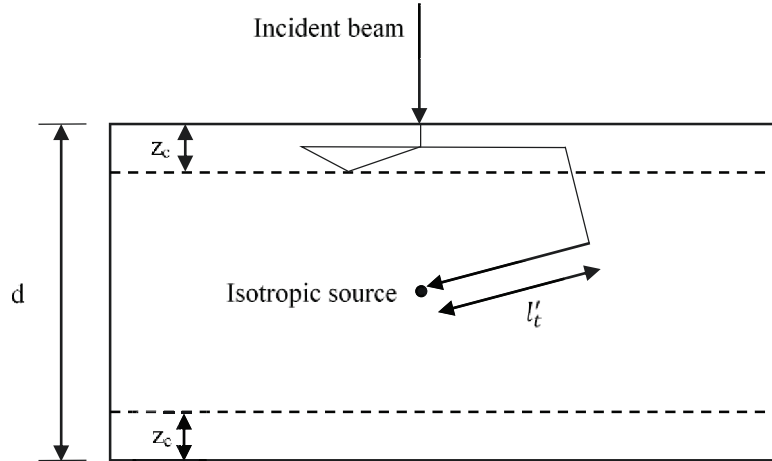


Figure 2.3. Example simulation medium using both MC and Diffusion, beginning with an incident beam. Here, the final step (outlined with  $l'_t$ ) converts the photon to a point source [75].

## 2.11 Hybrid Method of Monte Carlo and Diffusion

The previous sections illustrated two significantly different methods for evaluating the fluence. Research has shown how these two methods can be combined to increase the overall efficiency [76 – 79]. There are other hybrids as well [80], but these will not be discussed in this thesis. In this hybrid, MC is used to evaluate the sections near to a boundary and the initial source, while Diffusion is used once the photons have propagated far enough away [75]. The critical component of the hybrid involves accurately transferring the weight of the photons from MC to Diffusion [75].

To convert the photon into an isotropic point source, the photon packets must travel far enough away from the boundaries or initial source [75]. Once they then scatter and are redirected, if in this new direction the mean free path of the photon stays far enough away from any boundaries, its weight can then be converted into a point source in an isotropically scattering medium [75]. This medium's scattering coefficient is the transport scattering coefficient [75].



Once all of the photons have been transmitted, numerical techniques can be used to determine the fluence from the resulting point sources. This method combines the accuracy of MC with the speed of Diffusion to produce its results, leading to accurate results which are typically 100 times faster than MC alone [43].

## 2.12 Discrete Ordinates

Discrete Ordinates is a method of solving the RTE by discretizing it over a specified number of solid angles [81 – 91]. It was originally introduced by Chandrasekhar [92], and unlike the other previously listed methods, Discrete Ordinates has found more recent use solving the time independent RTE in the field of heat transfer [81 – 88]. It is used here as it does lead to solutions of the RTE, and some of the ideas for how it solves the RTE can be applied to photon transport.

Blackbody radiation is typically used as a source, with number of solid angle segments being either 24 from  $S_4$ , or 96 from the  $LC_{11}$  quadrature scheme [82 – 88]. Although some authors have attempted to use this method with photon transport [89 – 91], it is not common. Typical solutions use the divergence theorem and discretize the surface of the elements.

In Discrete Ordinates the medium is first discretized into small pixels or voxels. The solid angle segments at these locations are also discretized. From the RTE, the scattering term becomes

$$\mu_s \int_{4\pi} L(\vec{r}, \hat{s}') P(\hat{s}' \cdot \hat{s}) d\Omega' \rightarrow \mu_s \sum_{i=1}^N L(\vec{r}, \hat{s}'_i) P(\hat{s}'_i \cdot \hat{s}_k) \quad (2-31)$$

Each other term in the RTE would then be solved for the direction  $\hat{s}_k$ , and solving the RTE involves iteratively solving each solid angle segment independently.

# Chapter 3

## Matrix Method Theory

The Matrix Method (MM) was developed by combining concepts from several light propagation techniques with the expectation of increasing the overall efficiency of these simulations. The main goal of this method, as well as the proposed process for simulating within a 2D homogeneous medium, is described within this chapter. The founding concepts are introduced as a repeating two step matrix multiplication, with each individual matrix examined and discussed. Due to computational complexity MM is re-imagined as the implementation of a series of convolution masks. The overall process will be analyzed in terms of its computational requirements and memory requirements.

## 3.1 How the Matrix Method is Equivalent to Monte Carlo

In Monte Carlo (MC), as photon packets are projected throughout a medium, they undergo a scattering and absorption event. The fraction which is absorbed is then stored [32]. This stochastic method determines the fluence ( $J/m^2$ ) by projecting the photon packets one at a time, with each photon packet undergoing multiple scattering and absorption events prior to the release of the next one [73]. To obtain the fluence with small deviations in the results between one location and another many photon packets must be used. The number of photon packets varies depending upon the simulation, however  $10^4$  to  $10^8$  is a commonly cited range [73].

MM does away with photon packets by discretizing the radiance ( $W/m^2 \cdot sr$ ) around the center of a pixel. The radiance subtended within these solid angle segments is then projected throughout the medium. After projection, it undergoes a scattering and absorption event. This is modeled by a reduction in the radiance throughout each pixel and each solid angle, followed by a redirection of the scattered radiance. These iterations are repeated until the remaining radiance is insignificant relative to the recorded fluence. Energy is conserved as the projection step does not reduce the radiance. It is only reduced in the scattering step.

## 3.2 Matrix Method Fundamentals

### 3.2.1 Grid Creation

In MM, a grid is constructed representing the media where the fluence is to be determined. In 2D simulations, the grid is composed of pixels which can be any shape or size. In this thesis, the pixels are square and each will have the same dimensions.

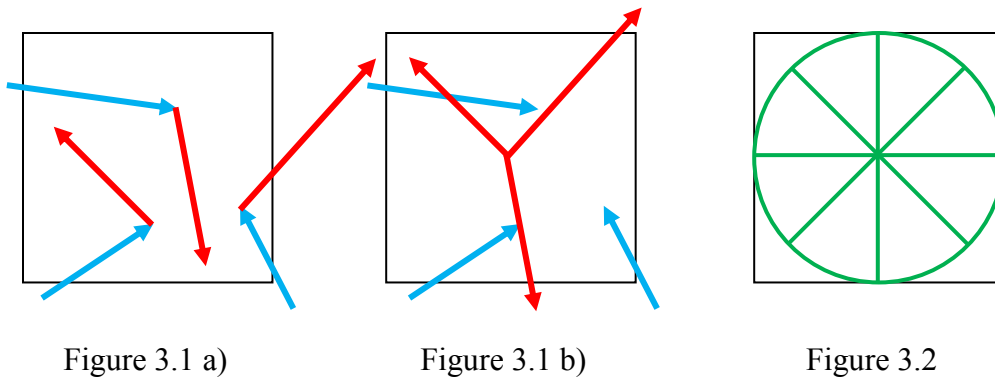


Figure 3.1. Examples of photon scattering in a typical simulation using a) MC and b) MM. The blue arrows represent inward photons while the red arrows represent scattered photons.  
 Figure 3.2. The discrete representation of the radiance with 8 different solid angle segments, each 45 degrees wide.

Using pixels with the same size and orientation allows one to use convolution in the projection step, which will be discussed later in this chapter. As a note to the reader, convolution can be done on any grid where the base elements are the same shape and orientated the same.

### 3.2.2 Approximations

In order to increase the overall efficiency of light propagation, several approximations are made. One of these approximations is that all of the scattering events occurring within a pixel occur at its center (Fig. 3.1). These scattering events are represented by redirecting the radiance ( $W/m^2 \cdot sr$ ). At the center of the pixel, the radiance is separated into  $J$  solid angles (Fig. 3.2). Each of these directions represents a solid angle segment of the radiance. Allocating scattering to the center of the pixel greatly reduces the total number of calculations.

A second approximation comes from Beer's Law (eq. 3-2), which is discussed in section 3.5 and 3.8. In MC, the theoretical maximum distance a photon can travel between scattering events is infinite [32]. Here, MM uses the same distribution (a rearrangement based on Beer's Law) up to a threshold fraction of intensity, where it's capped (i.e. further values are set to zero).

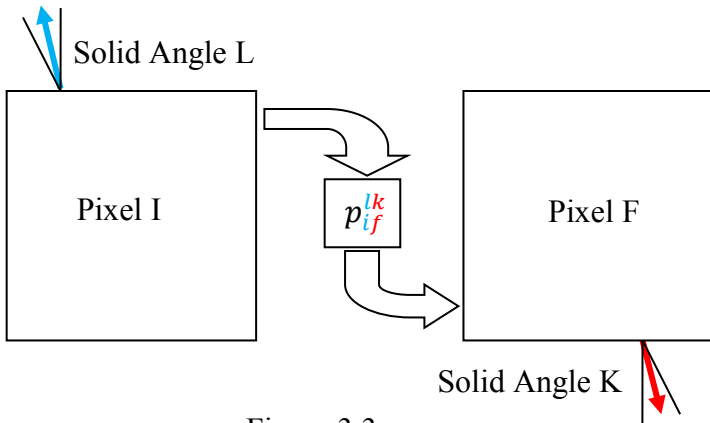


Figure 3.3

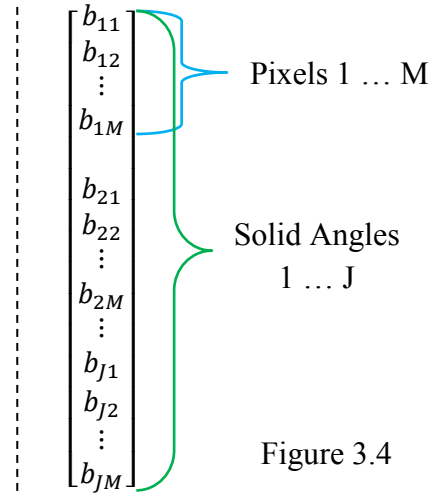


Figure 3.4

Figure 3.3. A pictorial representation of the propagation of radiance from pixel I, solid angle segment L to pixel F, solid angle segment K using the propagation matrix element  $p_{if}^{lk}$ .

Figure 3.4. The radiance ( $W/m^2 \cdot sr$ ) column vector  $\underline{b}_{\{ji\}}$ , used in the projection step. It is sorted such that the index of each element is given by  $i + M * (j - 1)$ , where  $i$  is the pixel and  $j$  is the solid angle segment.

The fraction of remaining intensity (cap) is expected to be less than  $10^{-4}$ . This particular approximation allows the simulation to maintain a high degree of accuracy while removing insignificant and unnecessary computations. By making these approximations, a new series of tools can be used. Instead of using two pseudorandom values, matrices with pre-calculated values are used. One matrix models the scattering effect while the other propagates the radiance.

### 3.3 Projection Step of Matrix Method

#### 3.3.1 Radiance

MM uses matrices to simulate the progression of the radiance as it propagates throughout the medium. The matrix operates on a vector  $\underline{b}_{\{ji\}}$ , which is used to represent the radiance of each and every solid angle segment within the medium.  $\underline{b}_{\{ji\}}$  is a column vector of size  $JM$  where each element represents the radiance within a pixel  $i$  and solid angle segment  $j$ .

$$\begin{bmatrix}
\left[ \begin{array}{cccc} p_{11}^{11} & p_{21}^{11} & \dots & p_{M1}^{11} \\ p_{12}^{11} & p_{22}^{11} & \dots & p_{M2}^{11} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{11} & p_{2M}^{11} & \dots & p_{MM}^{11} \end{array} \right] &
\left[ \begin{array}{cccc} p_{11}^{21} & p_{21}^{21} & \dots & p_{M1}^{21} \\ p_{12}^{21} & p_{22}^{21} & \dots & p_{M2}^{21} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{21} & p_{2M}^{21} & \dots & p_{MM}^{21} \end{array} \right] &
\dots &
\left[ \begin{array}{cccc} p_{11}^{J1} & p_{21}^{J1} & \dots & p_{M1}^{J1} \\ p_{12}^{J1} & p_{22}^{J1} & \dots & p_{M2}^{J1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{J1} & p_{2M}^{J1} & \dots & p_{MM}^{J1} \end{array} \right] \\
\left[ \begin{array}{cccc} p_{11}^{12} & p_{12}^{12} & \dots & p_{M1}^{12} \\ p_{12}^{12} & p_{22}^{12} & \dots & p_{M2}^{12} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{12} & p_{2M}^{12} & \dots & p_{MM}^{12} \end{array} \right] &
\left[ \begin{array}{cccc} p_{11}^{22} & p_{21}^{22} & \dots & p_{M1}^{22} \\ p_{12}^{22} & p_{22}^{22} & \dots & p_{M2}^{22} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{22} & p_{2M}^{22} & \dots & p_{MM}^{22} \end{array} \right] &
\dots &
\left[ \begin{array}{cccc} p_{11}^{J2} & p_{21}^{J2} & \dots & p_{M1}^{J2} \\ p_{12}^{J2} & p_{22}^{J2} & \dots & p_{M2}^{J2} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{J2} & p_{2M}^{J2} & \dots & p_{MM}^{J2} \end{array} \right] \\
& \vdots & & \vdots \\
\left[ \begin{array}{cccc} p_{11}^{1J} & p_{21}^{1J} & \dots & p_{M1}^{1J} \\ p_{12}^{1J} & p_{22}^{1J} & \dots & p_{M2}^{1J} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{1J} & p_{2M}^{1J} & \dots & p_{MM}^{1J} \end{array} \right] &
\left[ \begin{array}{cccc} p_{11}^{2J} & p_{21}^{2J} & \dots & p_{M1}^{2J} \\ p_{12}^{2J} & p_{22}^{2J} & \dots & p_{M2}^{2J} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{2J} & p_{2M}^{2J} & \dots & p_{MM}^{2J} \end{array} \right] &
\dots &
\left[ \begin{array}{cccc} p_{11}^{JJ} & p_{21}^{JJ} & \dots & p_{M1}^{JJ} \\ p_{12}^{JJ} & p_{22}^{JJ} & \dots & p_{M2}^{JJ} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^{JJ} & p_{2M}^{JJ} & \dots & p_{MM}^{JJ} \end{array} \right]
\end{bmatrix}$$

Figure 3.5. The projection matrix with  $M$  pixels and  $J$  solid angle segments.

$J$  represents the total number of solid angle segments within a pixel and  $M$  is the total number of pixels in a medium.  $b_{\{ji\}}$  uses two values for its index, and the values are arranged such that index  $\{ji\}$  corresponds to entry  $i + M * (j - 1)$ . Figure 3.4 shows an example column.

### 3.3.2 Projection Matrix

The matrix  $\underline{P}$  (Fig. 3.5) is used to simulate the propagation of photons between scattering events. In the projection matrix, each element  $p_{if}^{lk}$  corresponds to the fraction of the radiance propagating from the solid angle segment  $l$  within pixel  $i$  to solid angle segment  $k$  within pixel  $f$  (Fig. 3.3). This matrix has a size of  $M^2 J^2$ . In a simulation using 501 x 501 pixels with 60 solid angle segments, the matrix would represent nearly 227 trillion values. This is impractical, and some analysis must be done to compress its size to a realistic value. By re-examining MC, photons do not change direction prior to undergoing a scattering event. The projection matrix can assume that radiance doesn't propagate into different solid angle segments.



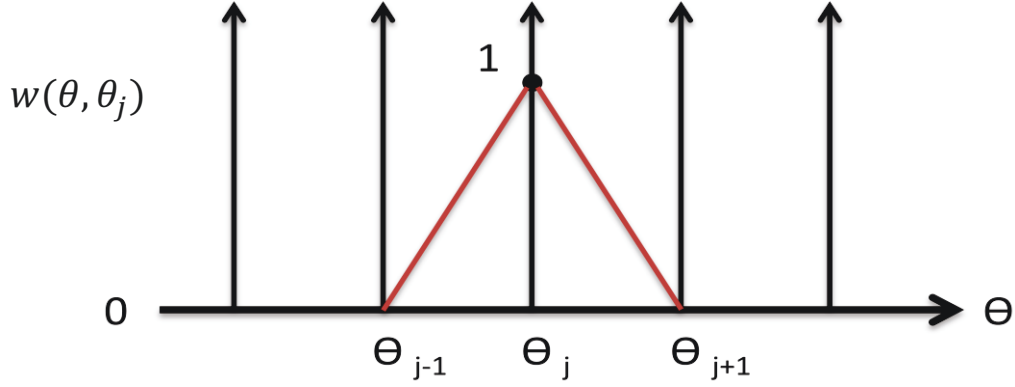


Figure 3.7. A plot of the interpolation function. This function is used as a linear fit for the radiance, and will be represented by the weighing function  $w(\theta, \theta_j)$ . Here index  $j$  represent any single solid angle segment from 1 to  $J$ .

The fraction of radiance which scattered within the pixel needs to be determined. Due to conservation of energy (COE), the initial intensity  $I_0$  will either be transmitted ( $T$ ), absorbed ( $A$ ) or scattered ( $S$ ) along any specific path [28]. Radiance can replace intensity as it also follows COE. This gives rise to equation 3-3 (reflectance removed and scattering term added from [93]).

$$I_0 = I_T + I_A + I_S \quad (W/m^2) \quad (3-3)$$

$A$  and  $S$  are related by a simple ratio [94]

$$\frac{I_A}{I_S} = \frac{\mu_a}{\mu_s} \quad (3-4)$$

And from these relations the fraction of photons scattered along a given path  $d$  with constant scattering and extinction coefficients is (replacing intensity with radiance ( $W/m^2 \cdot sr$ ))

$$R_S = R_0 \frac{\mu_s}{\mu_t} (1 - e^{-\mu_t d}) \quad (W/m^2 \cdot sr) \quad (3-5)$$

An important feature to address involves ray tracing effects. The projections from individual discrete radiance values would not include the entire range of the solid angle segment.



The scattering over the entire solid angle segment from each of these additional rays is added to the matrix element. To account for this, a linear interpolation factor (Fig. 3.7) is used. This will be represented by a weighing function  $w(\theta, \theta_j)$ , and is defined (eq. 3-6)

$$w(\theta, \theta_j) = \begin{cases} 1 - \frac{(\theta_j - \theta)}{(\theta_j - \theta_{j-1})}, & \theta_{j-1} \leq \theta \leq \theta_j \\ \frac{(\theta - \theta_j)}{(\theta_{j+1} - \theta_j)}, & \theta_j \leq \theta \leq \theta_{j+1} \\ 0, & \textit{otherwise} \end{cases} \quad (3-6)$$

The scattering contribution along  $\theta$  due to a single solid angle segment  $j$  is given by

$$R_S(\theta) |_j = w(\theta, \theta_j) R_0(\theta_j) e^{(-\mu_t * d_{i \rightarrow f})} \frac{\mu_s}{\mu_t} (1 - e^{-\mu_t d(\theta)}) \quad (3-7)$$

The factor  $d(\theta)$  represents the path length through a pixel along angle  $\theta$ ,  $d_{i \rightarrow f}$  is the path length to pixel  $f$  from pixel  $i$  along angle  $\theta$ , and  $R_0(\theta_j)$  ( $W/m^2 \cdot sr$ ) is the value of the discretized radiance within solid angle segment  $j$  from the initial pixel.

The total fraction of the radiance scattered at any angle  $\theta$  uses both the solid angle segments greater than and less than the angle itself. Assuming that  $\theta_{j+1} > \theta > \theta_j$ , the total scattering contribution along a given path length  $d(\theta)$  is

$$R_S(\theta) = [R_0(\theta_j)w(\theta, \theta_j) + R_0(\theta_{j+1})w(\theta, \theta_{j+1})] \times e^{(-\mu_t * d_{i \rightarrow f})} \frac{\mu_s}{\mu_t} (1 - e^{-\mu_t d(\theta)}) \quad (3-8)$$

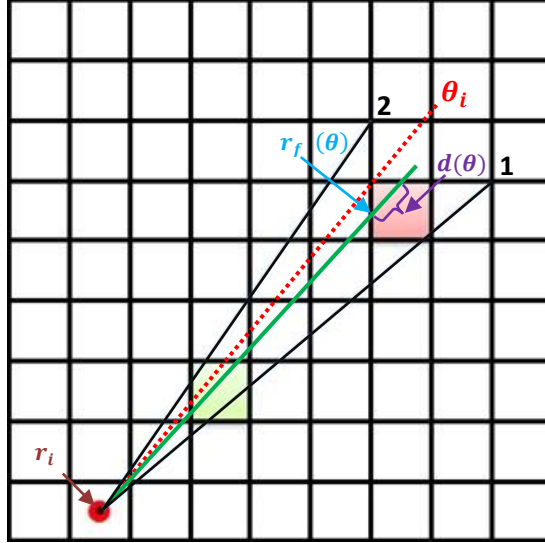


Figure 3.8. An example of a green ray projected through a medium and through a specific pixel. Rays 1 and 2 – black, centered at  $\theta_1$  and  $\theta_2$  – are the boundaries of the solid angle segment. For the destination pixel,  $r_i$ ,  $r_f(\theta)$ , and  $d(\theta)$  are illustrated. Here, one can see the angular span for each value of  $p_{if}^k$  will change, depending on the pixel. In the red pixel, although the solid angle segment spans from  $\theta_1$  to  $\theta_2$ , the span from  $\theta_1$  to  $\theta_i$  is used.  $\theta_i$  is the angle which intersects the upper left corner, and is outlined as a single dotted red ray.

With an inhomogeneous medium, one can still use Beer's Law (eq. 3-2). Using  $r_i$ ,  $r_f(\theta)$ ,  $\mu_t(l)$  ( $cm^{-1}$ ), the location along the pixels' edge and the extinction coefficient, Beer's Law yields

$$I_0(\theta) = I e^{(-\int_{r_i}^{r_f(\theta)} \mu_t(l) dl)} \quad (W/m^2) \quad (3-9)$$

With these results  $p_{if}^k$  can be evaluated. As these values represent the fraction of the radiance which is projected into a final pixel  $f$  from initial pixel  $i$  within solid angle segment  $k$ , they can be calculated by using an integral over the entire range of the solid angle segment.

The final piece to this integral involves accounting for the case in which the range of the solid angle segment extends beyond the boundaries of the pixel. This can be accounted for with a simple adjustment to the integral range.

The integral becomes

$$p_{if}^k = \int_{\theta_i}^{\theta_f} w(\theta, \theta_k) e^{(-\int_{r_i}^{r_f(\theta)} \mu_t(l) dl)} \frac{\mu_s}{\mu_t} (1 - e^{-\mu_t d(\theta)}) d\theta \quad (3-10)$$

With  $\theta_i$  and  $\theta_f$  representing the range of angles which enter the pixel within the solid angle segment. These are either the minimum and maximum angles within the solid angle segment or the angles which coincide with the corners of the pixel (Fig. 3.8).

## 3.4 Limit Analysis of Projection Elements

### 3.4.1 Scattering Limit Analysis

It is important to confirm that this expression evaluates to known results in specific circumstances. When the limit of the pixel size approaches zero, it is expected that the fraction of the radiance scattered within the pixel also approaches zero.

To take the limit as the pixel size approaches zero involves taking two separate steps. The first step condenses the angle range to a central value, while the second step involves the limit as the path length approaching zero.

In the first step, condensing the angle range to the angle passing through the center of the pixel occurs, yielding

$$p_{if}^k(\theta_c) = w(\theta_c, \theta_k) e^{(-\int_{r_i}^{r_f(\theta_c)} \mu_t(l) dl)} \frac{\mu_s}{\mu_t} (1 - e^{-\mu_t d(\theta_c)}) \quad (3-11)$$

Upon inspection this is simply a coefficient multiplied by a term dependent upon the length of a beam crossing through the pixel. The coefficient is

$$C = w(\theta_c, \theta_k) \frac{\mu_s}{\mu_t} e^{-\int_{r_i}^{r_f(\theta_c)} \mu_t(l) dl} \quad (3-12)$$

The second step involves taking the limit as the length of the beam within the pixel approaches zero. This is represented as

$$\lim_{d(\theta_c) \rightarrow 0} p_{if}^k(\theta_c) \quad (3-13)$$

Evaluating this expression yields

$$\begin{aligned} \lim_{d(\theta_c) \rightarrow 0} (p_{if}^k(\theta_c)) &= \lim_{d(\theta_c) \rightarrow 0} C (1 - e^{-\mu_t d(\theta_c)}) \\ &= \lim_{d(\theta_c) \rightarrow 0} C (1 - (1 - \mu_t d(\theta_c))) \\ &= \mu_t C \lim_{d(\theta_c) \rightarrow 0} (d(\theta_c)) \\ &= 0 \end{aligned}$$

Which is the expected result for the values of  $p_{if}^k$ . In the event that there is no area or path where scattering can occur, the amount of scattering must be zero.

Another limit test involves determining what fraction of the radiance is projected to the pixel, as the pixel size approaches zero. In this case the projection term should evaluate to Beer's Law (eq. 3-2).

### 3.4.2 Projection Limit Analysis

The transmission to the center of the pixel ( $t_{if}^k$ ) will be used instead of  $p_{if}^k$ . The expression uses

$$T = e^{-\mu_t d(\theta_c)/2} \quad (3-14)$$

This modifies the expression for  $p_{if}^k$  to

$$t_{if}^k(\theta_c) = w(\theta_c, \theta_k) e^{-\int_{r_i}^{r_f(\theta_c)} \mu_t(l) dl} (e^{-\mu_t d(\theta_c)/2}) \quad (3-15)$$

Before this analysis is continued, it is useful to mathematically define  $r_{fc}$ . Here,

$$r_{fc} = r_f(\theta_c) + d(\theta_c)/2 \quad (3-16)$$

with  $r_{fc}$  as the location of the center of the pixel (Fig. 3.8). Simplifying yields

$$t_{if}^k(\theta_c) = w(\theta_c, \theta_k) e^{-\int_{r_i}^{r_{fc}} \mu_t(l) dl} \quad (3-17)$$

Taking the limit as the length goes to zero yields Beer's Law with the weighing factor.

$$\lim_{d(\theta_c) \rightarrow 0} (t_{if}^k(\theta_c)) = w(\theta_c, \theta_k) e^{-\int_{r_i}^{r_{fc}} \mu_t(l) dl} \quad (3-18)$$

These tests demonstrate that the expression for  $t_{if}^k$  yields the classical results when the pixel it projects to has no area.

$$\begin{bmatrix}
\begin{bmatrix} p_{11}^1 & p_{21}^1 & \dots & p_{M1}^1 \\ p_{12}^1 & p_{22}^1 & \dots & p_{M2}^1 \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^1 & p_{2M}^1 & \dots & p_{MM}^1 \end{bmatrix} & |0\rangle & \dots & |0\rangle \\
|0\rangle & \begin{bmatrix} p_{11}^2 & p_{21}^2 & \dots & p_{M1}^2 \\ p_{12}^2 & p_{22}^2 & \dots & p_{M2}^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^2 & p_{2M}^2 & \dots & p_{MM}^2 \end{bmatrix} & \dots & |0\rangle \\
\vdots & \vdots & \ddots & \vdots \\
|0\rangle & |0\rangle & \dots & \begin{bmatrix} p_{11}^J & p_{21}^J & \dots & p_{M1}^J \\ p_{12}^J & p_{22}^J & \dots & p_{M2}^J \\ \vdots & \vdots & \ddots & \vdots \\ p_{1M}^J & p_{2M}^J & \dots & p_{MM}^J \end{bmatrix}
\end{bmatrix}
\begin{bmatrix} b_{11}^r \\ b_{12}^r \\ \vdots \\ b_{1M}^r \\ b_{21}^r \\ b_{22}^r \\ \vdots \\ b_{2M}^r \\ \vdots \\ b_{j1}^r \\ b_{j2}^r \\ \vdots \\ b_{jM}^r \end{bmatrix} = \begin{bmatrix} b_{11}^{r+1} \\ b_{12}^{r+1} \\ \vdots \\ b_{1M}^{r+1} \\ b_{21}^{r+1} \\ b_{22}^{r+1} \\ \vdots \\ b_{2M}^{r+1} \\ \vdots \\ b_{j1}^{r+1} \\ b_{j2}^{r+1} \\ \vdots \\ b_{jM}^{r+1} \end{bmatrix}$$

Figure 3.9. An example of the projection step acting on  $\underline{b}_{\{ji\}}$ . The index  $r$  is used as an indicator.

### 3.5 Reducing the Number of Coefficients due to Beer's Law

Beer's Law (eq. 3-2) introduces a concept which allows the number of elements in  $\underline{P}$  to be further reduced. Due to the high number of scattering events occurring in tissue, the photons will only propagate to nearby pixels before undergoing a scattering event. Assuming a value of  $100 \text{ cm}^{-1}$  for the scattering coefficient, only a fraction ( $10^{-10}$ ) of the radiance has not yet scattered when projected 2.5 mm. If the values for  $p_{if}^k$  past this distance are set to zero, the memory needed for the projection matrix can be further reduced with little impact on the final results.

Each sub matrix was assumed to use  $M^2$  elements. By changing the number of non-zero entries in each row to 100, this reduces the number of stored elements to  $100M$  per sub matrix. Using a  $501 \times 501$  grid with 60 solid angle segment segments would now only use slightly over 3 gigabytes of memory. The projection step (Fig. 3.9) is the matrix multiplication (eq. 3-19)

$$\underline{P} \underline{b}_{\{ji\}}^r = \underline{b}_{\{ji\}}^{r+1} \quad (3-19)$$

$$\begin{bmatrix} b_{11} \\ b_{12} \\ \vdots \\ b_{1J} \\ \\ b_{21} \\ b_{22} \\ \vdots \\ b_{2J} \\ \vdots \\ \\ b_{M1} \\ b_{M2} \\ \vdots \\ b_{MJ} \end{bmatrix}$$

Figure 3.10. This is the radiance column vector  $\underline{b}_{\{ij\}}$ , used in the scattering step. It is sorted such that the index of each element is given by  $j + J * (i - 1)$ , where  $i$  is the pixel and  $j$  is the solid angle segment.

In figure 3.9, the projection matrix is operating on the radiance vector. As can be seen, the radiance is changed to a new value due to the projection operation. Index  $r$  does not represent a complete iteration; it is only used as an index for the projection step of the cycle.

### 3.6 Scattering Step of Matrix Method

The scattering step of MM uses a low number of calculations, which leads to a higher performance per iteration. As previously mentioned, the scattering step can be simulated by matrix multiplication. The scattering matrix  $\underline{S}$  simulates the scattering operation within every pixel. Every element in  $\underline{S}_{\{ji\}}$  represents the fraction of the radiance within solid angle segment  $i$  that is scattered into solid angle segment  $j$ . By rearranging the column vector  $\underline{b}_{\{ji\}}$  to  $\underline{b}_{\{ij\}}$  (Fig. 3.10), the scattering matrix can be written in the form of a block diagonal matrix operating on individual pixels (Fig. 3.12).

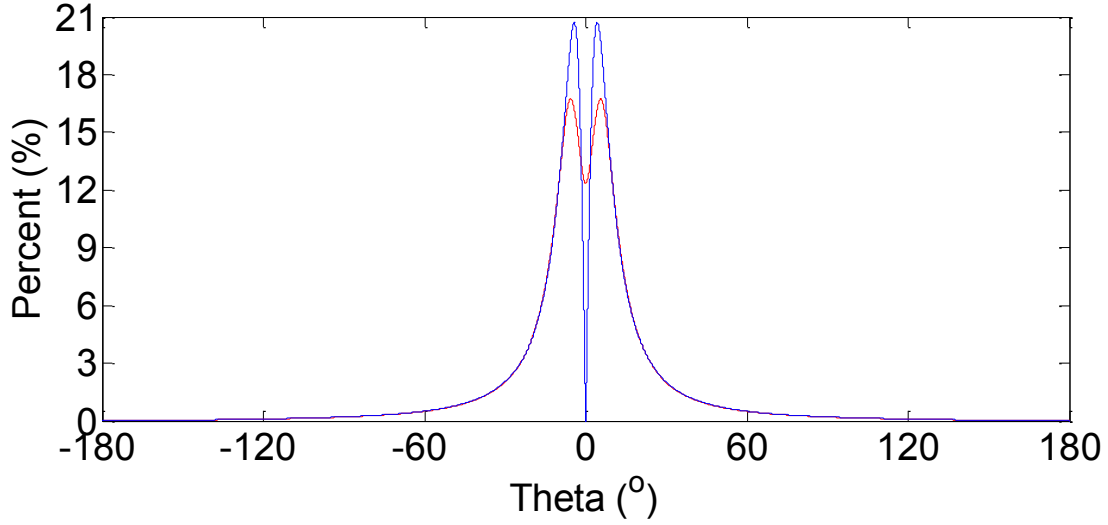


Figure 3.11. The (blue) PDF and (red) modified PDF of the Henyey-Greenstein function.

The matrix  $\underline{T}$  is used to represent the reordering operation from column vector  $\underline{b}_{\{ji\}}$  to  $\underline{b}_{\{ij\}}$ . The column vector, after matrix multiplication with  $\underline{T}$ , represents a list of solid angle segments within their respective pixels. Its inverse  $\underline{T}^{-1}$  can return the column vector  $\underline{b}_{\{ji\}}$ .

$$\underline{T} \underline{b}_{\{ji\}} = \underline{b}_{\{ij\}} \quad (3-20)$$

The reordering operator  $\underline{T}$  is used to structure the radiance vector in a way to benefit subsequent scattering operations. In practice this is done by simple element reordering rather than matrix multiplication but the matrix operator provides a convenient notation for analysis.

### 3.6.1 Scattering Matrix Elements

The values within the scattering matrix must be determined. From MC, the Henyey-Greenstein function (HG) is a commonly used probability density function (PDF) which determines the direction of the photon packet after a scattering event (eq. 3-21) [95].



$$P_{HG}(\theta) = \frac{1}{2} \frac{(1 - g^2)|\sin \theta|}{(1 + g^2 - 2g \cos \theta)^{3/2}} \quad (3-21)$$

where  $g$  is the anisotropy and  $\theta$  is the deflection a photon packet undergoes.

HG was originally developed to address galactic scattering [96], but is currently finding use in the field of biomedical optics [56 – 72]. It is a phenomenological phase function used with practical success and is attractive compared to Mie scattering phase functions owing to its simplicity. The entries of the scattering matrix use the fraction of the photons which are scattered over the whole range of the solid angle segment, not just the value at the solid angle segments' center. To find these values, the PDF is scaled by the weighting factor at each angle, and the results are summed. Then they are scaled such that the sum of these values is one. To illustrate the differences between HG and the modified PDF, they are plotted in figure 3.11. The modified PDF is plotted with 60 points.

To determine the entries of the scattering matrix, the modified PDF needs to be discretized. The discretized values of the modified PDF represent the fraction of photons which will be scattered into that direction. By recording the values in the modified PDF corresponding to the center of the solid angle segments, the overall scattering effect can be simulated. These value become the entries of the scattering matrix  $\underline{S}_{\{ji\}}$ .

An interesting feature of the scattering matrix comes from the following fact. The probability of a photon scattering by any given angle is independent of its initial direction. Extending this fact to encompass radiance, it can be stated that the fraction of the radiance subtended within any given solid angle segment which scatters by any specific angle will scatter by that same angle regardless of its initial solid angle segment.



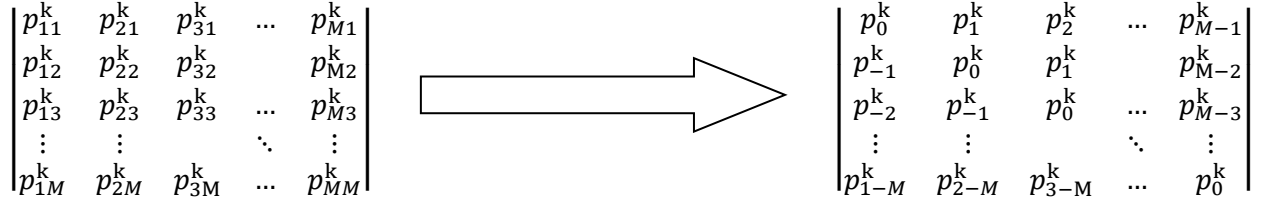


Figure 3.13. Transition from a typical sub matrix to one which is only dependent on distance.

### 3.7 Overall Matrix Equation

The overall final matrix multiplication completing one cycle in MM is

$$\underline{b}^{i+1} = \underline{T}^{-1} \underline{STP} \underline{b}^i \quad (3-22)$$

with  $i + 1$  representing the next iteration in values of the radiance vector  $b_{\{ji\}}$ . Although in some applications matrix multiplication is feasible, most are simply too complicated, and become impossible due to the large amount of memory needed. For example, in a medium represented with 501 x 501 pixel elements and 60 solid angle segments the  $\underline{P}$  matrix alone uses over 6 gigabytes of memory, and this is for a 2D problem. 3D systems clearly need more analysis to make this a feasible method. Further simplifications are discussed in section 3.8.

### 3.8 Toeplitz Matrices

The projection step of MM currently uses a matrix which, in most applications, would simply be too large. This is especially true in 3D simulations. Unfortunately,  $\underline{P}$  uses the smallest amount of memory it can without further simplifications. In the case of homogeneous conditions, another simplification can be applied. For homogeneous conditions  $p_{if}^k$  can be rewritten as

$$p_{if}^k = \frac{\mu_s}{\mu_t} e^{-\mu_t |r_f - r_i|} \int_{\theta_{k-1}}^{\theta_{k+1}} w(\theta, \theta_k) (1 - e^{-\mu_t d(\theta)}) d\theta \quad (3-23)$$

$$\begin{pmatrix} p_0^k & p_1^k & p_2^k & \dots & p_t^k & \dots & 0 \\ p_{-1}^k & p_0^k & p_1^k & & & & 0 \\ p_{-2}^k & p_{-1}^k & p_0^k & \dots & & & 0 \\ \vdots & & \vdots & \ddots & & & \vdots \\ p_{-t}^k & & & & p_0^k & & p_t^k \\ \vdots & & & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & p_{-t}^k & \dots & p_0^k \end{pmatrix}$$

Figure 3.14. Projection sub matrix within a homogeneous medium. This is an example of a Toeplitz matrix. Each row only has non-zero values in the entries ranging from  $p_{-t}^k$  to  $p_t^k$ . These values represent the threshold where any radiance propagating past it can be considered negligible to the simulation.

Eq. 3-23 depends on the distance between the initial pixel and the edge of the final pixel closest to the initial pixel. There is also a factor associated with the distance through to the other side of the pixel. This is significantly different the previous expression for  $p_{if}^k$  (eq. 3-10), which also depended on the specific path to the pixel.

The main benefit is that for any entry  $p_{if}^k$ , as long as it projects the radiance within a homogeneous section and the solid angle segment does not change, the entry  $p_{if}^k$  will not change. This holds true as long as the difference between  $f$  and  $i$  does not change.

The projection sub matrices can now be relabeled as Toeplitz matrices. The Toeplitz matrix is similar to the circulant matrix in that rows are repeated; however in Toeplitz matrices these values are not wrapped around [98]. For the projection sub matrix the entries representing the projection to pixels past the threshold range can be set to zero.

This is a significant development as Toeplitz matrices are simply a way to represent convolution through matrix multiplication [98]. The projection step can be re-defined with convolution. By using convolution, the number of entries required for projection can be reduced from  $50M$  per sub matrix to 50 entries per convolution mask, with one convolution mask per solid angle segment. This development allows MM to be feasible.

Threshold Intensity	Distance (cm)	Number of Pixels	Total Mask Size
$10^{-5}$	0.1151	23	47 x 47
$10^{-8}$	0.1842	37	75 x 75
$10^{-10}$	0.2303	46	93 x 93

Table 3.1. Sizes of the convolution mask determined by using an extinction coefficient of  $100 \text{ cm}^{-1}$  and a grid with square elements of size of  $50 \text{ }\mu\text{m}$  by  $50 \text{ }\mu\text{m}$ . The threshold intensity is the ratio of final intensity to initial intensity.

### 3.9 Projection with Convolution

The convolution masks used in MM are much smaller than  $M^2$ . Their size is determined by setting a minimum threshold value for the projected radiance. Using Beer's Law (eq. 3-2), the distance a ray would travel before reaching a threshold value can be evaluated. As the pixel size is known, the number of pixels which that ray would enter can be determined. As rays are projected in every direction and begin at a center pixel, the size of the masks is simply twice the number of pixels used for one ray, with one more to account for the center pixel (Table 3.1).

Each value within the convolution masks represents the fraction of radiance which undergoes scattering and absorption within that pixel. As these masks are designed to be used in a homogeneous medium, equation 3-23 can be used to determine the values within the masks.

In practice, it is more computationally efficient to use a ray tracing subroutine to determine the coefficients for these masks. This subroutine generates equally spaced rays and evaluates the fraction of these rays' which scatters or absorbs radiance within a given pixel. By repeating this process for each solid angle, the convolution masks required for the simulation can be constructed. Each masks has a size of  $K \times K$  and has non-zero values corresponding to the values from the projection sub matrix of  $p_{-t}^k$  to  $p_t^k$ .

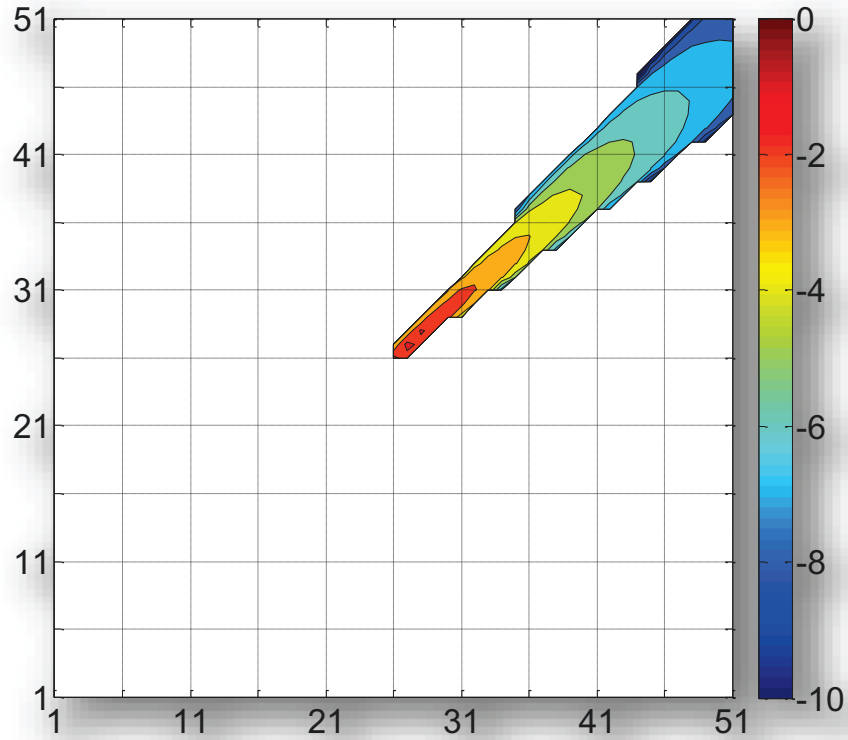


Figure 3.15. Examples of a 51 x 51 convolution mask. Here, the mask covers the 8<sup>th</sup> solid angle segment out of 60 total solid angle segments. This corresponds to a central angle of 42°. The scale is the log<sub>10</sub> values of  $p_{if}^k$ .

### 3.10 Absorption

During each projection and scattering step, absorption must also occur. Here, the radiance is reduced by a factor of  $(1 - "a")$ , with “a” being the albedo [32]. The albedo is the ratio of the scattering coefficient to the total extinction coefficient [99].

$$albedo = \frac{\mu_s}{(\mu_a + \mu_s)} \tag{3-24}$$

Each absorption step requires  $MJ$  calculations and  $M$  memory to complete. As this is a very fast step it will be dropped from both the computational requirements and the memory analysis portion of this thesis.

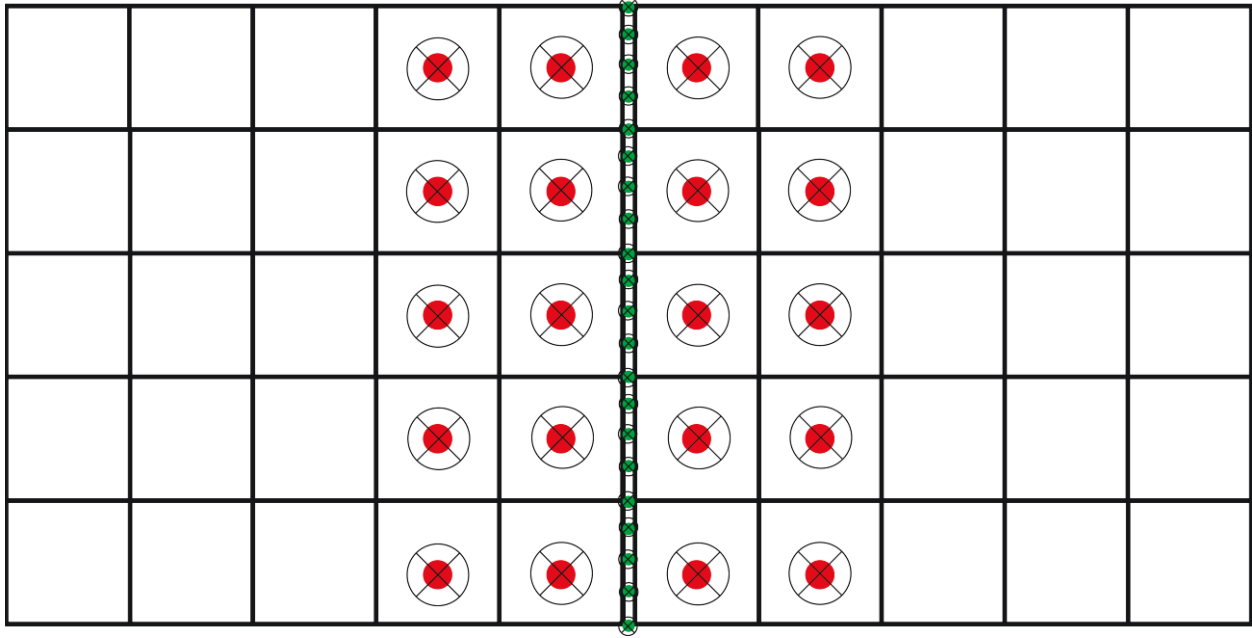


Figure 3.16. An example of a boundary crossing. The red dots with the overlaid radiance segments represent the normal neighbouring pixels while the green dots are the boundary points. Each of these green dots has the same number of solid angle segments.

### 3.11 Boundaries

Although this thesis focuses on homogeneous mediums, a discussion addressing boundaries will be included. Boundaries play a critically important role in many light transport applications, and biomedical applications are no exception. With MM boundaries can be challenging to address, but like MC additional programming simply needs to be incorporated. There are a several ways to compensate for boundaries.

One approach involves lining the boundary with large number of points. By using a ray tracing subroutine similar to what was proposed in the projection step, a simple matrix can be built to transfer the radiance within neighbouring pixels to the boundary points. Another matrix can then be built to transfer these values back to the pixels. Each of these boundary points would have the same number of solid angle segments as the pixels (Fig. 3.16).

### 3.11.1 Index-Matched Boundaries

With boundary points, there are two situations that need to be considered. The first involves an index-matched boundary. Here, the only steps are determining which pixels the boundary points receive their radiance from and where that radiance is projected to. Once this is done, a ray tracing program using Beer's Law (eq. 3-2) can be developed to determine the transfer coefficients for moving the radiance to and from these boundary points. These coefficients can be built into a matrix – or in some cases a mask – to improve its efficiency.

### 3.11.2 Index-Mismatched Boundaries

The second situation involves index-mismatched boundaries. Although the same initial setup from index-matched boundaries is used, one additional step is required to address reflection and refraction. A reflection and refraction matrix (RR) is created to address the index mismatch. Unlike scattering where each pixel with the same anisotropy can use the same matrix, RR matrices change for each boundary point that has a different slope or different index mismatch. The RRs are built directly from the Fresnel equations (eq. 3-25 & 26) [100] – reflectance is reflectivity squared.

$$R_S = \left| \frac{n_1 \cos \theta_i - n_2 \cos \theta_t}{n_1 \cos \theta_i + n_2 \cos \theta_t} \right|^2 \quad (3-25)$$

$$R_P = \left| \frac{n_1 \cos \theta_t - n_2 \cos \theta_i}{n_1 \cos \theta_t + n_2 \cos \theta_i} \right|^2 \quad (3-26)$$

$n_1$  and  $n_2$  are the indexes of refraction of the incident and second medium, while  $\theta_i$  and  $\theta_t$  are the incident and transmitted angles, respectively.



The average of these two equations is used to determine the fraction of the radiance which is reflected. Conservation of energy then allows the fraction transmitted to be calculated, as no absorbance occurs at a single point (section 3.4.1).

The solid angle segments where these fractions are either transmitted or reflected can be determined using two equations, Snell's Law and the Law of Reflection (eq. 3-27 & 28) [100]

$$n_1 \sin \theta_{inc} = n_2 \sin \theta_{trans} \quad (3-27)$$

$$\theta_{inc} = \theta_{refl} \quad (3-28)$$

These equations depend only on the relative index of refraction and the angle of incidence from the projecting solid angle segment. This is a critical development as a series of small matrices (size  $J^2$ ) can be developed to simulate the reflection and transmission at boundary points. If these matrices are created during the initial step, it is important to note that they will be sparse due to the physics involved. Another option would be to simply have the program generate these values as needed, if memory is an issue.

## 3.12 Addressing Boundaries in Pencil Beam and Point Source

As the issue of boundaries arises in both the point source and the pencil beam simulations of this thesis, it seems necessary to explain in detail how they are addressed. The infinite medium and semi-infinite medium cannot be calculated with MM simply due to memory limitations, and so turbid media within a non-scattering ambient medium are used as models. To address the boundaries between the scattering and non-scattering media, after each projection operation the radiance outside of the scattering media are set to zero. This ensures this radiance does not scatter back into the turbid media. Note: This applies to index-matched boundary conditions.

Although not yet implemented, if the boundaries were index mismatched then the point system seen in figure 3.16 could be used. This would entail projecting the radiance to the boundary, using the RR matrices to determine the reflected magnitude, and re-transmitting the reflected portion back into the turbid media. The transmitted portion would be set to zero to ensure the ambient radiance would not scatter back into the medium. As the boundaries are planar, only one RR matrix would be needed. The angle which the radiance is reflected to can be found using eq. 3-28.

### 3.13 Memory

One of the major differences between MM and MC is the significant increase in memory MM requires. The memory requirement for MM comes from several main factors, being the grid, the number of solid angles, and the size of the convolution masks. The memory from the scattering matrix and the convolution masks are insignificant in a homogeneous medium. There are  $MJ$  radiance values, with each value represented as a floating point number using 8 bytes.

This comes to 120 MB for a 501 x 501 grid with 60 solid angle segments. In a 3D medium with dimensions of 501 x 501 x 501, the memory needed increases to 60 GB. In 2D this is a trivial issue; however for 3D systems the memory requirements can be very cumbersome.

It should be noted additional time is required as each operation which either accesses or returns a value from memory takes a considerably longer when there are significantly more values. In future applications, 3D systems are expected to be simulated using parallel processing and cloud computing techniques. As this simulation method can compute a result by segmenting the radiance, these methods will reduce the computational load from memory access operations.

## 3.14 Computational Complexity

In order to determine how MM would compare to other techniques the computational complexity must be examined. Multiplication operations will be counted to determine the overall complexity. Other operations, such as accessing memory and addition, will not be included.

As MM is an iterative method, the focus of this section will only address the iterative section, being the projection, scattering and absorption steps. Absorption only uses one multiplication for every pixel, and so will not be included in the overall analysis. The first step which will be analyzed is the scattering step.

### 3.14.1 Scattering Operation

As previously mentioned, the scattering operation uses a  $J^2$  matrix to simulate the scattering effect. Here, each element  $\underline{S}$  represents the fraction of the radiance scattered from an initial solid angle segment  $i$  to a final solid angle segment  $j$ .

Each scattering operation uses  $J^2$  operations. As there are  $M$  pixels, there are  $O(MJ^2)$  operations. In the case when the Fourier Transform is used, the total number of operations is reduced to  $O(MJ \ln(J))$  per iteration.

### 3.14.2 Projection Operation

A convolution mask (with a size of  $K \times K$ ) is used to project the radiance around the medium. Combined with each solid angle segment, the total number of operations in this step is  $O(MJK^2)$  operations. In the event that sparse masks can be used for the convolution, and

assuming there is a mean of  $W$  non-zero elements within these masks, the number of operations reduces to  $O(MJW)$ . For example, if  $W$  is 100,  $W$  is proportional to  $K$ , which is smaller than  $K^2$ .

### 3.14.3 Overall Operations

The overall complexity of MM, assuming sparse masks and one iteration, would be

$$O(MJW + MJ^2) \quad (3-29)$$

In the event that the Fourier Transform is used, its number of operations reduces to

$$O(MJW + MJ \ln(J)) \quad (3-30)$$

Using a grid of 501 x 501 pixels, with 60 solid angle segments, convolution masks that are 51 x 51 pixels in size with 100 non-zero elements, the number of operations needed for this approach is approximately 2.41 gigaflops, or 1.57 gigaflops with Fourier, per iteration. A processor operating at 20 GFLOPS should be able to simulate 2,000 iterations in 241 seconds. In the frequency domain, it is expected that this time can be reduced to 157 seconds. These calculations assume a perfectly optimized simulation software, with no time required for accessing data values or re-arranging them.

### 3.15 Potential for Error Reduction

One of the main advantages offered by MM over MC is that there is little to no standard error in the results. Another advantage involves the speed increase offered by MM when there are many locations to be calculated. This can actually be illustrated mathematically by examining the errors associated with the results of both MM and MC. To find this, the final fluence from each simulation will be determined.

### 3.15.1 Fluence Analysis

The final fluence from these simulations is the sum of the fraction of photons which scatter within the pixel, scaled by their respective weight loss. These sums are scaled by the absorption coefficient. This gives, for any pixel  $i$ , the fluence

$$F[i] = \frac{A[i]}{\mu_a} \quad (J/m^2) \quad (3-31)$$

With  $A[i]$  as the absorbance in each pixel and the absorption coefficient  $\mu_a$  ( $cm^{-1}$ ). The absorbance is calculated differently for each method. In MM, the absorbance is calculated on a per iteration basis. The absorbance for MM can be rewritten as, separating out the weight drop.

$$A_{MM}[i] = \sum_{iter=1}^{\infty} w_{iter} \left[ \sum_{j=1}^{Jmax} Radiance_{iter}[i,j] \right] \quad (3-32)$$

The weight drop is represented as  $w_{iter}$ . MC absorbance is given by

$$A_{MC}[i] = \sum_{pht=1}^{\infty} A_{pht}[i, w] \quad (3-33)$$

With  $w$  representing the weight drop of the photon. As the photons can be represented on a per weight iteration (i.e. all photon packets which lose the same weight are one group), this sum can be represented as the sum of the absorbance per weight group per photon, yielding

$$A_{MC}[i] = \sum_{iter=1}^{\infty} w_{iter} \left[ \sum_{pht=1}^{\infty} N_{pht}[i, iter] \right] \quad (3-34)$$

### 3.15.2 Time & Error Analysis

In eq. 3-34 the absorbance has been changed to a multiplication between the sum of the number of photons which scatter within a medium and the corresponding weight drop. This has been summed over all of the weight drops that occur. Comparing eq. 3-34 to eq. 3-32 leads to the conclusion that the difference between the simulation time of MC and MM involves the time required to calculate the radiance and the number of photons per iteration. The theoretical calculations required to determine the radiance for MM were shown in eq. 3-29. For MC, this required time will be discussed.

For MC, each photon is expected to take the same amount of time per scattering and absorption event. This is dependent on the time for each part of the procedure in MC. The total number of photons which are needed depends on the desired accuracy. As MC depends on random sampling, there's a standard error associated with its results. Each pixel would need to have enough samples to accurately predict the fluence.

To compare the errors from both of these methods, there is a true value for the absorbance, which is assumed to be the absorbance after infinitely many photons packets have been used. As this is obviously impractical, the absorbance recorded by MC has an intrinsic error due to a lack of samples. The absorbance from one of these weight iterations to pixel  $a$  is

$$A_{[a,iter]} = \sum_{pht=1}^{\infty} N_{pht}[i, iter - 1] * p(i \rightarrow a) \quad (3-35)$$

with  $p(i \rightarrow a)$  as the probability that a photon from pixel  $i$  undergoes scattering and absorption within pixel  $a$ . The value of the probability changes depending on the distance and

the direction of the initial photon. The lack of samples typically leads to errors in pixels far from the source, as both the number of photons remaining with low weights and the probability of those pixels reaching the distant pixels is small.

For MM, it is also impossible to conduct infinitely many iterations, however MM reaches a point when increasing the number of iterations has no effective impact on the final results, simply due to the decrease in the remaining radiance.

The total error from MM ( $\epsilon_{MM}$ ) can be represented by a double summation involving each pixel within the medium, with the errors involving each projection and scattering event.

$$\epsilon_{MM} = \sum_{iter=1}^{\infty} \sum_{i=1}^{pixels} Error[i, iter] \quad (3-36)$$

For MC, the errors are assumed to be from the standard error  $SE_{MC}$ . The standard error decreases proportionally to the square root of the number of samples taken [100]. If 100 times more samples are used, the standard deviation decreases by 10 fold.

$$SE_{MC} = \sum_{pht=MC+1}^{\infty} \sum_{i=1}^{pixels} \left| \frac{s[i, pht]}{\sqrt{n[i]}} \right| \quad (3-37)$$

Both of these errors depend on the number of pixels which are present. This is important as the number of operations in MM is proportional to the number of pixels, whereas the number of operations for MC is proportional to the square of the number of photons needed per pixel measured. For simulations with a small number of pixels, MC is considerably better. When there are a large number of pixels, or there are many pixels depending on large amounts of small weight photon packets, MM is superior.

### 3.15.3 Error Comparison

MM can be understood as calculating the result of launching all photons as one ensemble into a medium. MM operates similar to MC, except that there is no standard error issues due to sampling. Without the random sampling components, there is no variance issues in the results of any simulation. If there are errors, it will be from the approximations made to derive the method. These will appear as systematic errors, which may be addressed with future modifications.

### 3.16 Overall Block Diagram

To best convey MM, a series of block diagrams have been constructed to illustrate a possible process for the method. The values and equations used in the diagram assume an environment consistent with the medium seen in this thesis. This block diagram will be divided into four parts, being Declare Variables (Fig. 3.17), Create Projection Masks (Fig. 3.18), Creating Scattering Matrix (Fig. 3.19) and Radiance Propagation (Fig. 3.20).



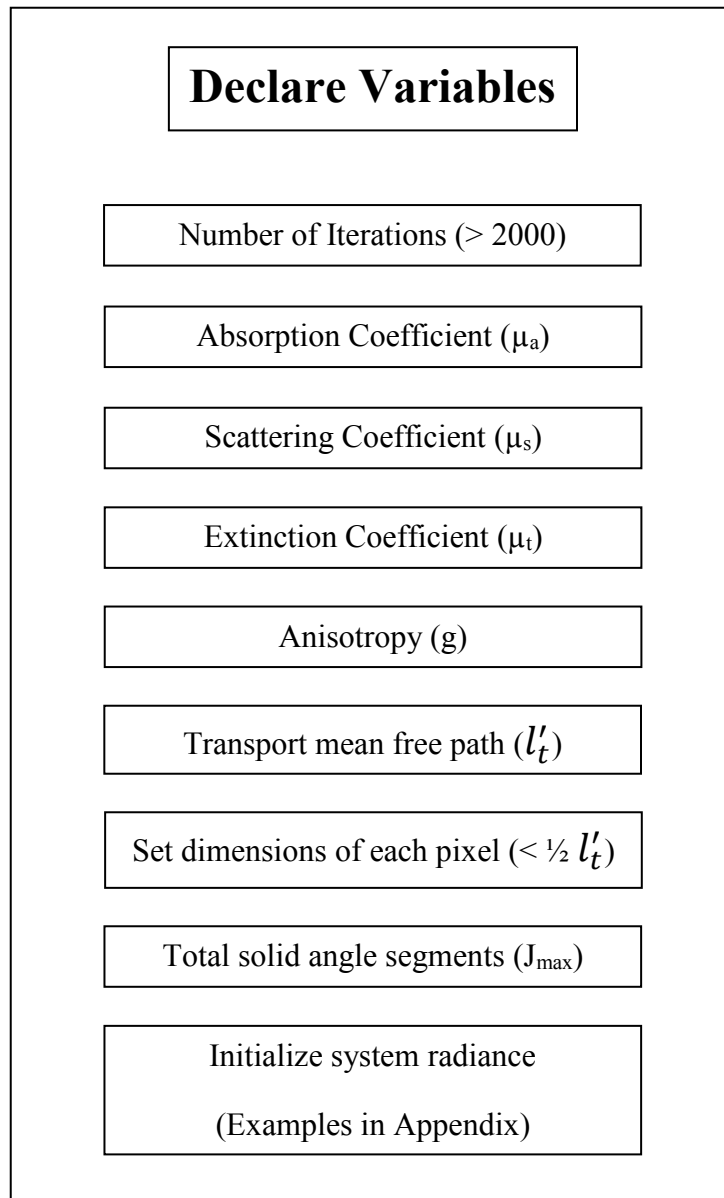


Figure 3.17. Block diagram for Declare Variables section.

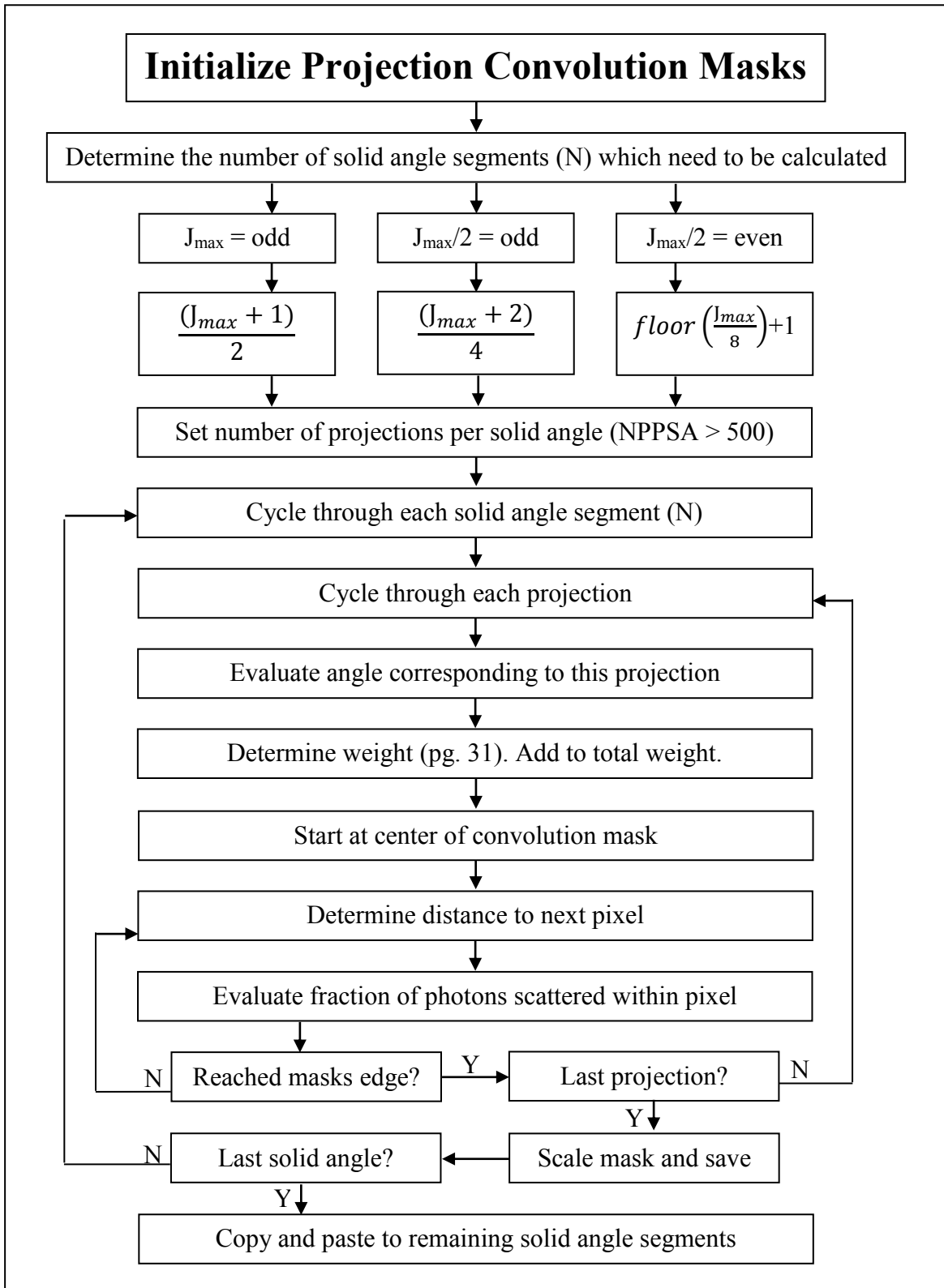


Figure 3.18. Block diagram for Create Projection Masks section.

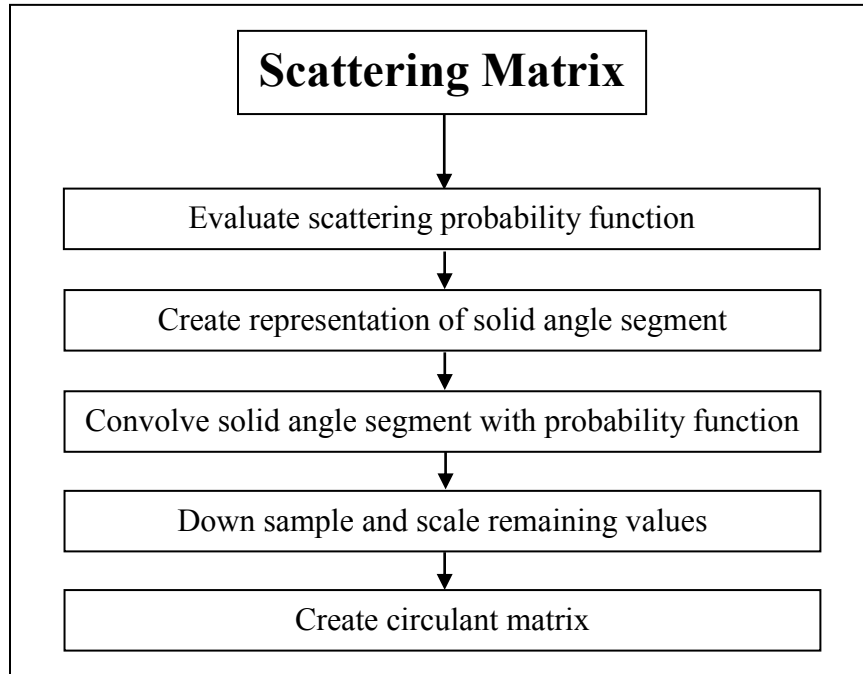


Figure 3.19. Block diagram for Create Scattering Matrix section.

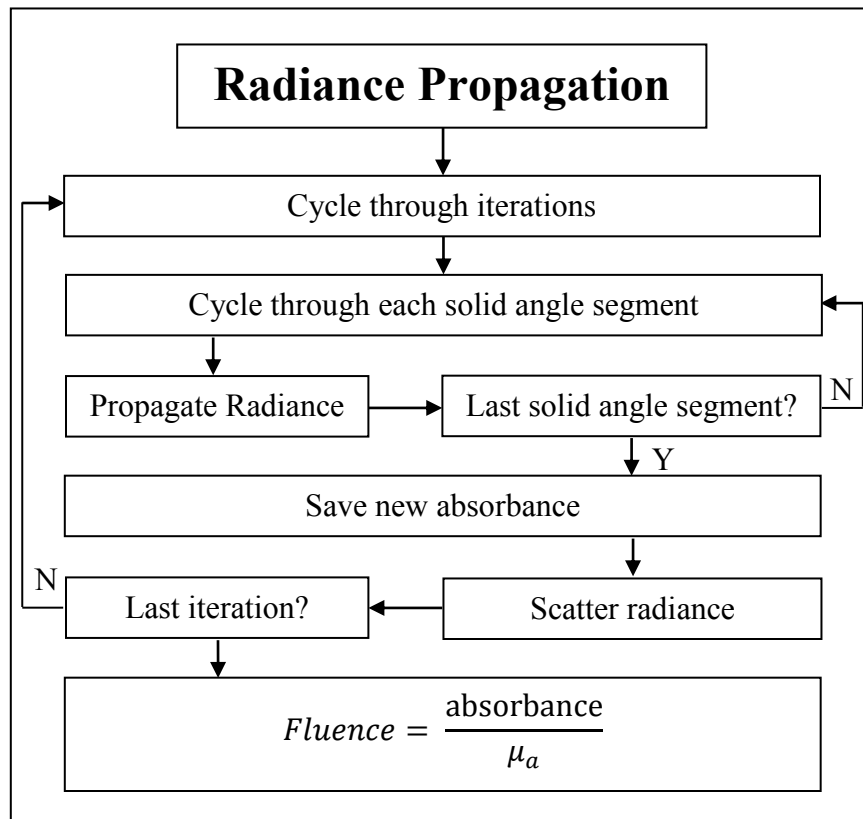


Figure 3.20. Block diagram for Radiance Propagation section.

# Chapter 4

## Matrix Method Results

Matrix Method (MM) is expected to show an increase in the overall computational efficiency in calculating light propagation compared to Monte Carlo (MC). It is also expected to greatly reduce the statistical variability in simulated results compared to MC. To confirm these hypotheses two classical light propagation simulations will be done using MC and MM. These are the isotropic point source in an infinite turbid media and the pencil beam in a semi-infinite turbid media. With each simulation two metrics of error analysis will be used, being Mean Square Error (MSE) and Mean Absolute Percent Error (MAPE). The results of these tests and their errors will be discussed. Methods to reduce the error in each of these steps will also be illustrated. A table of times measured for MC and MM simulations will be included.

## 4.1 Introduction

The two main simulations within this thesis are the isotropic point source in an infinite turbid media and the pencil beam in a semi-infinite turbid media. To compare these two simulations with their MC counterparts several tests will be done to evaluate the efficiency of MM. The first tests will involve analyzing the error between these two methods. Both MSE and MAPE will be used for the error analysis. MSE is used as its results can show the error between larger values, while MAPE is used to show the differences between smaller values.

### 4.1.1 Mean Square Error

The error will be analyzed in terms of the MSE between the fluence of MM and MC. The MSE in these cases will be evaluated by using the following equation [102]

$$MSE = \frac{1}{n} \sum_{i=1}^n (MM_i - MC_i)^2 \quad (4-1)$$

An average of MC results will be used as the true values for the MSE (represented by  $MC_i$ ),  $MM_i$  will represent the values from the MM where the n different pixels in the medium will be compared. The iterations of the MM create a fluence distribution, and each iteration is then compared to the final values from the MC simulation. This will generate an estimate for how many iterations are necessary to effectively calculate the larger values within a medium. If this number is small, it is possible that other methods which increase the efficiency and speed of MC can also be applied to further reduce the computation time (i.e. GPU). In future simulations (not included here), if an analytical solution exists it will be used instead of MC as the true values. One example is with the isotropic point source in an infinite turbid media (eq. 2-21).

## 4.1.2 Mean Absolute Percent Error

As many of the results of these simulations will have a wide range of values which differ by several orders of magnitude, it is expected that the errors from MSE will be biased towards certain values in the simulation. These are the values associated with the locations closest to the source, as they will be the largest. To examine the values further from the source another less commonly used means of error analysis will be used. This is MAPE [102] (eq. 4-2)

$$MAPE = \frac{100 \%}{n} \sum_{i=1}^n \left| \frac{MM_i - MC_i}{MC_i} \right| \quad (4-2)$$

The variables for MAPE are identical to those used in MSE. This error definition can be used so long as there are no pixels with a fluence of zero from MC.

## 4.1.3 Time Analysis

The overall efficiency isn't just determined by the differences between values. It is also determined by the time it takes to complete each simulation to reach an adequate level of accuracy. Comparing MM to MC means estimating the number of photons (or additional computational time) needed to reduce the stochastic error from MC to less than the expected bias from MM.

With the isotropic point source and pencil beam simulations,  $10^8$  photon packets for MC will be compared to 3000 iterations for MM. The number of iterations for MM was determined experimentally while the number of photons is from the upper limit of current literature [73].

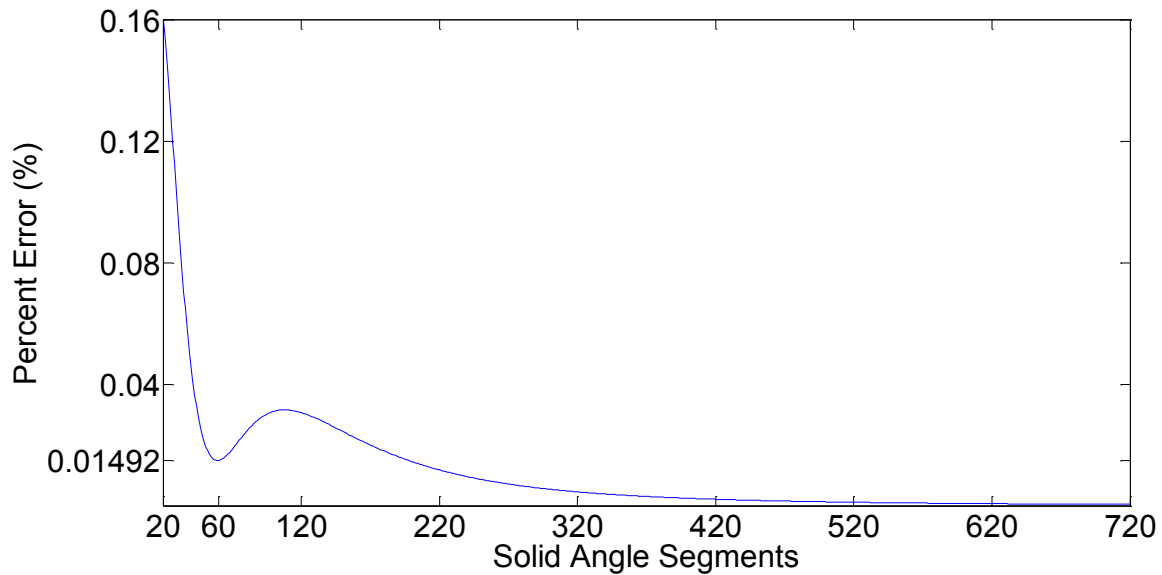


Figure 4.1. The Mean Square Error between the modified HG function and it's linearly interpolated version.

#### 4.1.4 Simulation Parameters

In each simulation there are several parameters which are kept constant. The absorption coefficient  $\mu_a$  is set to  $0.1 \text{ cm}^{-1}$ , the scattering coefficient  $\mu_s$  is  $100 \text{ cm}^{-1}$  and the anisotropy factor  $g$  is 0.9. These are common in biological tissue [28, 32].

The pixels are kept to the same size regardless of the simulation. In order to avoid cytoplasmic heterogeneities (different values for the extinction coefficient throughout the cell) the pixels must be larger than the cell, but small enough so as to minimize any errors associated with MM approximations. The pixel size was set to  $40 \mu\text{m} \times 40 \mu\text{m}$ .

Since media can be infinitely large, it seems practical to limit the size of the simulations to those within a small range. The vast majority of the photons will be absorbed in a couple centimeters of the initial source. The total size of the media in every simulation was set to  $2 \text{ cm} \times 2 \text{ cm}$ .

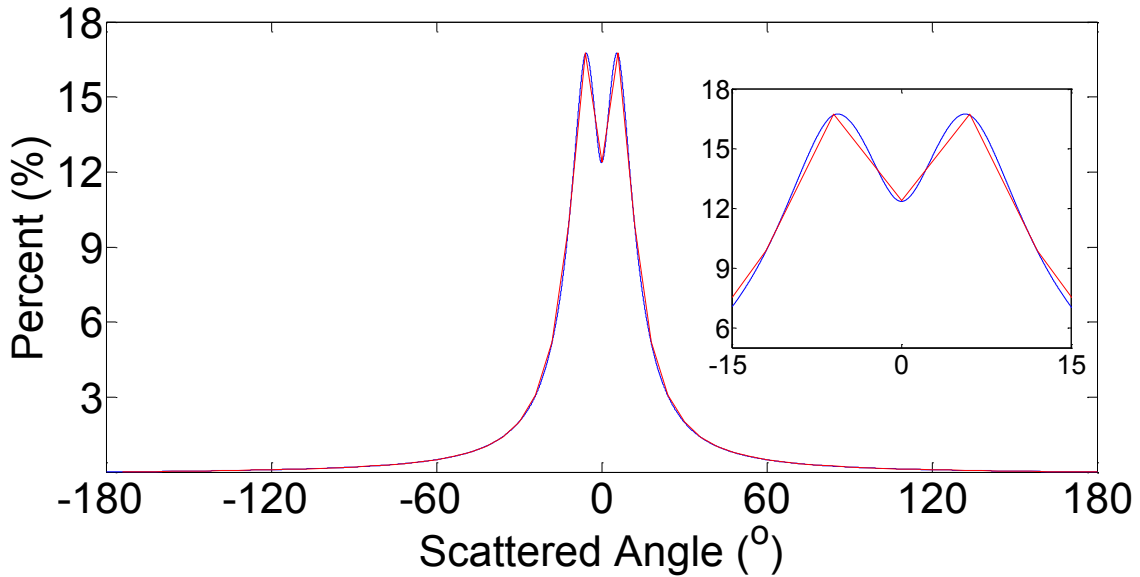


Figure 4.2. The modified HG function and it's linearly interpolated version over 360°. Insert ranges from -15° to 15°.

In MC, the point source initially uses 12,500,000 photon packets, which is then averaged so the results effectively use 100,000,000 photon packets. The pencil beam uses 50,000,000 photon packets, and can also use averaging to bring the effective total to 100,000,000. This is accomplished through symmetry. The sizes of the masks in MM are determined by the ratio of the initial to final intensity along a single line. Here, that ratio is set to  $5 \times 10^{-5}$ , leading to masks which are 51 x 51 pixels.

Each simulation will undergo 3000 iterations before the fluence will be finalized. The initial radiance ( $W/m^2 \cdot sr$ ) in the isotropic point source and initial intensity ( $W/m^2$ ) in the pencil beam simulation is set to a value of 1, so as to make these results simpler to analyze and compare. The simulation software MATLAB 2013b is used to run the programs. These simulations are run on an Intel core i7 970 processor with 24 gigabytes of RAM using a 64-bit Windows 7 operating system. For both MC and MM, multithreading options will be used for each and every simulation as the relative accuracy of these methods is analyzed.



Due to the fact that methods for parallel processing are likely to change over time, the time measurements will serve for commentary only. The time measurements are expected to show a significant difference between MC and MM.

### 4.1.5 Number of Solid Angle Segments

The number of solid angle segments is determined by error analysis. Figure 4.1 shows the error between the modified HG function and the linearly interpolated function generated using these points. Here, the minimum error within the first 120 points occurs with 59 solid angle segments (Error of  $1.491 \times 10^{-2}$  %). 60 solid angle segments is preferred, however, as it allows for backscattering to be considered with only a slightly larger error ( $1.492 \times 10^{-2}$  %).

198 solid angle segments must be used for the error to be less than  $1.491 \times 10^{-2}$  %. With 198 solid angle segments, the simulation time would increase by 7 fold. Even if the Fourier method is applied, this would still lead to an increase of a factor of 3.4. This considerable increase in simulation time, however, does not lead to an increase in the accuracy.

Less than 20 solid angle segments will not be considered as the percent errors would increase. Aliasing would occur, which are errors due to a lack of solid angle segments. Aliasing errors would cause an incorrect radiance distribution after every scattering iteration. More than 720 segments were not considered, as the increase in computational time would have little to no effect on reducing the error. Therefore, the number of solid angle segments will be 60.

Figure 4.2 is an image of the modified HG phase function and the linearly interpolated version with 60 points. The range from  $-15^\circ$  to  $15^\circ$  has been magnified to show the more significant values. As can be seen, there is little difference between the two functions.

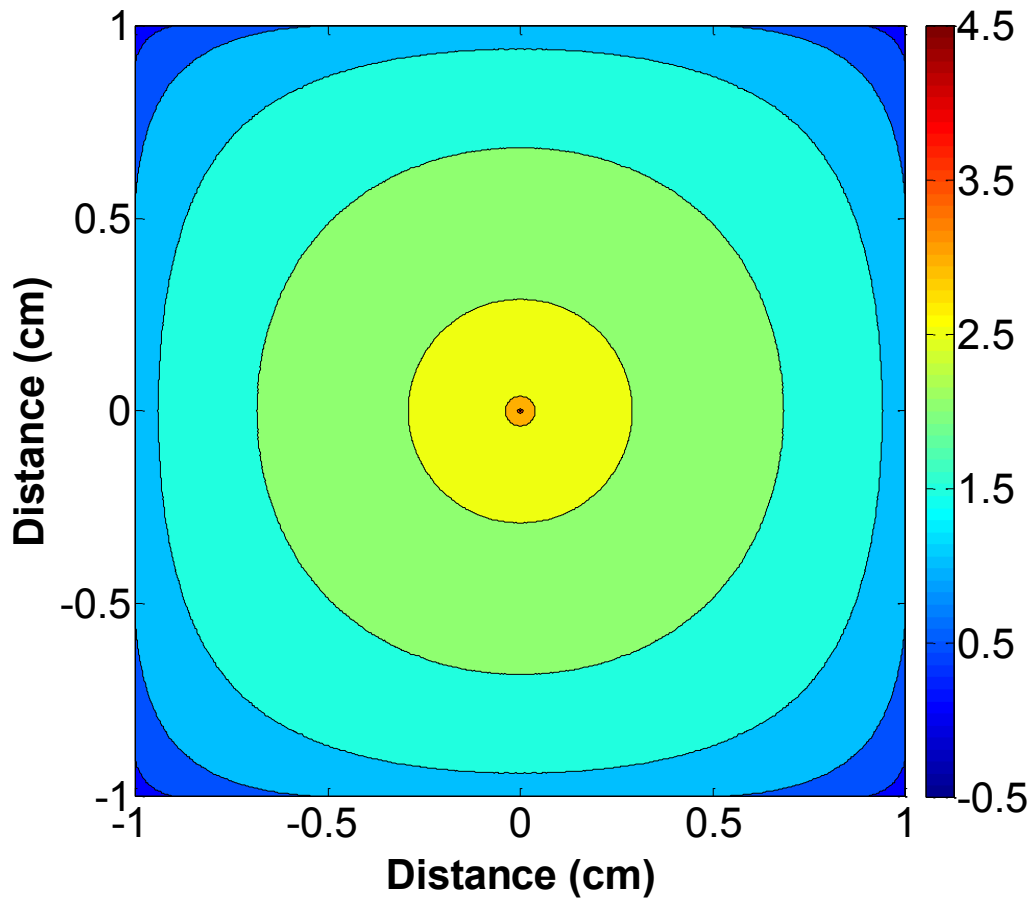


Figure 4.3. The fluence ( $J/m^2$ ) distribution result of the MC simulation with the isotropic point source in an infinite turbid media. Results are in logarithmic (base 10) scale.

## 4.2 2D Isotropic Point Source Results

### 4.2.1 Monte Carlo Results

In the MC simulation the grid was capped at a size of 501 by 501 elements, with the center of the grid as the initial point source. The results after 100,000,000 photons packets were used can be seen in figure 4.3. This is the control, or gold standard, to which the MM simulations will be compared to. The rounding effect near to the boundaries is due to the termination of photon propagation once they leave the grid. Boundary pixels can be used to mediate this effect.

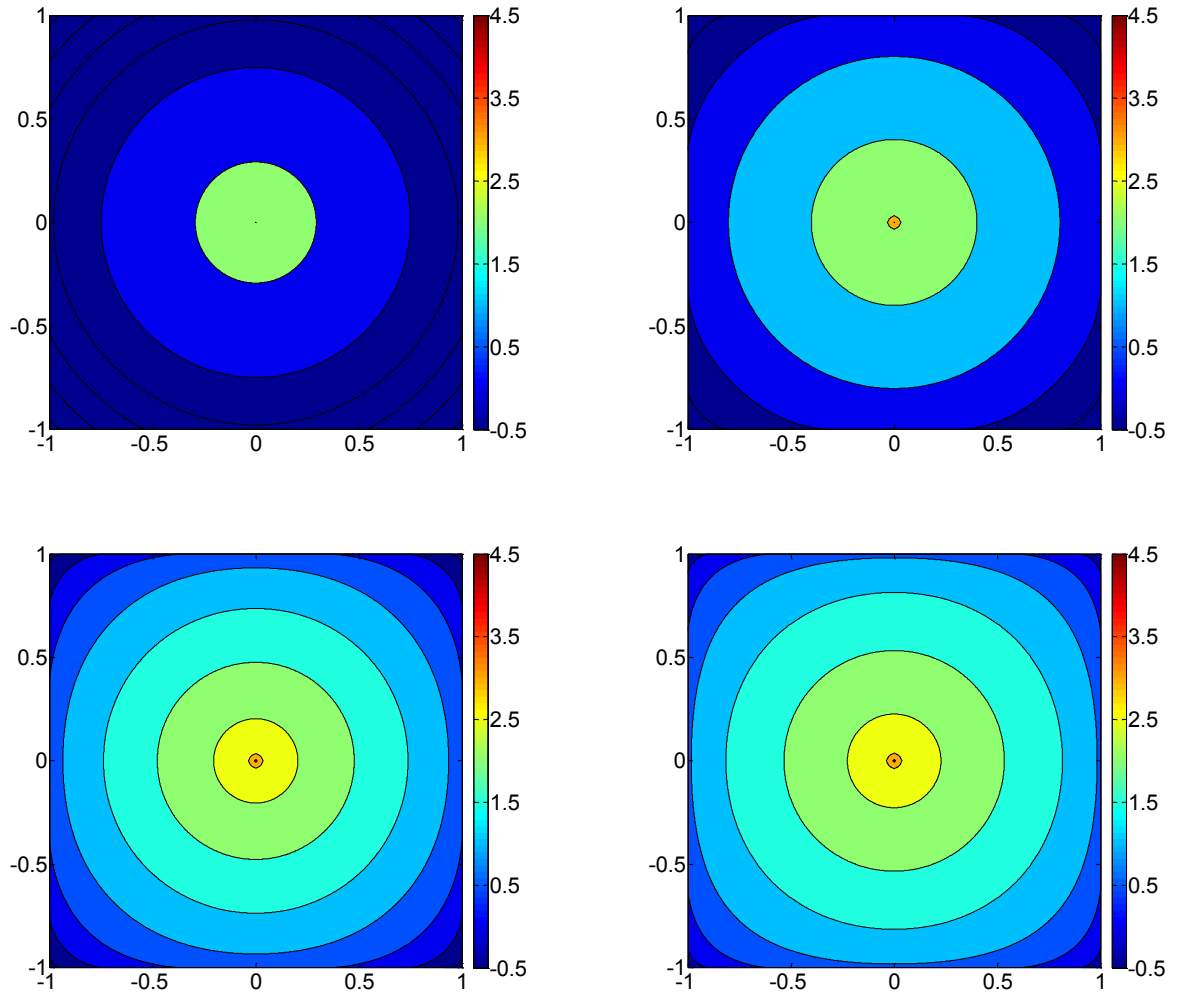


Figure 4.4. The fluence distributions from MM involving an isotropic point source in an infinite turbid media. These are displayed logarithmically (base 10) to ease comparisons. From top left to bottom right we have 100, 200, 300 and 400 iterations. The axes outline the distances of the medium, in cm.

## 4.2.2 Matrix Method Results

In this simulation a single isotropic point source is placed at the center of a 501 by 501 grid. Here, the initial radiance ( $W/m^2 \cdot sr$ ) is set to a value of 1, which is evenly divided between each of the solid angle segments. The fluence ( $J/m^2$ ) after 100, 200, 300 and 400 iterations can be seen in figure 4.4. Each of these images displays an expansion of the fluence as the iterations progress.

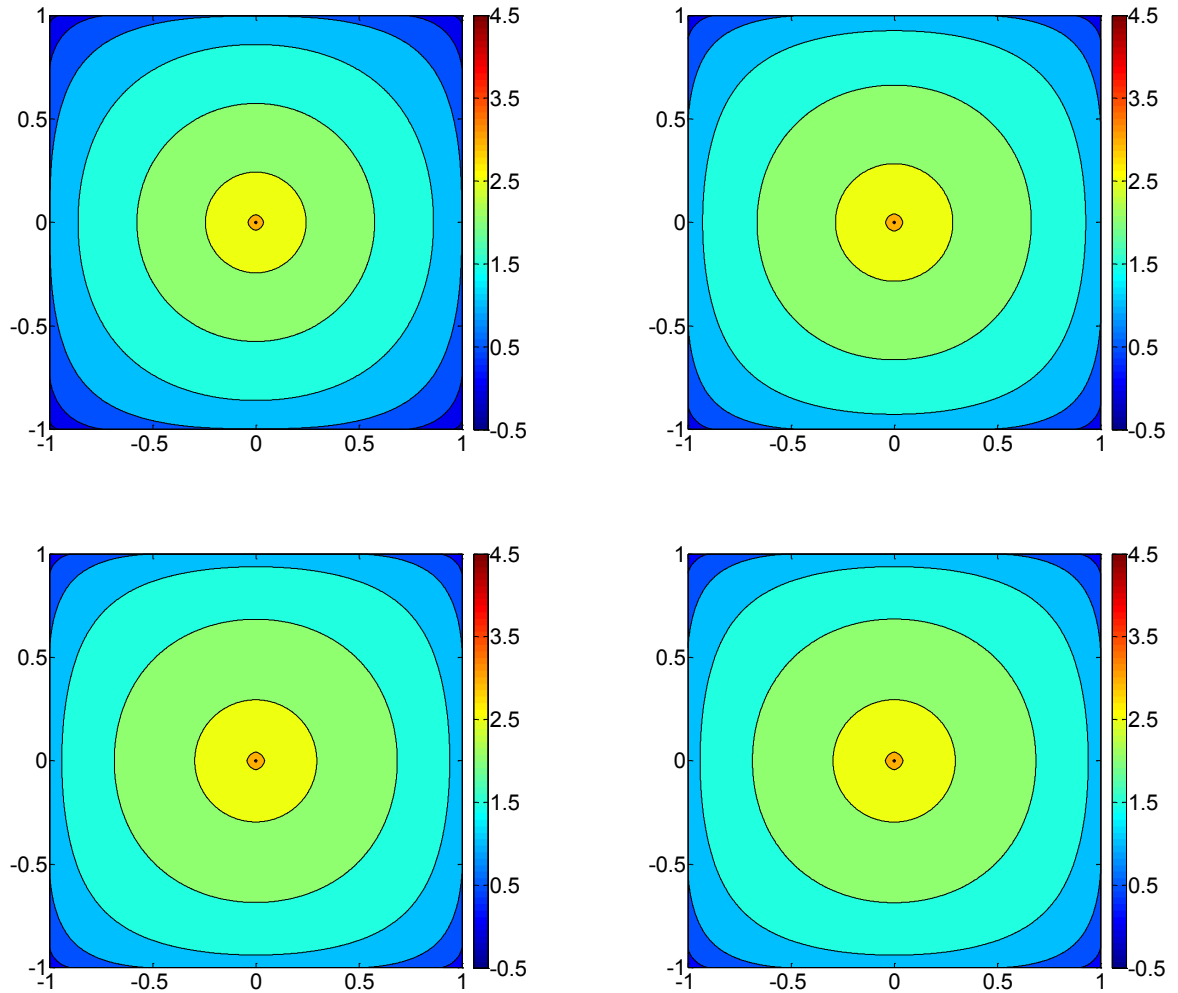


Figure 4.5. The fluence distributions from MM involving an isotropic point source in an infinite turbid media. These are displayed logarithmically (base 10) to ease comparisons. From top left to bottom right we have 500, 1000, 2000 and 3000 iterations. The axes outline the distances of the medium, in cm.

Figure 4.5 displays the images for the expansion of the fluence ( $J/m^2$ ) after 500, 1000, 2000 and 3000 iterations. After 1000 iterations there are only small changes in the overall fluence, which are attributed to the significant decrease in the remaining radiance. The radiance is reduced by both absorption in each iteration and by the fraction which leaves the medium. There are also noticeable differences between the center of the image and the edges. This is due to the boundary effects. Beyond the boundary the radiance is simply set to zero. If pixels near a boundary are required, a larger field of view can be used if the central results are taken.

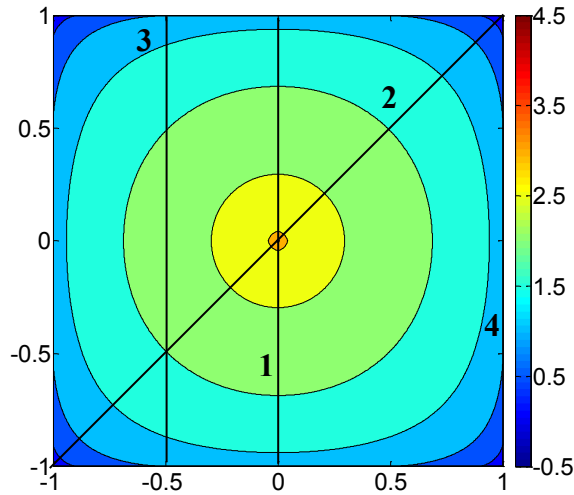


Figure 4.6. Numbered cross sections used in analysis. The axes outline the distance, in cm.

Traditionally, MC is known for simulating the isotropic point source in an infinite medium. However, as MM is a grid based method there are limits to the amount of memory available which can be used. This means the MM will only simulate the isotropic point source within a finite medium.

Examining the results, one can see that the values closer to the source term reach their final values faster than those further away. This is expected as the amount of energy which is available in the medium slowly decreases after each iteration. The simulations from both MC and MM seem comparable, however this will be confirmed with cross sectional analysis and error analysis.

### 4.2.3 Cross Section Results

Here, four main cross sections will be assessed (Fig. 4.6). Firstly, the central axis which crosses the source term. Secondly, the diagonal axis crossing the central source. Thirdly is the vertical line midway between the source and the edge. Fourthly is the edge of the medium.

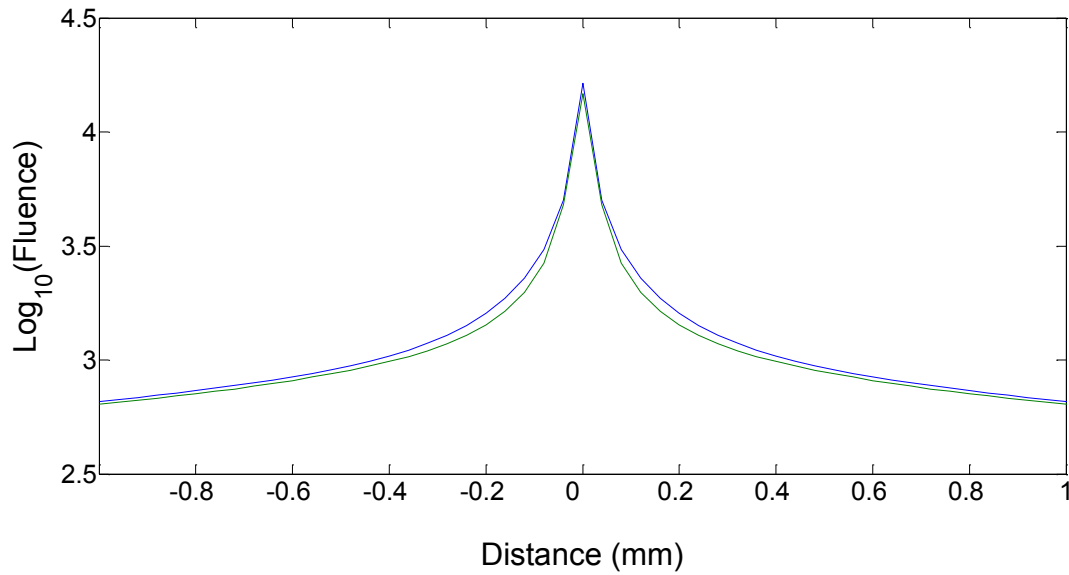
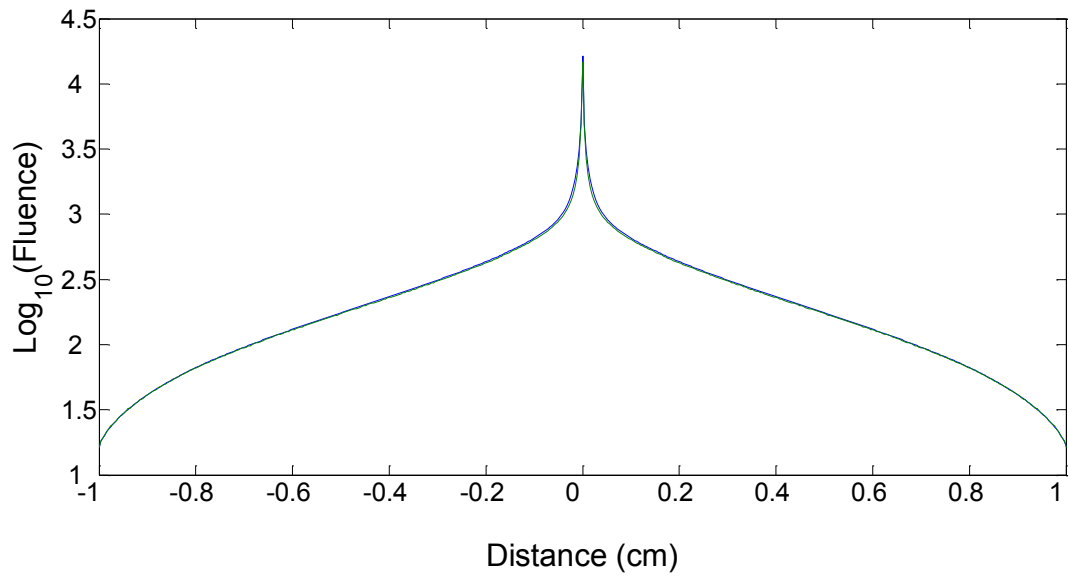


Figure 4.7. The vertical cross section of the fluence through the source term. Blue is MM and green is MC. The upper plot represents the entire range, while the lower plot is the region within 1 mm of source term.

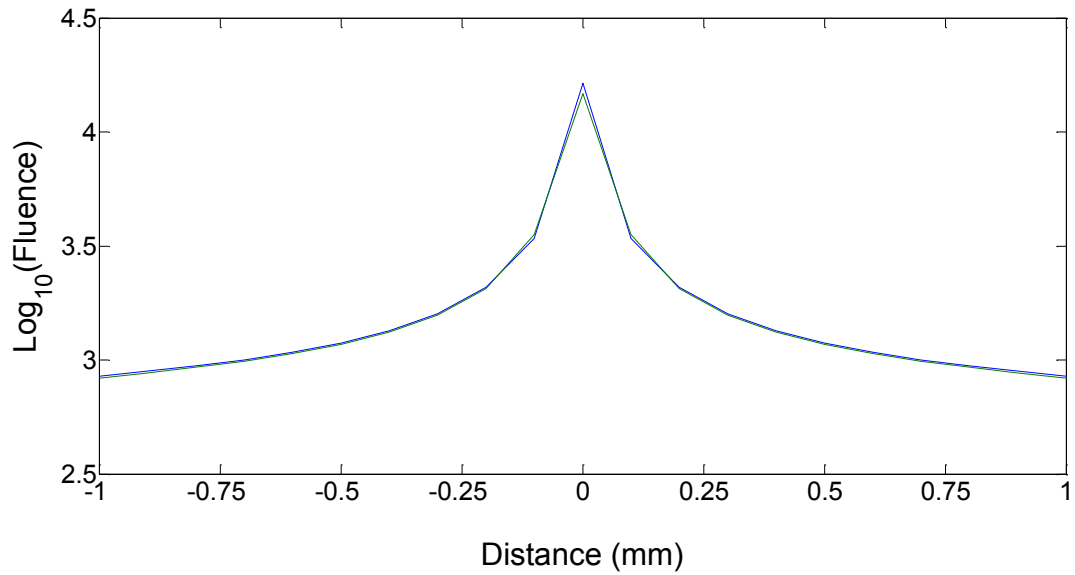
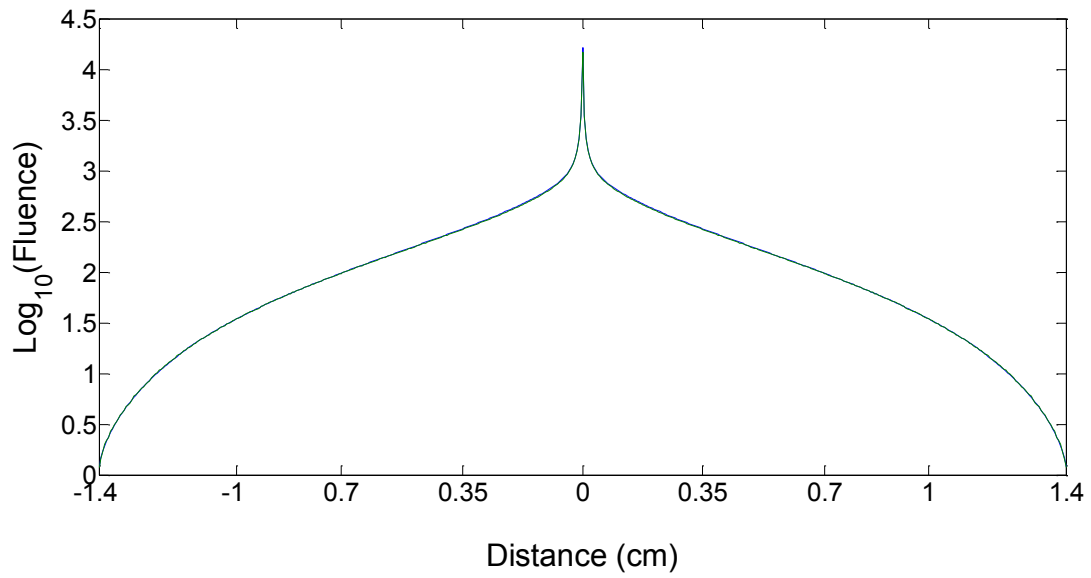


Figure 4.8. The diagonal cross section of the fluence through the source term. Blue is MM and green is MC. The upper plot represents the entire range, while the lower plot is the region within 1 mm of source term.

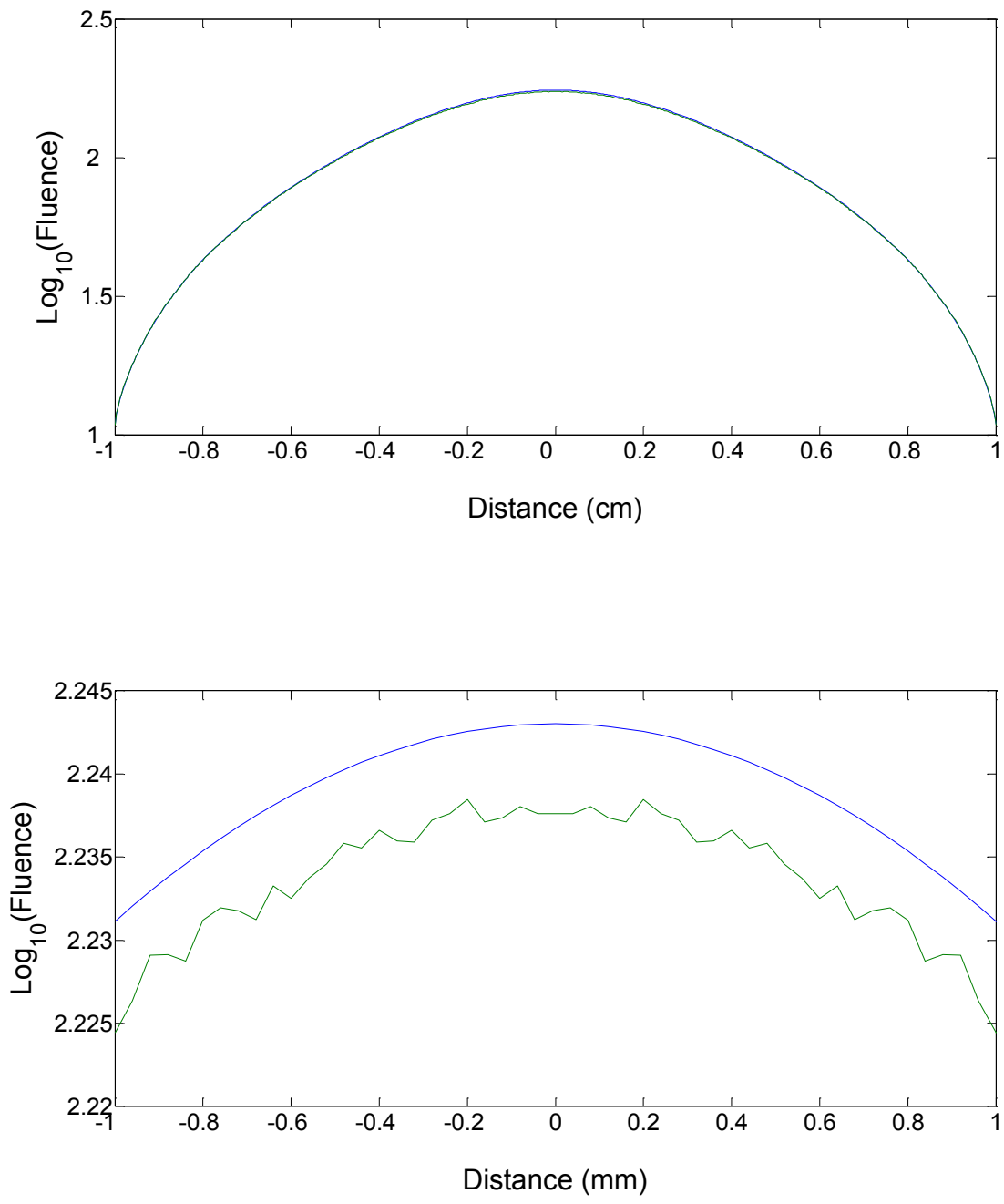


Figure 4.9. The cross section midway between the source and the edge. Blue is MM and green is MC. The upper plot represents the entire range, while the lower plot is the region within 1 mm of the cross sections' center.



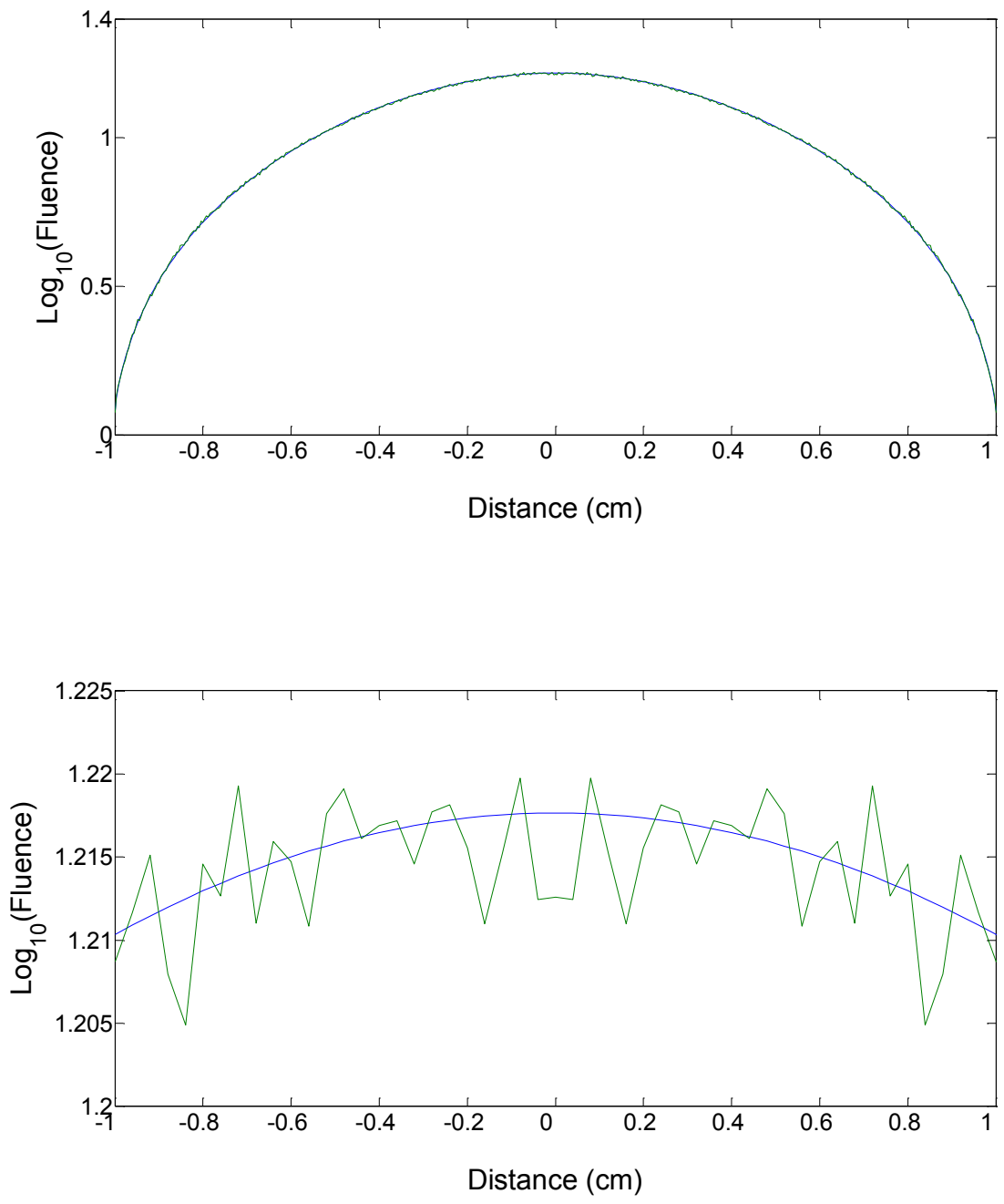


Figure 4.10. The cross section of the edge of the medium. Blue is MM and green is MC. The upper plot represents the entire range, while the lower plot is the region within 1 mm of the cross sections' center.

The vertical central axis is shown in figure 4.7. From the first figure the two methods show a highly similar fluence. They each have a high central peak corresponding to the pixel with the source term, followed afterwards by a sharp decline. The rate of this decline rapidly decreases in amplitude, reaching close to zero after the fluence is 0.5 mm away from the source. The boundary effects occurring 1 cm away from the source term cause identical effects with both methods. This leads to the conclusion that extra boundary pixels would effectively address any boundary issues within the medium in both simulations.

These results can also be verified with the diagonal cross section (Fig. 4.8). Here the same overall shapes from figure 4.7 can be easily seen. The most significant difference between these two plots involves the points nearest to the source term. Along the diagonal the values are slightly more accurate than across the vertical axis. The boundary effects can be seen too, with both MC and MM experiencing the same impact.

In the case of the final two cross sections (Fig. 4.9 and Fig. 4.10) the striking difference as compared to the other two cross sections involves the accuracy near the center of the plots. Here, both MM and MC seem to come to the same results. MC still shows random errors at these distances, while MM has no random effects whatsoever.

#### 4.2.4 MAPE & MSE Results

The cross section analysis allows us to see some of the values within the medium. However, the only way to be certain of the results is to use both MSE and MAPE as means of determining the overall error. These were calculated during the simulation, and are plotted in figure 4.11.

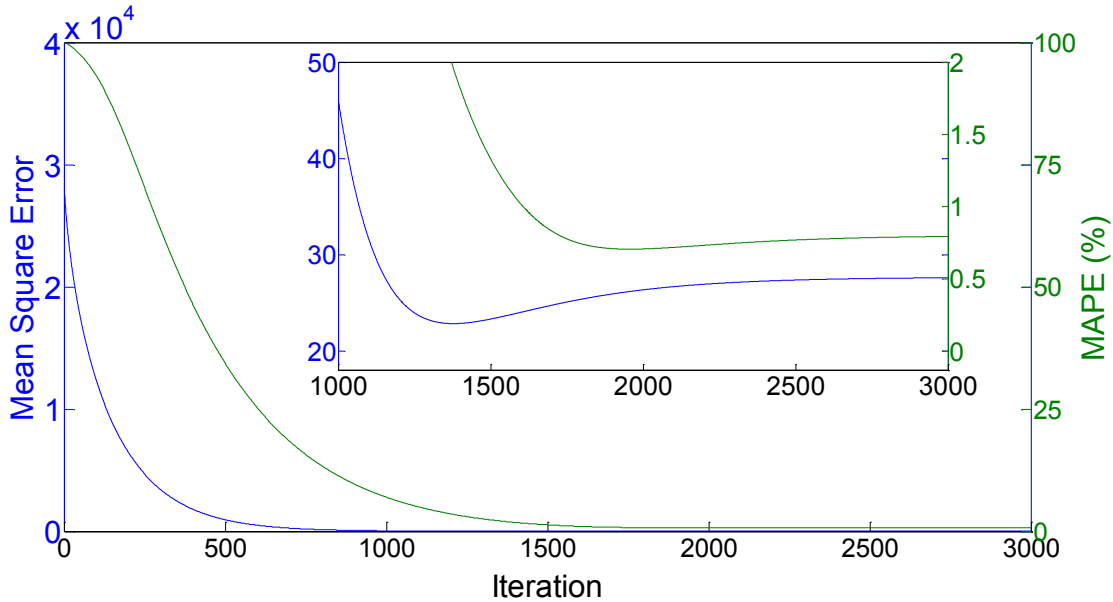


Figure 4.11. MSE (blue) and MAPE (green) of the MM isotropic point source simulation with respect to MC. The insert ranges from 1000 to 3000 iterations.

Both the MSE and the MAPE show a sharp decrease as the number of iterations grow. This decrease is steady for MSE until 1378 iterations, where it reaches a minimum value of 22.84. The MSE then slightly increases, reaching 27.60 after 3000 iterations. This trend is similar to the trend seen with MAPE. MAPE reaches a minimum value of 0.7062 % after 1957 iterations, then it increases to 0.7946% after 3000 iterations.

MSE was used to illustrate the accuracy of the values closest to the source. It decreases by 1024 fold over the course of the simulation. This is strong evidence that the results MM produces near the source are comparable to those from MC.

This evidence is enhanced by the results of MAPE. Here the error decreases by 126 fold during simulation. As MAPE reflects the results further from the source, this decrease implies the results further from the source may be comparable as well. MSE and MAPE increase after reaching minimums since MC has few photons remaining after 1,700 absorption and scattering events. Using more photons in MC would prevent the increases seen in MSE and MAPE.

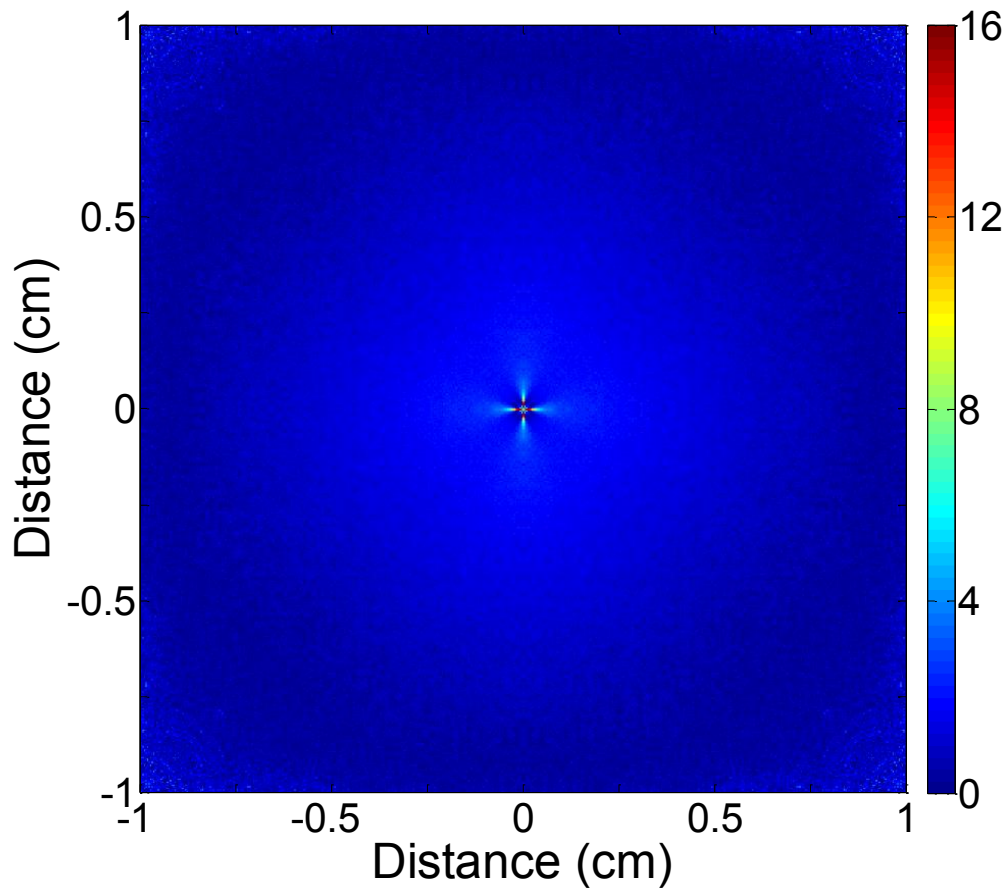


Figure 4.12. The absolute percent error in the isotropic point source between MM and MC.

### 4.2.5 MAPE Plots

To better understand which the pixels have the most significant impacts on MAPE and MSE, the absolute percent error is shown (Fig. 4.12). Two locations mainly contribute to the magnitude of these errors. They are in the corners and near the source. In the corners, less photons scatter here during MC simulations. Errors there are likely due to a lack of samples, leading to an increase in the standard error. To determine if these errors are due to MC, errors of less than 2% will be removed (Fig. 4.13), and a scale from 2% to 4 %, and from 4 % to 6 % will be included. Errors larger than 6% will appear white.

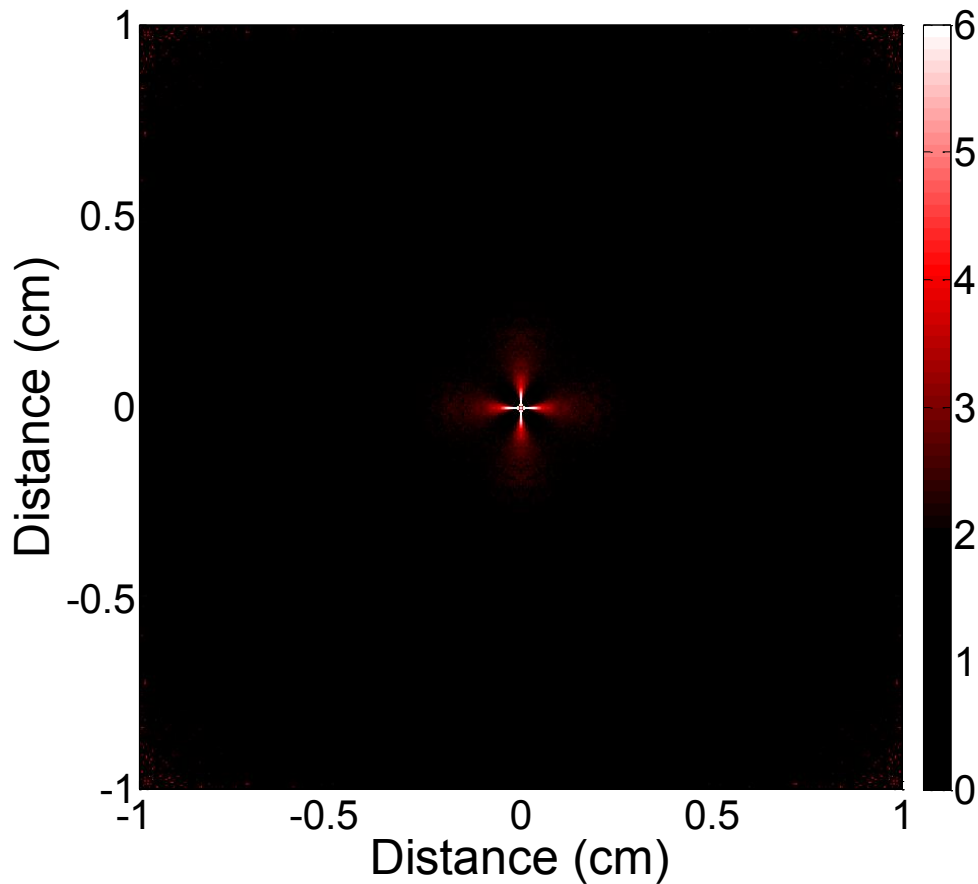


Figure 4.13. The absolute percent error in the isotropic point source between MM and MC. Errors larger than 2% are shown. A scale beginning at 2 % (black) to 4 % (red) and to 6 % (white) is included for better contrast.

A significant number of the corner errors have been removed. As the errors there sparsely populate the corner, any further errors in the corners are expected from a lack of photons reaching this region. Figure 4.10 helps confirm this as the standard error from MC can be easily seen. The significant errors which are assumed to be affiliated with MM are the clover (red) and the cross (white) (Fig. 4.13). Errors in this region are critical for MM as other methods (i.e. Diffusion Theory) can be used to predict fluence far from a source term. To confirm that these errors are affiliated with MM, the pencil beam source will be simulated. If these errors are affiliated with MM, it is likely that they will be present in the pencil beam simulation as well.

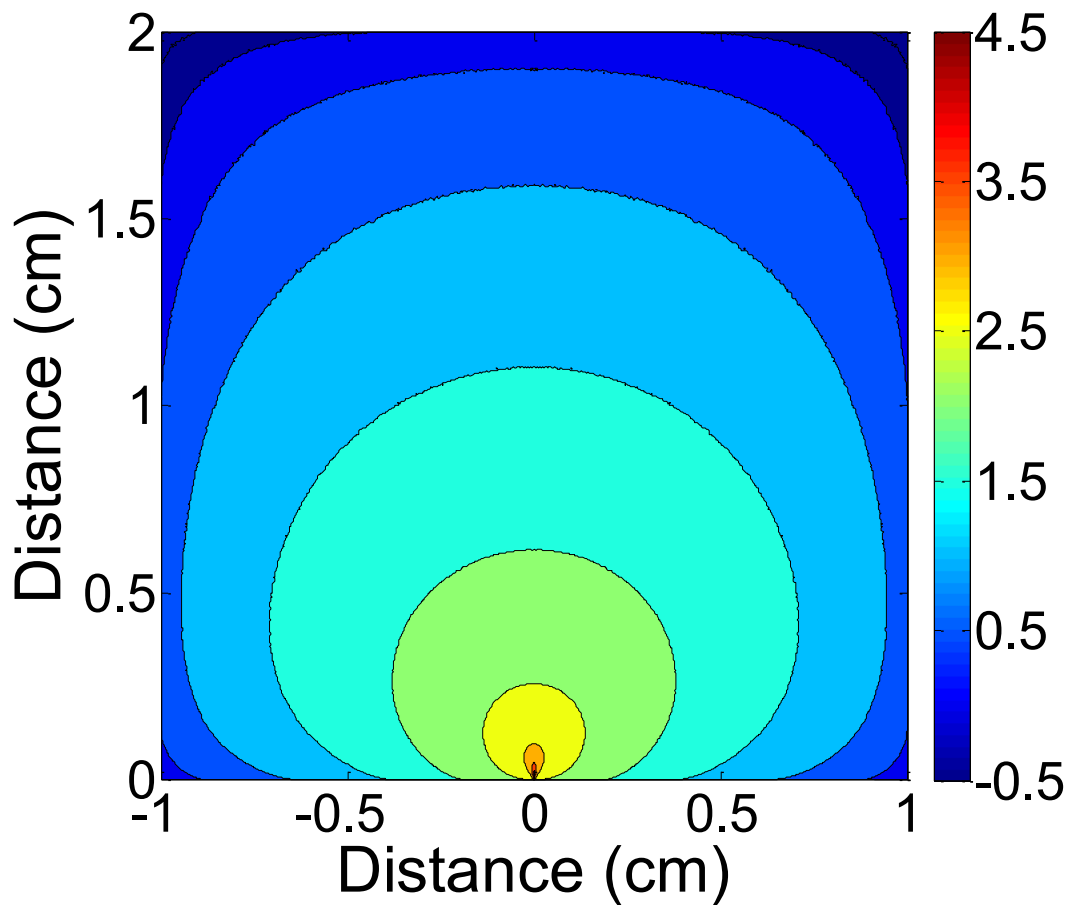


Figure 4.14. The fluence ( $J/m^2$ ) distribution result of the MC simulation results of MC simulation of a pencil beam incident on a surface of a semi-infinite turbid media. Results are in logarithmic (base 10) scale.

## 4.3 2D Pencil Beam Results

### 4.3.1 Monte Carlo Results

In the MC simulation the grid was again capped at a size of 501 by 501 elements, with the center of the bottom edge as the initial source. The results with 100,000,000 photons packets are shown in figure 4.14. As with the isotropic point source, this is the control to which the MM simulations will be compared to.

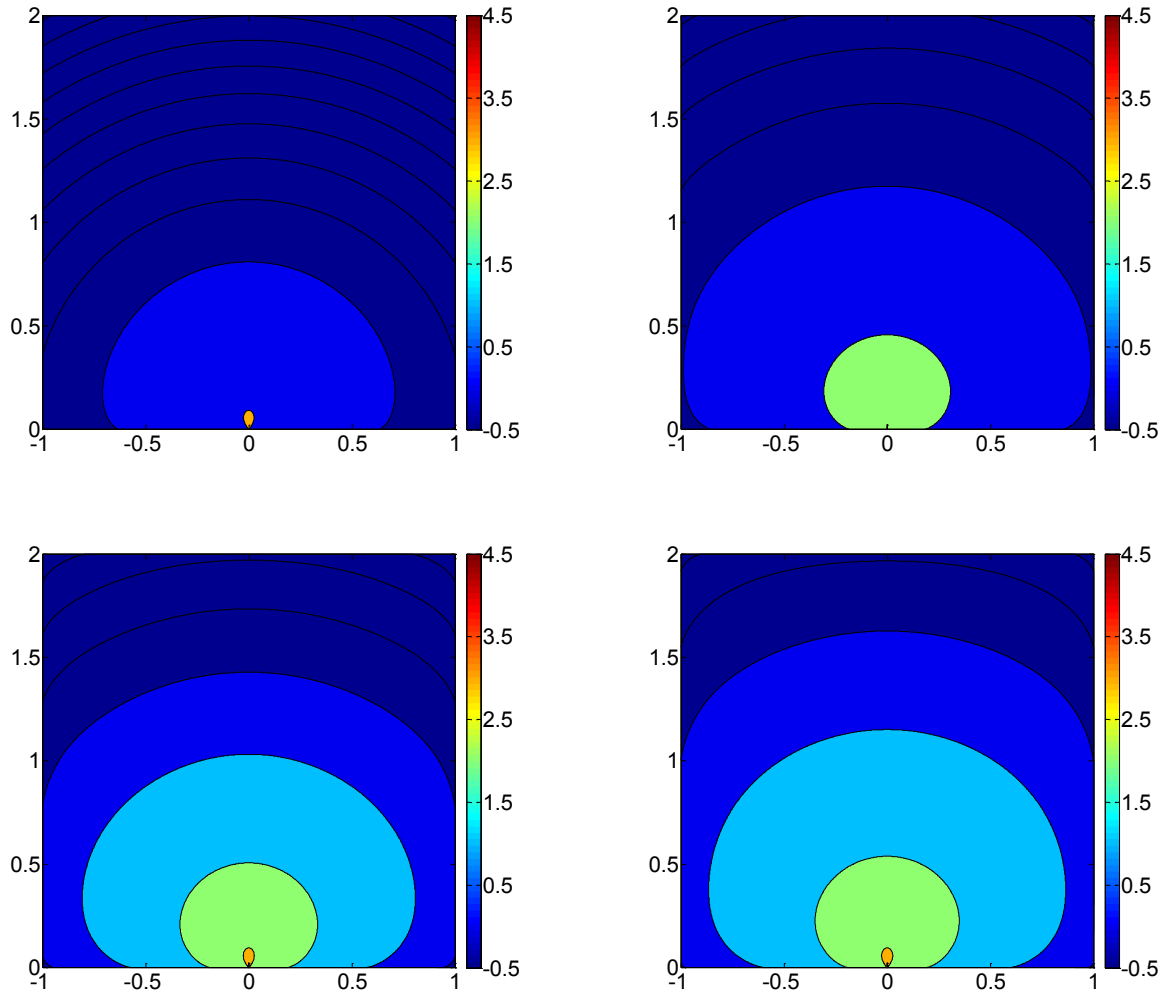


Figure 4.15. The fluence distributions from MM involving a pencil beam source incident to a semi-infinite turbid media. These are displayed logarithmically (base 10) to ease comparisons. From top left to bottom right we have 100, 200, 300 and 400 iterations. The axes outline the distances of the medium, in cm.

### 4.3.2 Matrix Method Results

Here, a pencil beam of light is sent through a medium across its center. The initial intensity that is sent into the medium was set to a value of 1. The spread across the central axis in accordance with Beer's Law. The fluence after 100, 200, 300 and 400 iterations can be seen in figure 4.15. As with the isotropic point source, each of these images displays an expansion of the fluence as the iterations progress.

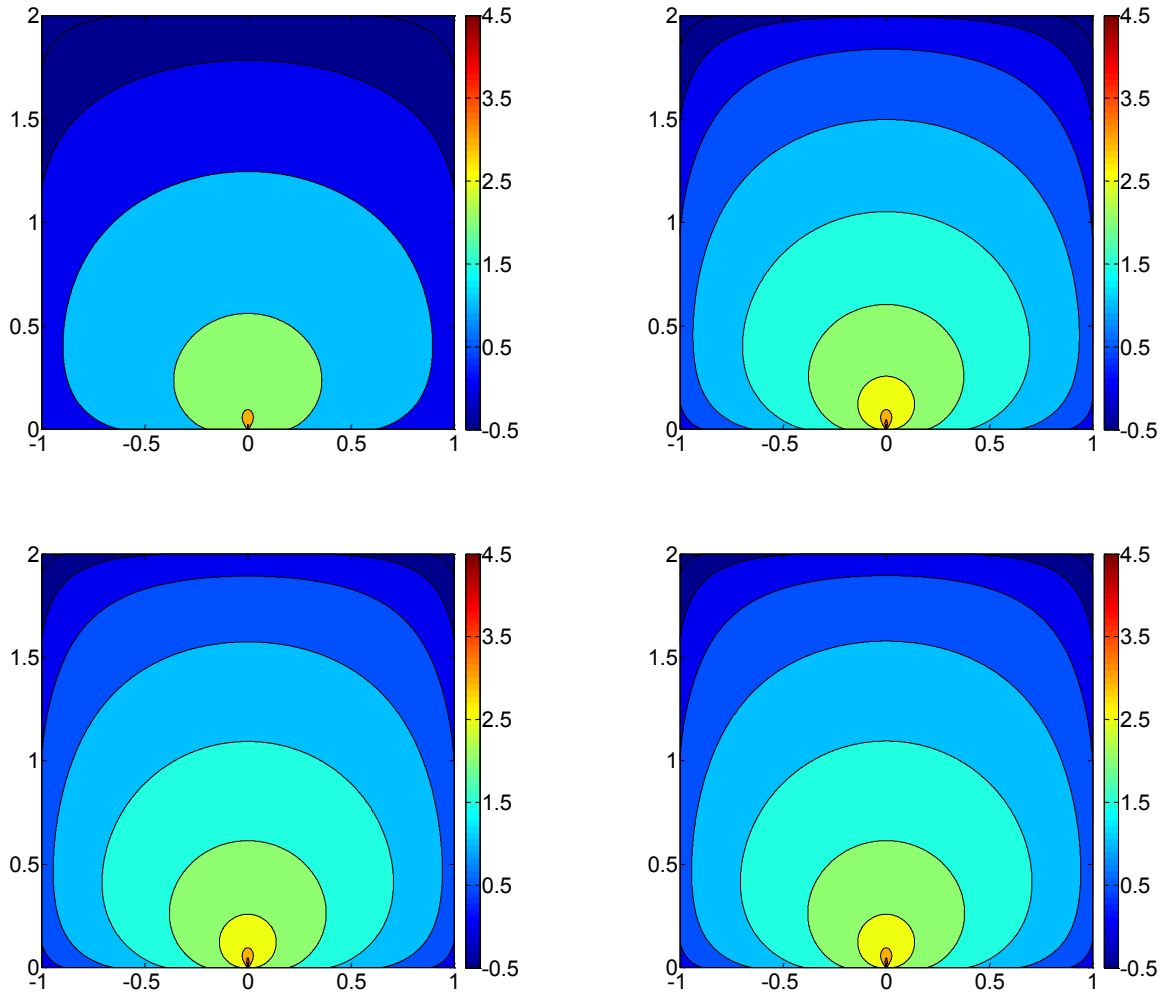


Figure 4.16. The fluence distributions from MM involving a Pencil beam source incident to a semi-infinite turbid media. These are displayed logarithmically (base 10) to ease comparisons. From top left to bottom right we have 500, 1000, 2000 and 3000 iterations. The axes outline the distances of the medium, in cm.

Figure 4.16 displays the images for the expansion of the fluence after 500, 1000, 2000 and 3000 iterations. After 1000 iterations there are still significant changes in the overall fluence, which can be attributed to the increased distance which the radiance must propagate to reach the upper edge. Although the radiance near the lower boundary is stable after 2000 iterations, it is likely that there may be small errors close to the upper boundary due to a lack of iterations. However, when the total number of iterations was increased to 10,000, no significant difference could be seen in these values.



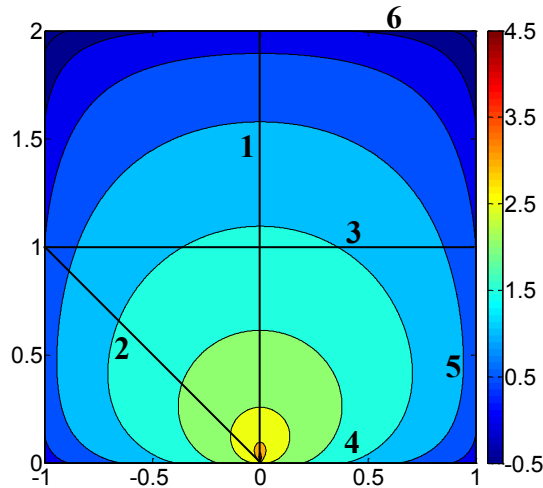


Figure 4.17. Numbered cross sections used in analysis. The axes show the distance, in cm.

Similar to the situation with the isotropic point source, MM can only simulate the pencil beam within a finite medium. Examining the results, one can see that the values closer to the source term reach their final values faster than those further away. It, however, takes considerably longer than in the isotropic point source. The extra distance to the upper boundary, as well as virtually no distance to the lower boundary, accounts for this extra time. It would take longer for the larger radiance values to reach the upper pixels, and more iterations are needed for the pixels closer to the lower boundary since a large fraction of the radiance is set to zero as it leaves the lower boundary. Both cross sectional analysis and error analysis will be used to further examine the accuracy.

### 4.3.3 Cross Section Results

Here, six main cross sections will be assessed (Fig. 4.17). Firstly the central axis which crosses the source term. Secondly the diagonal crossing the source. Thirdly is the center horizontal axis. Fourthly, fifthly and sixthly are the lower, side and upper borders, respectively.

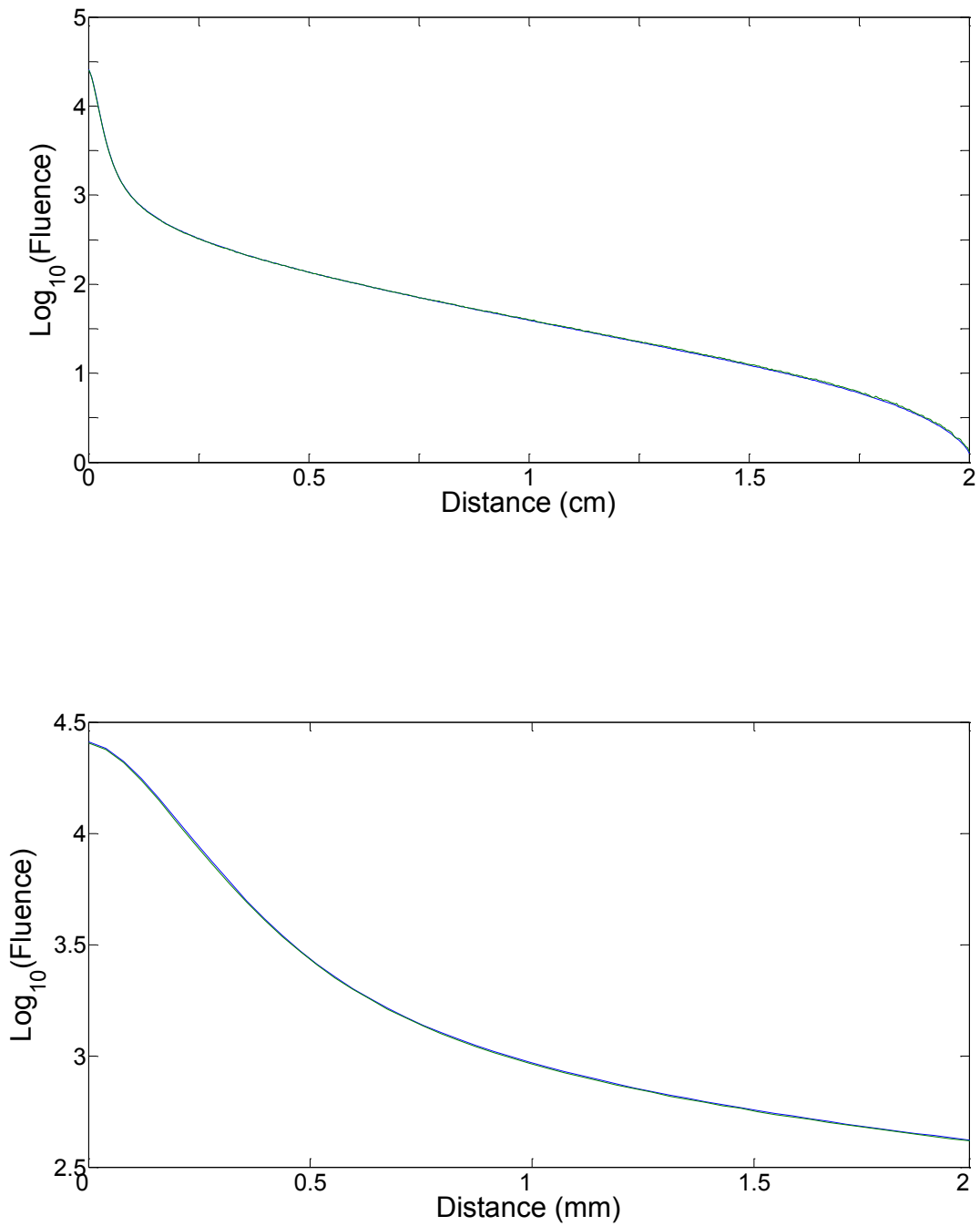


Figure 4.18. The cross section along the center of the medium, through the source. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region up to 2 mm from the source.

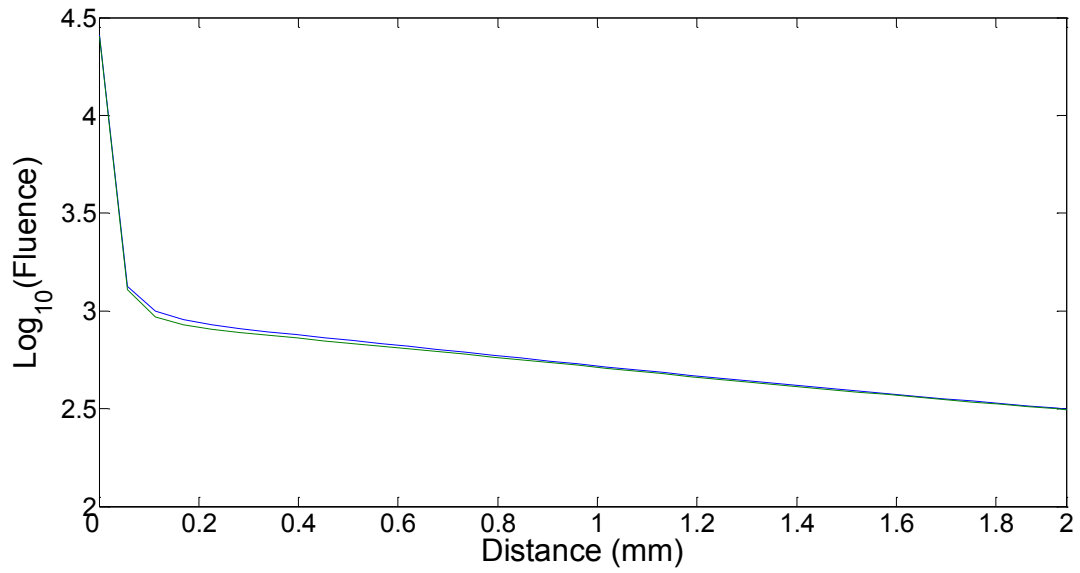
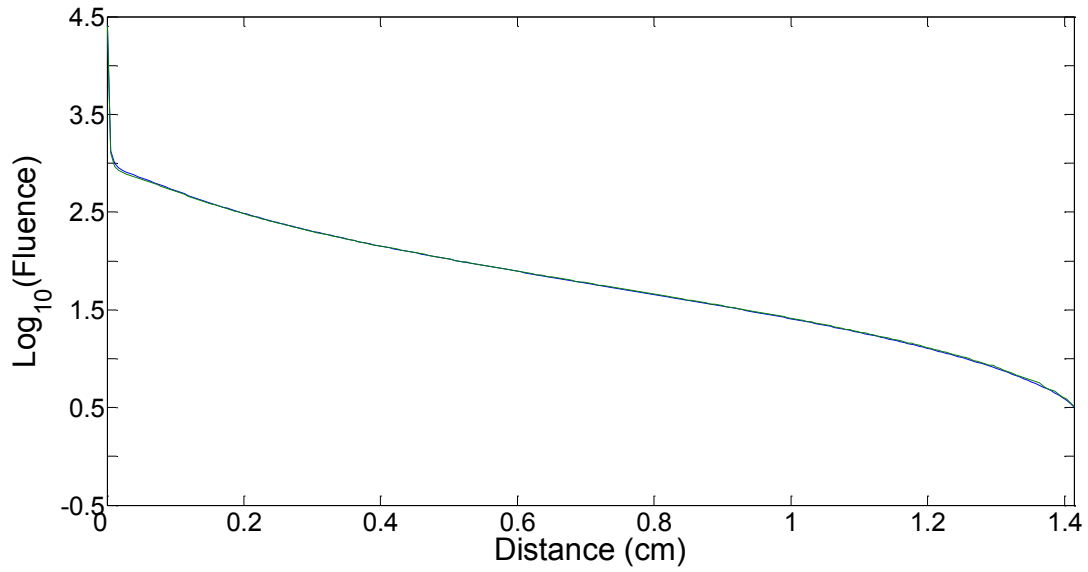


Figure 4.19. The cross section along the 45° diagonal through the source to the edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region up to 2 mm from the source.

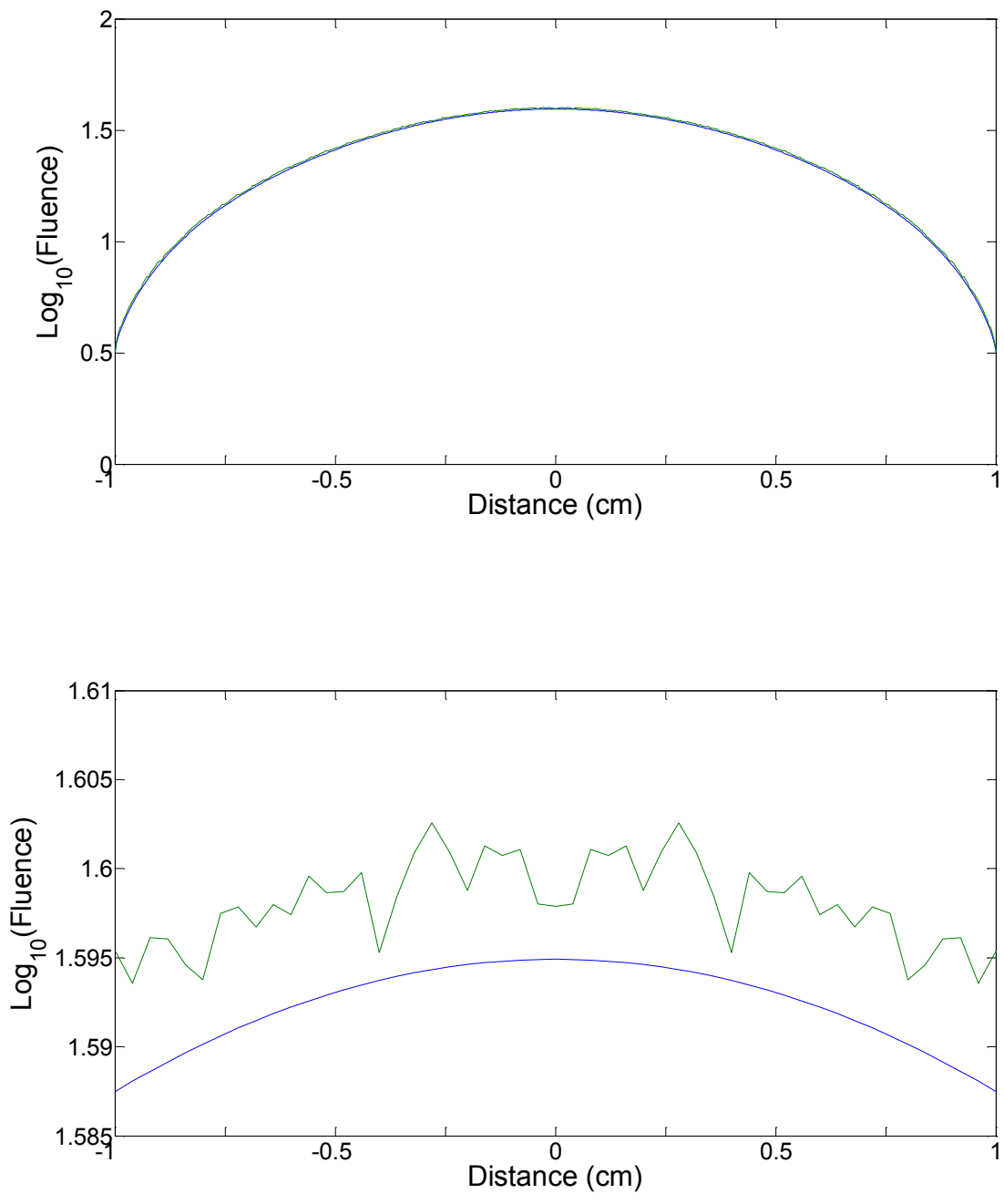


Figure 4.20. Horizontal cross section midway between the source and the upper edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region within 1 mm of the center.

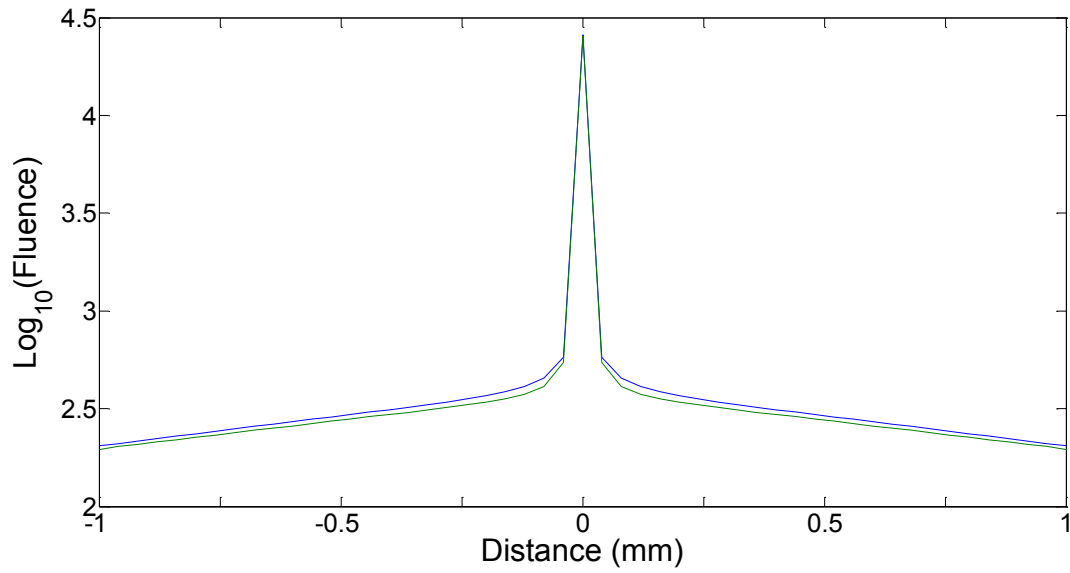
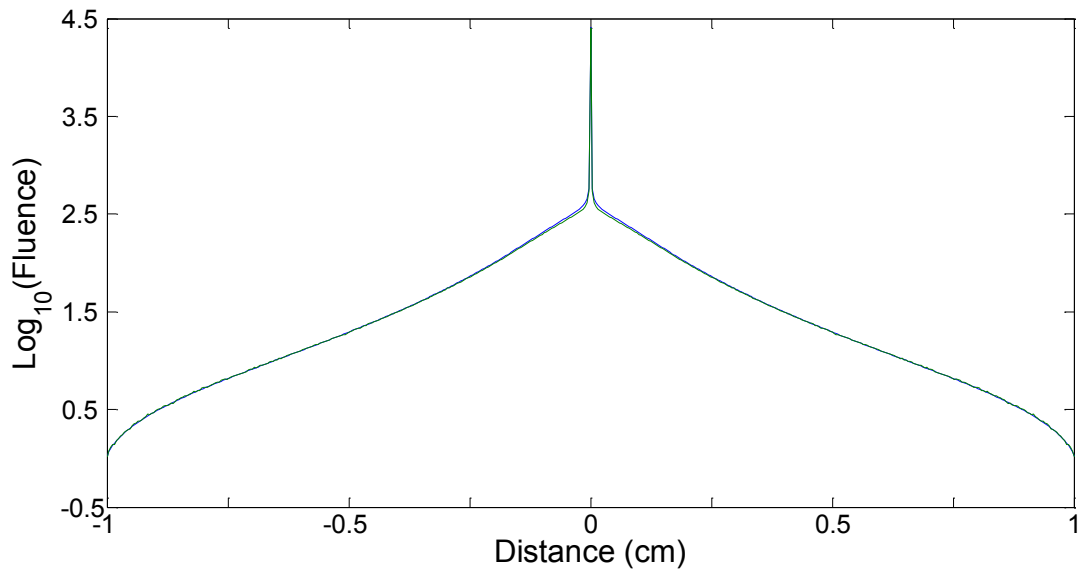


Figure 4.21. The bottom edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region within 1 mm of the center.

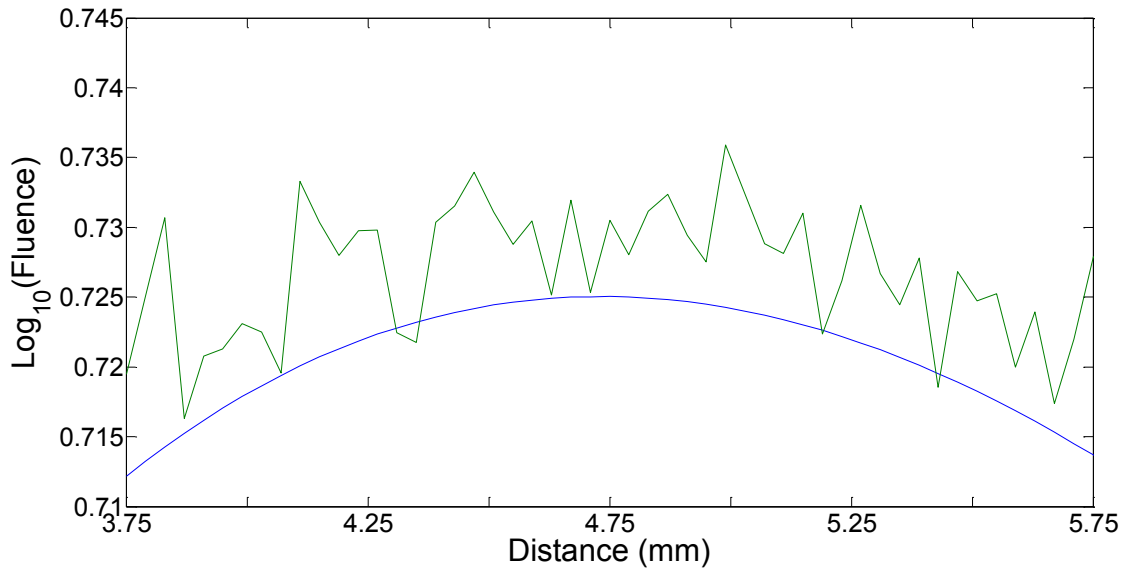
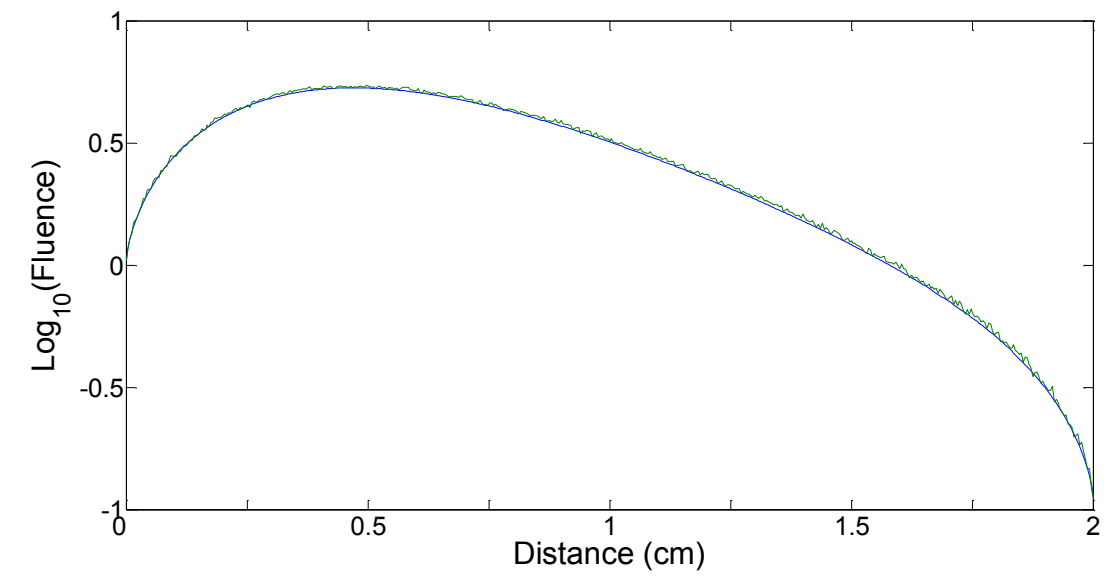


Figure 4.22. The side edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region within 1 mm of the largest value along this edge.

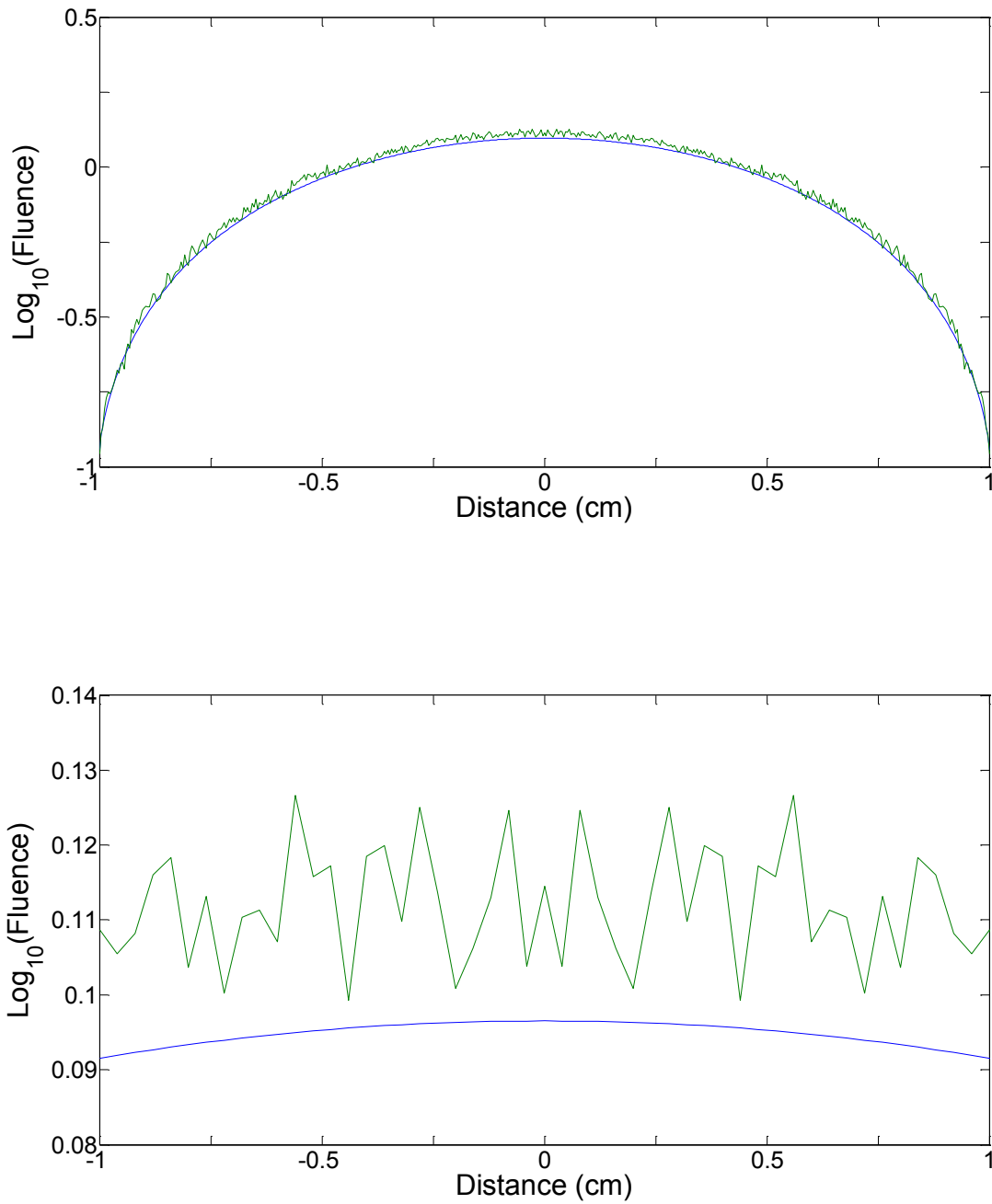


Figure 4.23. The upper edge. Blue is MM and green is MC. The upper plot shows the entire range, while the lower plot shows the region within 1 mm of the largest value along this edge.

The cross section along the center of the medium is shown in figure 4.18. Here it can be easily seen that the two methods reach values which are highly comparable to one another. The incident location has the highest value, at which point it undergoes a sharp decline. This is followed by a slow decrease, until the boundary effects near the upper edge take effect. As with the case of the isotropic point source, the boundary effects are identical in both methods. The same conclusion with regards to a need for extra boundary points can be applied here as well.

These results can also be verified with the diagonal cross section (Fig. 4.19). Here the same overall shapes from figure 4.18 can be easily seen. The most significant difference between these two plots involves the points near the source term, just after the sharp decline. Here MM has a slightly larger value than MC. In contrast with the isotropic point source, the diagonal values are slightly less accurate than what was seen in figure 4.18. The boundary effects are visible as well, with both MC and MM experiencing the same effect. Figure 4.21 demonstrates the deviation after the initial sharp peak.

The horizontal cross section midway between the source and the upper edge (Fig. 4.20) shows fluence values for MC which are slightly larger than MM, although there is still a significant randomness with the MC fluence. The upper edge (Fig. 4.23) outlines a similar situation, with a much larger standard error. It is likely that even though 100,000,000 photon packets were simulated, remaining errors from MC would still increase the errors measured by MSE and MAPE. In figure 4.22, the standard errors seem to increase as the distance from the source increases. This is expected as more photon packets absorption events would occur closer to the source than further from it. As with the isotropic point source, MC still shows standard error issues at these distances, while MM has no randomness effects whatsoever.



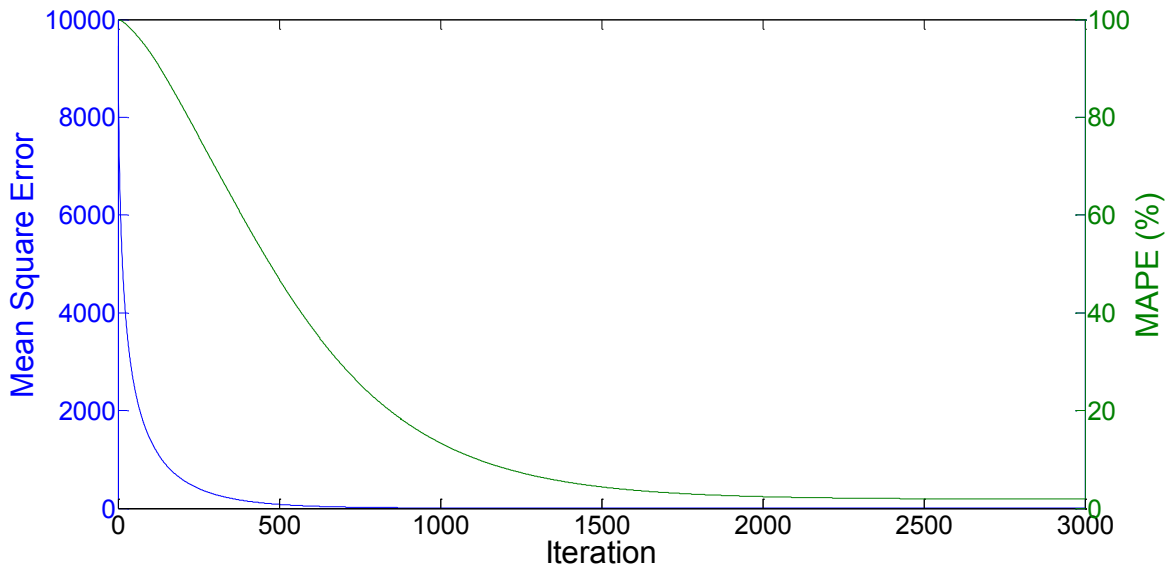


Figure 4.24. MSE (blue) and MAPE (green) of the MM pencil beam simulation.

#### 4.3.4 MAPE & MSE Results

As with the isotropic point source, both MSE and MAPE as will be used to determine the overall error (Fig. 4.24). Both MSE and MAPE decrease as the number of iterations grow. This decrease is steady for MSE, where once 3000 iterations have passed it reaches its minimum value of 4.897. MAPE reaches a minimum value of 1.873 % after 3000 iterations. The MSE decreases by 1931 fold over the course of the simulation. This shows that, similar to the isotropic point source, the results MM produces near the source are likely highly comparable to those from MC.

MAPE decreases by 53 fold during simulation. This is considerably smaller than what was seen from MSE. As MAPE reflects the results further from the source, this decrease implies the results further from the source are not as accurate. Also, there is no increase in MSE or MAPE. The standard error in the results of MC may be larger than expected.

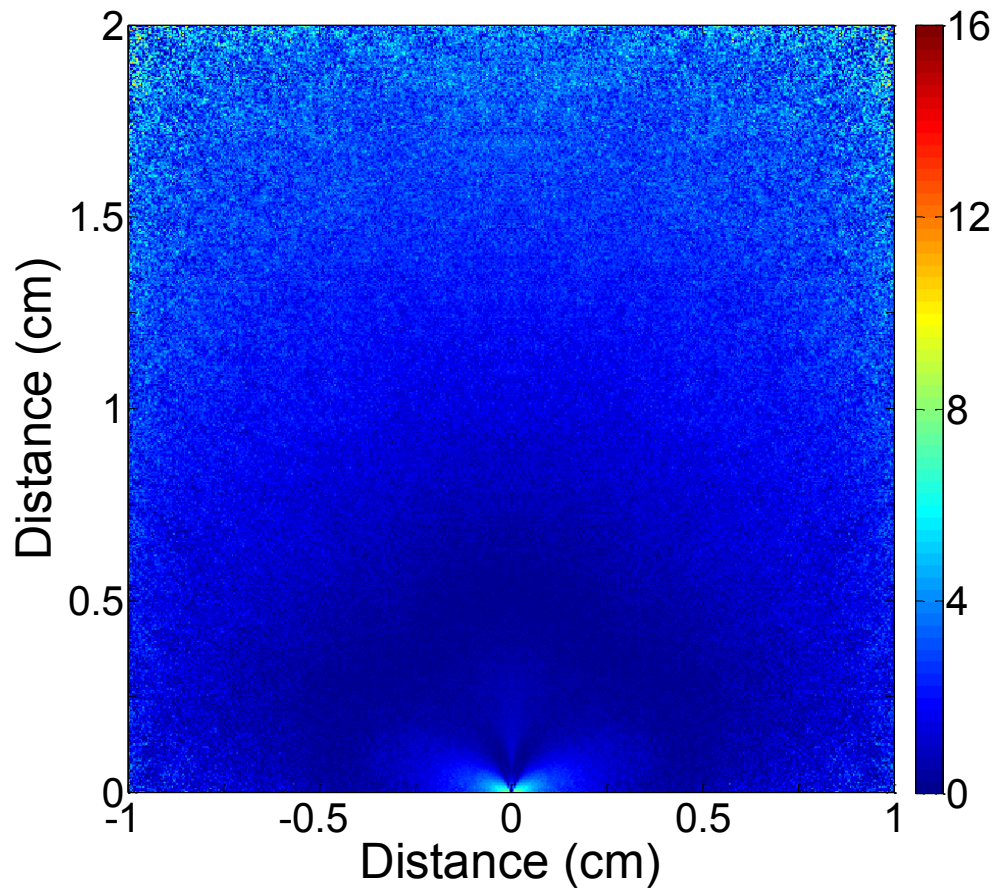


Figure 4.25. The absolute percent error in the pencil beam simulation in a semi-infinite turbid media between MM and MC.

### 4.3.5 MAPE Plots

The absolute percent error is plotted in figure 4.25. Here, the white pixels have an error of less than 1%. Unlike with the isotropic point source, a significant portion of the pixels contribute to MAPE. There are two main regions, being near the source and further away from the source. Due to the distance, less photons will undergo a scattering and absorption event 1 cm above the source. Errors here are likely due to a lack of samples, increasing the standard error. To show they are due to MC, errors of less than 2% will be removed (Fig. 4.26), and a scale from 2% to 4 %, and from 4 % to 6 % will be included. Errors larger than 6% will appear white.

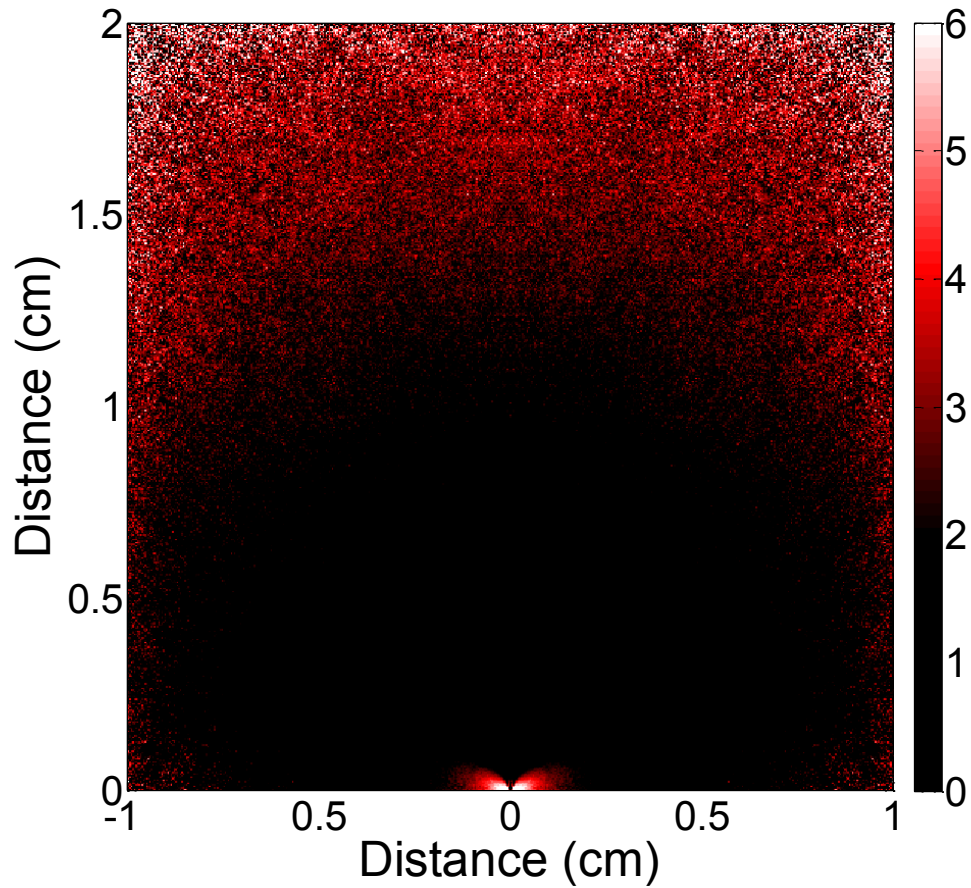


Figure 4.26. The absolute percent error in the isotropic point source between MM and MC. Errors larger than 2% are shown. A scale beginning at 2 % (black) to 4 % (red) and to 6 % (white) is included for better contrast.

As is seen, some of the errors close to the edges and greater than 1 cm away from the source term are still present. It is likely that these errors are still from MC, not MM. In fact, figures 4.20, 4.22 and 4.23 show that the significant differences between these values are mainly from random errors due to MC. As these are standard error related issues, they will not be further discussed. The errors which are assumed to be affiliated with MM are the bow tie like shape near the source (Fig. 4.25). This bow tie error is similar in nature to the errors seen from the isotropic point source (i.e. the clover and the cross). As these errors are present with both simulations, these errors must be due to MM.

## 4.4 Possible Matrix Method Errors

MSE and MAPE show that the primary errors between MM and MC were associated with values that are close to the source, far from the source, and near the boundaries. As stated previously, 100 million photon packets from MC did not generate enough samples to accurately predict the fluence near the boundaries or far from the source. The errors near the source, though, do have enough samples to accurately predict the fluence. The errors here are primarily from MM.

Any errors from MM must be due to the approximations associated with MM. From MM's derivation, there are three main approximations. These are the use of small masks to project the radiance, the use of the interpolated scattering values for the scattering step and the approximation that all scattering occurs at the center of the pixel. One other source of errors could be due to an increase in the total number of iterations which were done with MM over MC. First the errors themselves will be examined by cross sectional analysis. This will help understand how the errors are generated. Each of the possible error sources will then be examined, and the likelihood of errors associated with the source term will be discussed.

### 4.4.1 Bottom Boundary Cross Section per Iteration

From the MAPE plots (Fig. 4.25 and 4.26), the main source of errors near the source are directly along the lower boundary. To determine when these errors occur, the percent error of the cross sections along the bottom boundary after each iteration in MM will be calculated. The results of these calculations for the pencil beam simulation are in figure 4.27. As these errors stem from the same source, only the percent error, not MSE, is shown.

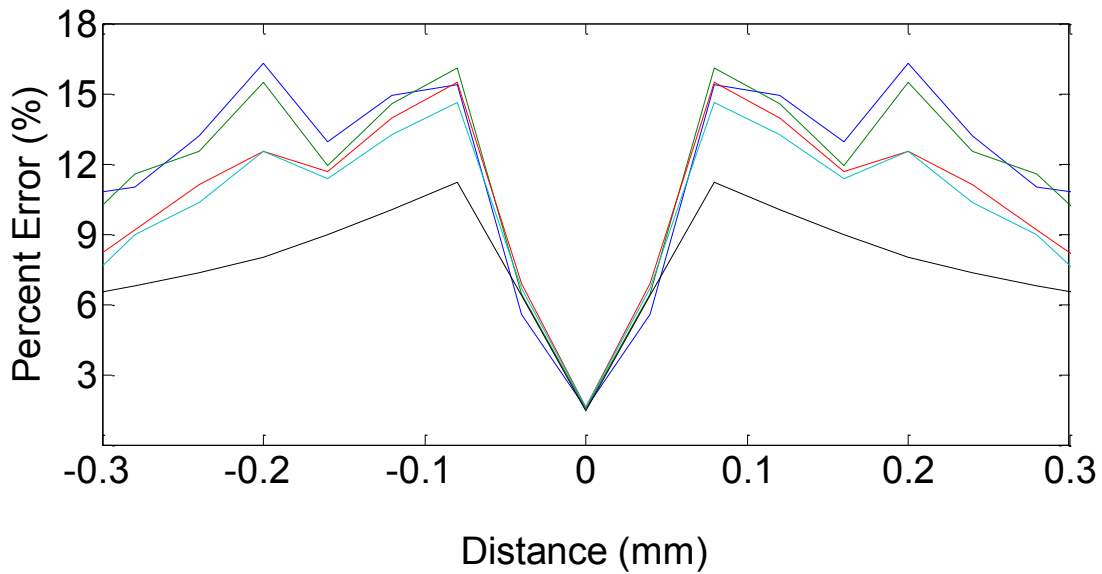
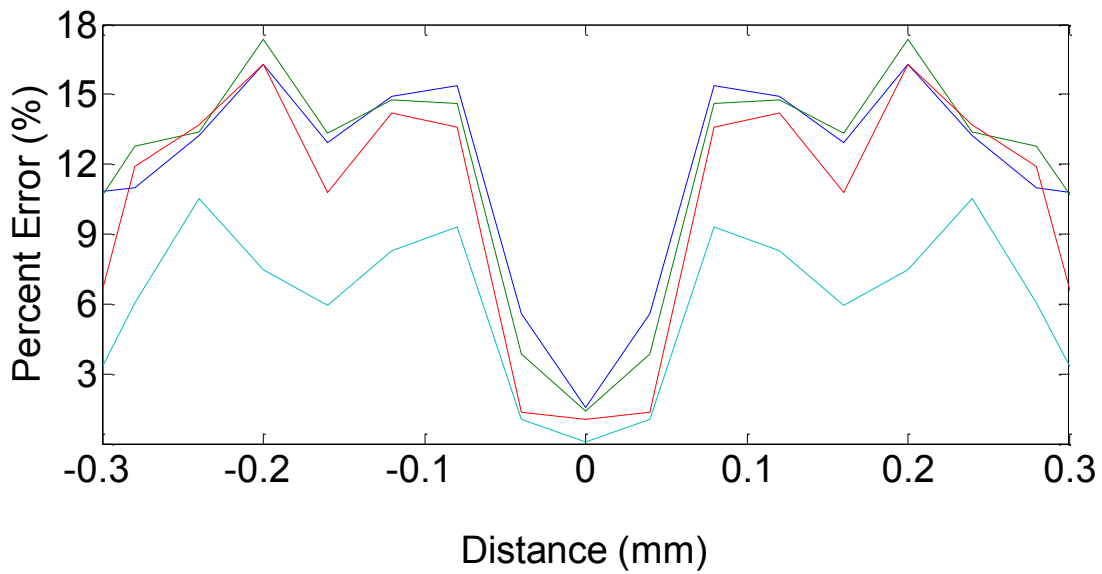


Figure 4.27. Pencil Beam. Horizontal cross section of the lower boundary for several iterations. The upper plot shows the percent error after the (cyan) first, (red) second, (green) third and (blue) fourth iteration. The lower plot shows the percent error after the (blue) fourth, (green) fifth, (red) tenth, (cyan) fifteenth and (black) 3000th iteration.

The percent error calculations show that there are significant errors produced in the first few iterations of MM. These errors also seem to stabilize over time, as evidenced by the final percent error calculation after 3000 iterations. An error in the approximations which would cause significant deviations during the first several iterations is most suspect. Errors which need many iterations to occur are likely irrelevant to the source errors.

#### 4.4.2 Small Masks

The size of the convolution masks would primarily affect values which are further away from the source. Increasing it would first lead to longer simulation times, with more accurate results farther away from the source. As the errors with MM are primarily located within the vicinity of the source, it is highly unlikely that this approximation is of any concern.

#### 4.4.3 Number of Solid Angle Segments

From the error analysis of the solid angle segments, there is less than 0.15% error between the values from the modified Henyey-Greenstein and the interpolated form. Any errors here would take many iterations before becoming significant.

#### 4.4.4 Number of Iterations

This is the least likely issue as the MM values deviate significantly from the MC values after only a few iterations. This error is expected to have little to no effect on pixels which are near the source terms.

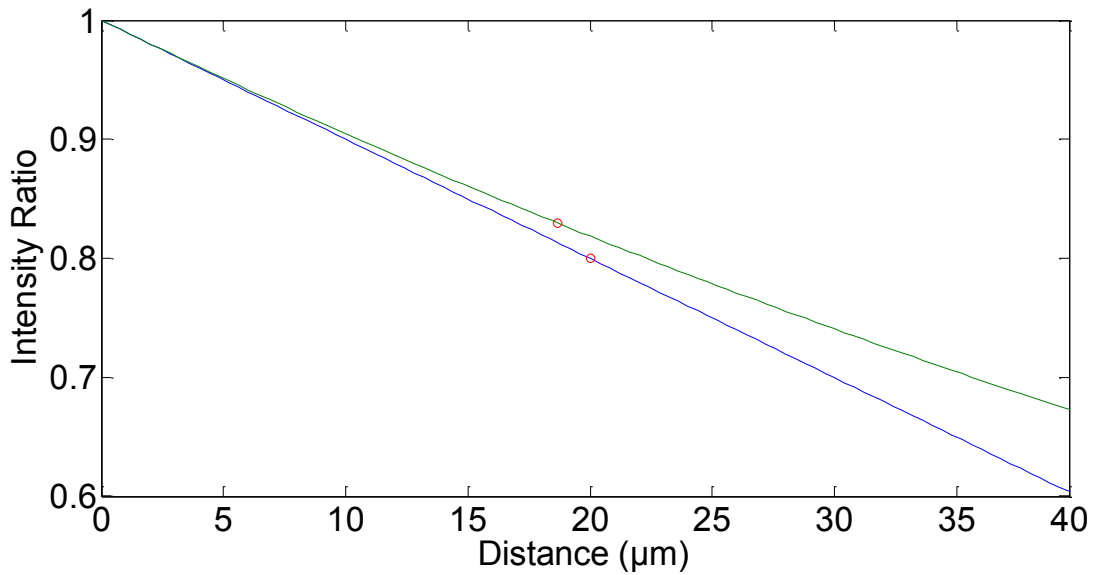


Figure 4.28. The location of mean absorbance and scattering within a pixel from (blue) a linear model and (green) Beer's Law. The location of mean absorbance and scattering using weighted average was found to be 20  $\mu\text{m}$  with linear and 18.7  $\mu\text{m}$  with Beer's Law within a pixel 20  $\mu\text{m}$  long.

#### 4.4.5 Scattering Location

The approximation which allows convolution to be used involves assuming the photons scatter at the center of the pixel. Errors which may occur with this approximation were assumed to either average out with incoming radiance from multiple directions or be negligible due to small pixel size. Either of these errors would be the most likely cause of errors near the source.

Seen in figure 4.28, the mean location of absorbance is at the center of the pixel using a linear absorption model. With Beer's Law, this mean location shifts away from the center. This shift would only occur near the source locations as the source generates a significant bias in the direction of the incoming radiance. These locations were found as follows. For both models, the 40  $\mu\text{m}$  path length was divided into 1000 sections, and each section saves the fraction of photons which would be scattered within it.

In the linear absorption model, the equation which is used to determine absorption is given by

$$I = I_o (1 - \mu_t d) \quad (W/m^2) \quad (4-3)$$

With  $d$  as the distance and  $\mu_t$  is the extinction coefficient ( $100.1 \text{ cm}^{-1}$ ). In the Beer's Law absorption model, the equation which is used is given by

$$I = I_o e^{(-\mu_t d)} \quad (W/m^2) \quad (4-4)$$

Using these two equations, the amount absorbed within each section can be found. With these values, a weighted average can be used to determine the location of mean absorbance.

The weighted average is given by, where  $x_{section}$  is the distance to the center of that section and  $A(section)$  is the fraction of the intensity absorbed at that section.

$$\bar{x} = \frac{\sum_{section=1}^{1000} A(section) * x_{section}}{\sum_{section=1}^{1000} A(section)} \quad (\mu m) \quad (4-5)$$

#### 4.4.6 Multiple Direction Averaging

In order for the assumption of all scattering occurring at the center of the pixel to be valid, no significant biases in the magnitude of the radiance projected to the pixel from any particular direction can be present. This does not need to hold true for any single iteration, but must hold true over the entire simulation. Any significant biases in the direction of the incoming radiance would lead to deviations in the location of mean scattering.



## 4.4.7 Pixel Size

The size of the pixels can affect several parameters. These include the fraction of radiance  $W/(m^2 \cdot sr)$  projected from other pixels which will scatter within the pixel as well as the portion of radiance which scatters within the same pixel multiple times. MM first calculates the portion of radiance which scatters within the same pixel which it was projected from. This portion is added to the radiance which is projected from other nearby pixels that will scatter within this pixel.

Although the fraction which scatters within the same pixel may be small for pixels far from the source, it represents a substantial portion of the radiance for pixels near to the source. The portion of radiance can be represented by  $p_{ii}^k$ , and equation 3-11 can be used to evaluate it. Using the values within this thesis, the fraction which scatters within its initial pixel evaluates approximately  $\sim 20\%$ .

These radiance values can cause errors as the location of mean scattering within the original pixel. Where it is assumed that all of the radiance can be projected from the center of the pixel, the radiance which has already scattered within this pixel actually projects a significant distance from the center, leading to errors.

Figure 4.29 allows for a glimpse into the errors which arise due to this bias. The mean location where the radiance which has scattered within this same pixel would project from is outlined by the red line in pixel 3. Propagating this radiance as though it came from the center would lead to errors for two reasons. First, it would decrease the radiance emitted from the pixel, and secondly increase the fraction which scatters and absorbs for a second time within the pixel.

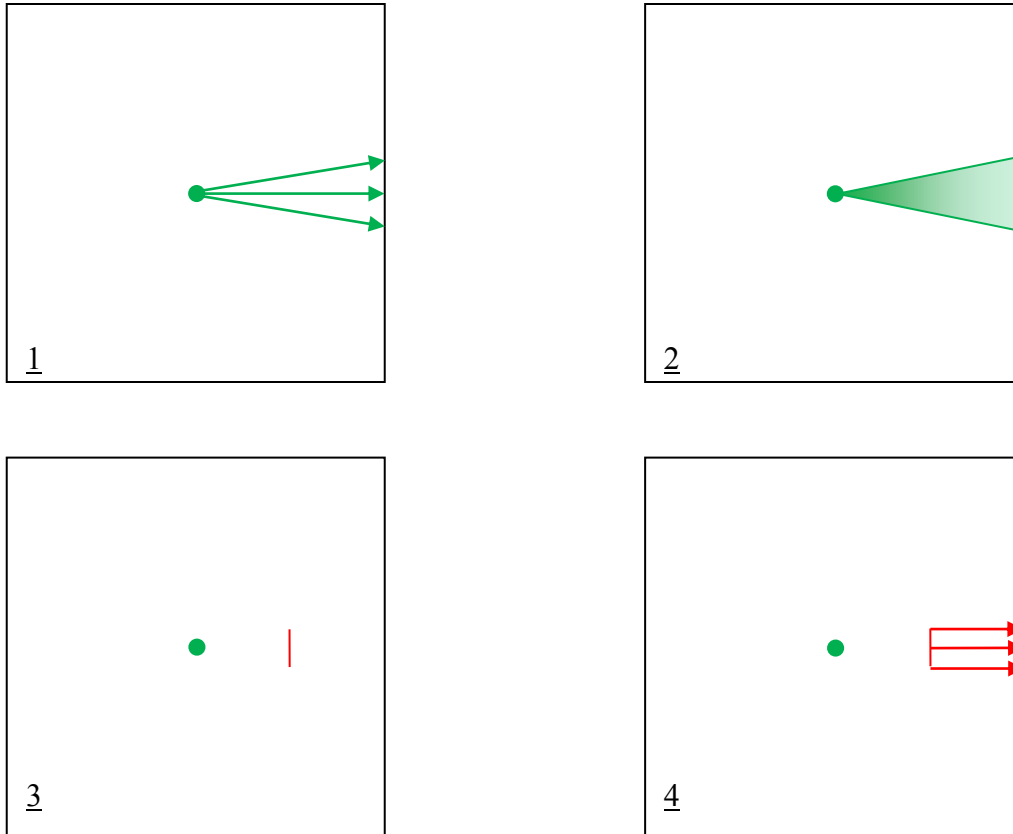


Figure 4.29. Above is 1) initial projection of radiance, 2) propagation of radiance from center pixel, 3) location of mean scattering from this propagation (red line) and 4) projection of radiance from red line. In MM this radiance is propagated from the center as opposed to the location of mean scattering (red line).

To mitigate these effects, one could develop a secondary set of convolution masks which can accurately correct for this bias in the radiance. Another way would be to simply scale this radiance so it can be project as though it were from the center of the pixel. After this scaling, subtracting out the excessive absorbance must occur.

### 4.4.9 Diagonal Pixels

Sections 4.4.5 to 4.4.8 outline the main sources of errors in pixels close to the source terms. If they were only dependent on their distance from the source, these errors would form circles around the source. Instead, a clover and a cross are seen from the isotropic point source,

and a bow tie from the pencil beam. The diagonals in these simulations have smaller errors than the pixels along the horizontal and vertical axes. Since the two factors mentioned above would still have an effect on the diagonals, why are the diagonals more accurate?

This is explained by taking the location of mean scattering and absorption errors and errors from radiance scattering within the same pixels which it had projected from into consideration. The incoming radiance does not cross a single side, but travels through a corner.

Scattering within this pixel would occur closer to the pixel's center, significantly reducing the impact of the source term. This is due to the fact that the location of mean scattering and absorption is closer to the center. As such, the difference in the fraction absorbed and scattered from projecting the radiance as though all of it came from the center of the pixel can be reduced. The errors here are further reduced as there would be an equivalent reduction in the bias from the source pixel.

## 4.5 Clover, Bow Tie and Cross Discussion

As the different potential sources of errors have been discussed, the focus will now turn to what causes the errors in the clover, bow tie and cross. The clover and bow tie likely have the same source of errors. They appear differently due to the initial ballistic projection of the pencil beam.

The lower error values from the clover are due to the propagation of erroneous radiance values from the cross. The scattering term spreads this error from the cross across to the other solid angle segments, which eventually leads to a series of smaller errors over a much broader range.

	Isotropic Point Source	Pencil Beam
Matrix Method (3000 Iterations)	37 minutes and 53 seconds	37 minutes and 46 seconds
Monte Carlo ( $10^8$ photon packets)	19 hours, 30 minutes and 23 seconds	27 hours, 39 minutes and 10 seconds

Table 4.1. Sample times taken for the simulations to run. Here, the programs were both written so they could benefit from MATLAB parallel processing toolbox, which can use multiple processors to run simulations.

This spread is also the main reason behind the size of the bow tie. To solve these errors, the cross must be addressed first. The errors from the cross come from two sources. These are the propagation of radiance which scattered within the same pixel it projected from, and the pixel size.

To test to see if correcting for these factors leads to a reduction in the error, a source correction factor will be included in the simulation. This source correction factor will entail using a scaling factor to scale the radiance scattered within the same pixel it originated from so it can be projected as though it were from the center of the pixel. Then, the excess absorbed radiance will be simply be subtracted out.

### 4.5.1 Point Source: Source Correction Factor

The isotropic point source simulation is re-run, this time including the above factor. The final MSE changed from 27.6 to 8.277 while the final MAPE changed from 0.795 % to 1.226 %. The large decrease in MSE is from the increased accuracy in the source terms, while the increase in MAPE can only be addressed using convolution masks specifically designed to propagate radiance which previously scattered within the same pixel it projected from.

## 4.6 Time Analysis

The reason why the increase in memory MM requires is acceptable is MM conducts the simulation faster than MC. To show this holds true for simulations with a large number of pixels, the time required for several simulations will be shown in table 4-1.

### 4.6.1 Simulation Time

The simulations for MM take approximately 1 minute and 15 seconds to run 100 iterations. This is substantially larger than the theoretically calculated time of 9.4 seconds. The difference is mainly due to programing elements, accessing memory values and rearranging the radiance between the projection and scattering phases of the simulation. As well, other programs are running with MATLAB, so the processors were not solely dedicated to the simulations.

Table 4-1 also shows the significant differences between the times required to calculate the four simulations. MM takes approximately the same amount of time for each simulation since it is dependent on the grid used. Here the grids are the same, meaning the calculations should, and do, take the same amount of time. MC takes two different times to run these simulations due to two factors.

The first one involves the pencil beam simulation using 50,000,000 photon packets, while the isotropic point source uses 12,500,000 photon packets to operate. Averaging brings the effective total to 100,000,000 photon packets for both simulations. The second factor is the distance of the source to the edge of the medium. As the source in the pencil beam simulation is on the boundary, a significant fraction of the photon packets will exit the medium with a high fraction of weight remaining. This terminates them early.

What is astonishing is the difference in times for both methods. It is almost 31 times faster to calculate the isotropic point source with MM than MC. With the pencil beam, it is around 44 times faster to evaluate. Not only that, but there are no standard error issues with the results of MM as compared to MC. This implies that to match the accuracy of MM, even more photon packets would be needed for MC.

# Chapter 5

## Conclusions

This thesis illustrates a novel modeling approach to determining the fluence from a source within or striking a scattering media. The theoretical background behind the Matrix Method was developed, and several tests involving the pencil beam in a semi-infinite turbid media and the isotropic point source in an infinite turbid media were conducted to evaluate the novel methods efficiency, benefits and drawbacks. These factors were compared to the gold standard method, Monte Carlo. Here, it was shown that the original method generated many results within 4% of the values of MC. This held true as long as the values were far from the source. Near the source, the errors increased due to pixel sizes and second order radiance effects. Using correction factors reduced some of these errors.

## 5.1 Chapter Summary

Chapter 1 explores the basic concepts behind scattering in tissue. Here, scattering occurs due to interactions with organelles and other objects within the tissue [28]. These interactions are dependent on the ratio between the wavelength of the electromagnetic radiation, the cross section of the objects with which it interacts, and the index of refraction of the medium and the object [10]. Tissue, however, has an effective window in the NIR range which allows photons to penetrate through [16-18]. How it penetrates through, and how to model the fluence, is a field of research.

Methods of modeling fluence have been proposed in the past. Chapter 2 defines the factors associated with light modeling, including fluence and radiance. Other methods which have been shown to model fluence are discussed. Notably, Monte Carlo is described and used as the gold standard to compare the topic of this thesis, the Matrix Method.

The Matrix Method is a technique which calculates the radiance propagation to determine the fluence. It is conceived from discrete ordinates, a method proposed by Chandrasekhar [92]. Chapter 3 goes into great detail over how the radiance can be divided into solid angle segments, and how these segments can be propagated through a medium with the use of two matrices. These matrices can be redeveloped as convolution operations, which greatly reduces the computational requirements of the method into smaller more useful operations.

The theory may be useful, however without confirmation it is simply conjecture. Chapter 4 examines two common simulations to compare the results from the Matrix Method and Monte Carlo. Several details were noticed. No random errors were present in the Matrix Method, and its results were close to the values from Monte Carlo far from the source, but not near the source. These errors stemmed from two sources, being the pixel size and second order radiance.



A second simulation was also conducted, correcting for some of the errors. The MSE changed from 27.6 to 8.28 while MAPE changed from 0.7946 % to 1.226 %. As these corrections were composed of scaling factors, it is likely that more accurate convolution masks could be used to further reduce any errors.

## 5.2 Future Directions

This work was exploratory in nature. Here, the final convolutions matrices necessary for the Matrix Method were not expected to be found. Whether or not this method could potentially be used as a means of determining the overall fluence was expected to be determined. And here, there is promising evidence that shows this method has great potential. To continue to study the Matrix Method some suggestions would be:

- Create convolution masks for second order radiance.
- Conduct the propagation and scattering steps in the frequency domain.
- Design the program such that the radiance values do not need to be reorganized.
- Use much smaller pixels to mitigate errors.
- Explore creating a hybrid between MM and MC.

For future simulations, there are several different simulations which should be attempted:

- Determining the fluence from a fluorescent molecule.
- Attempting index-matched boundary conditions.
- Attempting index-mismatched boundaries.
- Confirming the results in a tissue sample.
- Conduct simulations in a 3D medium.

# References

- [1] B. C. Wilson and M. S. Patterson, “The physics of photodynamic therapy”, *Physics in Medicine and Biology*, vol. 31, no. 4, pp. 327 – 360, 1986.
  
- [2] O. Barajas, et. al., “Monte Carlo modelling of angular radiance in tissue phantoms and human prostate: PDT light dosimetry”, *Physics in Medicine and Biology*, vol. 42, no. 9, pp. 1675 – 1687, 1997.
  
- [3] C. K. Hayakawa, et. al., “Perturbation Monte Carlo methods to solve inverse photon migration problems in heterogeneous tissues”, *Optics letters*, vol. 26, no. 17, pp. 1335 – 1337, 2001.
  
- [4] S. R. Arridge and J. C. Hebden, “Optical imaging in medicine: II. Modelling and reconstruction”, *Physics in Medicine and Biology*, vol. 42, no. 5, pp. 841 – 853, 1997.

- [5] A. A. Oraevsky, et. al., "Measurement of tissue optical properties by time-resolved detection of laser-induced transient stress", *Applied Optics*, vol. 36, no. 1, pp. 402 – 415, 1997.
- [6] L. V. Wang and H. Wu, "Sensing of Optical Properties and Spectroscopy", in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 7, sec. 7.1, pp. 135.
- [7] D. A. Boas, et. al., "Scattering of diffuse photon density waves by spherical inhomogeneities within turbid media: analytic solution and applications", *Pro. National Academy of Sciences*, vol. 91, no. 11, pp 4887 – 4891, 1994.
- [8] A. J. Welch, et. al., "Definitions and Overview of Tissue Optics" in *Optical-Thermal Response of Laser-Irradiated Tissue*, 2<sup>nd</sup> ed. New York: Springer, 2011, chp. 3, sec. 3.1, pp. 28.
- [9] W. C. Y. Lo and L. Lilge, "Accelerated 3D Monte Carlo light dosimetry using a graphics processing unit (GPU) cluster", *Laser Applications in Life Sciences*, International Society for Optics and Photonics, pp. 737609 – 737618, 2010.
- [10] L. T. Perelman, et. al., "Observation of periodic fine structure in reflectance from biological tissue: a new technique for measuring nuclear size distribution", *Physical Review Letters*, vol. 80, no. 3, pp 627 – 630, 1998.

- [11] R. Drezek, et. al., “Light scattering from cells: finite-difference time-domain simulations and goniometric measurements”, *Applied Optics*, vol. 38, no. 16, pp. 3651 – 3661, 1999.
- [12] A. E. Efimov, et. al., “Atomic force microscope (AFM) combined with the ultramicrotome: a novel device for the serial section tomography and AFM/TEM complementary structural analysis of biological and polymer samples”, *J. Microscopy*, vol. 226, no. 3, pp. 207 – 216, 2007.
- [13] B. J. Tromberg, et. al., “Non-invasive in vivo characterization of breast tumors using photon migration spectroscopy”, *Neoplasia*, vol. 2, no. 1-2, pp. 26 – 40, 2000.
- [14] A. J. Welch and M. J. C. van Gemert, “Overview of Optical and Thermal Laser-Tissue Interaction and Nomenclature” in *Optical-Thermal Response of Laser-Irradiated Tissue*, 2<sup>nd</sup> ed. New York: Springer, 2011, chp. 1, sec. 1.1, pp. 4.
- [15] G. Rojo. (2011, July 13). *Structure of Living Things* [Online]. Available: <http://www.rojo5th.net/2011/07/13/>
- [16] L. V. Wang and H. Wu, “Rayleigh Theory and Mie Theory for a Single Scatterer”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 2, sec. 2.5, pp. 21 – 23.
- [17] L. V. Wang and H. Wu, “Rayleigh Theory and Mie Theory for a Single Scatterer”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 2, sec. 2.2, pp. 17 – 18.

- [18] J. B. Reece, et. al., “A Tour of the Cell”, in *Campbell Biology, Custom for MacEwan University*, 2<sup>nd</sup> ed. San Francisco, California: Benjamin Cummings, 2011, chp. 6, sec. 6.3, pp. 102.
- [19] J. B. Reece, et. al., “A Tour of the Cell”, in *Campbell Biology, Custom for MacEwan University*, 2<sup>nd</sup> ed. San Francisco, California: Benjamin Cummings, 2011, chp. 6, sec. 6.5, pp. 110.
- [20] L. V. Wang and H. Wu, “Rayleigh Theory and Mie Theory for a Single Scatterer”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 2, sec. 2.4 – 2.5, pp. 20 – 24.
- [21] R. Weissleder, “A clearer vision for in vivo imaging”, *Nature biotechnology*. vol. 19, no. 4, pp. 316 – 317, 2001.
- [22] E. İ. Altinoğlu and J. H. Adair, "Near infrared imaging with nanoparticles" *Wiley Interdisciplinary Reviews: Nanomedicine and Nanobiotechnology*, vol. 2.5, pp. 461 – 477, 2010.
- [23] A. M. Smith, et. al., “Second window for in vivo imaging”, *Nature nanotechnology*, vol. 4, no 11, pp. 710 – 711, 2009.
- [24] L. V. Wang and H. Wu, “Diffuse Optical Tomography”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 11, sec. 11.8.4, pp. 274 – 275.

- [25] L. V. Wang and H. Wu, “Radiative Transfer Equation and Diffusion Theory”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 5, sec. 5.1 – 2, pp. 83 – 85.
- [26] A. J. Welch, et. al., “Definitions and Overview of Tissue Optics” in *Optical-Thermal Response of Laser-Irradiated Tissue*, 2<sup>nd</sup> ed. New York: Springer, 2011, chp. 3, sec. 3.2.2, pp. 32 – 35.
- [27] S. L. Jacques and S. A. Prahl, (1998). *Radiant Intensity* [Online]. Available: <http://omlc.ogi.edu/education/ece532/class1/intensity.html>
- [28] L. V. Wang and H. Wu, “Introduction”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 1, sec. 1.4 – 5, pp. 5 – 8.
- [29] A. J. Welch, et. al., “Basic Interactions of Light with Tissue” in *Optical-Thermal Response of Laser-Irradiated Tissue*, 2<sup>nd</sup> ed. New York: Springer, 2011, chp. 2, sec. 2.3.3, pp. 22 – 26.
- [30] S. L. Jacques, et. al., “Angular Dependence of HeNe Laser Light Scattering by Human Dermis”, *Lasers in the Life Sciences*, vol. 1, no. 4, pp. 309 – 334, 1987.
- [31] A. J. Welch, et. al., “Definitions and Overview of Tissue Optics” in *Optical-Thermal Response of Laser-Irradiated Tissue*, 2<sup>nd</sup> ed. New York: Springer, 2011, chp. 3, sec. 3.4.3.2, pp. 52.

- [32] L. V. Wang and H. Wu, “Monte Carlo Modeling of Photon Transport in Biological Tissue”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 3, sec. 3.4, pp. 39 – 50.
- [33] W. Cong, et. al., “Modeling photon propagation in biological tissues using a generalized Delta-Eddington phase function,” *Physical Review E*, vol. 76, no. 5, pp. 051913, 2007.
- [34] L. V. Wang and H. Wu, “Radiative Transfer Equation and Diffusion Theory”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 5, sec. 5.4, pp. 88 – 101.
- [35] L. V. Wang and H. Wu, “Radiative Transfer Equation and Diffusion Theory”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 5, sec. 5.3, pp. 85 – 88.
- [36] A. H. Hielscher, R. E. Alcouffe, and R. L. Barbour, “Comparison of finite-difference transport and diffusion calculations for photon migration in homogeneous and heterogeneous tissues”, *Physics in Medicine and Biology*, vol. 43, pp. 1285 – 1302, 1998.
- [37] J. Ripoll, et. al., “Boundary conditions for light propagation in diffusive media with nonscattering regions” *Journal of the Optical Society of America A*, vol. 17, no. 9, pp. 1671 – 1681, 2000.

- [38] E. D. Aydin, “Three-dimensional photon migration through voidlike regions and channels.” *Applied Optics*, vol. 46, no. 34, pp. 8272 – 8277, 2007.
- [39] B. Rutily and L. Chevallier, “Why is it so difficult to solve the radiative transfer equation?”, *EAS Publications Series*, vol. 18, pp. 1 – 23, 2006.
- [40] A. Liemert and A. Kienle, “Exact and efficient solution of the radiative transport equation for the semi-infinite medium,” *Scientific reports*, vol. 3, pp. 1 – 7, 2013.
- [41] E. Vitkin, et. al., “Photon diffusion near the point-of-entry in anisotropically scattering turbid media,” *Nature Communications*, vol. 2, pp. 587, 2011.
- [42] R. J. Zemp, “Phase-function corrected diffusion model for diffuse reflectance of a pencil beam obliquely incident on a semi-infinite turbid medium”, *Journal of Biomedical Optics*, vol. 18, no. 6, pp. 0067005, 2013.
- [43] L. V. Wang and H. Wu, “Hybrid Model of Monte Carlo Method and Diffusion Theory”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 6, sec. 6.1, pp. 119.
- [44] J. B. Schneider. (2010) “Introduction to the Finite-Difference Time-Domain Method: FDTD in 1D,” in *Understanding the FDTD Method* [Online]. Available: <http://www.eecs.wsu.edu/~schneidj/ufdtd/chap3.pdf>.
- [45] M. N. O. Sadiku, “Finite Element Method,” in *Numerical Techniques in Electromagnetics*, 2<sup>nd</sup> ed. Boca Raton, CRC Press LLC, 2001, chp. 6, sec. 6.2.2, pp. 382.



- [46] M. N. O. Sadiku, "Finite Element Method," in *Numerical Techniques in Electromagnetics*, 2<sup>nd</sup> ed. Boca Raton, CRC Press LLC, 2001, chp. 6, sec. 6.2.1, pp. 379.
- [47] M. N. O. Sadiku, "Finite Element Method," in *Numerical Techniques in Electromagnetics*, 2<sup>nd</sup> ed. Boca Raton, CRC Press LLC, 2001, chp. 6, sec. 6.1, pp. 377 – 378.
- [48] S. R. Arridge, et. al., "A finite element approach for modeling photon transport in tissue," *Medical Physics*, vol. 20, no. 2, pp. 299 – 309, 1993.
- [49] M. Schweiger and S. R. Arridge, "The finite-element method for the propagation of light in scattering media: frequency domain case," *Medical Physics*, vol. 24, no. 6, pp. 895 – 902, 1997.
- [50] M. Schweiger, et. al., "The finite element method for the propagation of light in scattering media: boundary and source conditions," *Medical Physics*, vol. 22, no. 11, pp. 1779 – 1792, 1995.
- [51] T. Tarvainen, et. al., "Finite element model fro the coupled radiative transfer equation and diffusion approximation," *International Journal for Numerical Methods in Engineering*, vol. 65, no. 3, pp. 383 – 405, 2006.
- [52] S. Richling, et. al., "Radiative transfer with finite elements," *Astronomy & Astrophysics*, vol. 380, pp. 776 – 788, 2001.

- [53] E. Hull and T. H. Foster, “Steady-state reflectance spectroscopy in the P3 approximation,” *Journal of the Optical Society of America A*, vol. 18, no. 3, pp. 584 – 599, 2001.
- [54] B. Yuan, “Applications of delta-P1 and delta-P3 approximations to optical tomography in small imaging volumes,” *SPIE BiOS: Biomedical Optics*, International Society for Optics and Photonics, 2009.
- [55] D. J. Dickey, et. al., “Light Dosimetry using the P3 approximation”, *Physics in Medicine and Biology*, vol. 46, no. 9, pp. 2359 – 2370, 2001.
- [56] H. Shen and G. Wang, “A tetrahedron-based inhomogeneous Monte Carlo optical simulator”, *Physics in Medicine and Biology*, vol. 55, no. 4, pp. 947 – 962, 2010.
- [57] M. Firbank, et. al., “An investigation of light transport through scattering bodies with non-scattering regions”, *Physics in Medicine and Biology*, vol. 41, pp. 767 – 783, 1996.
- [58] D. Y. Churmakov, et. al., “Analysis of skin tissues spatial fluorescence distribution by the Monte Carlo simulation”, *J. Physics D: Applied Physics*, vol. 36, pp. 1722 – 1728, 2003.
- [59] R. Zhang, et. al., “Comparison of diffusion approximation and Monte Carlo based finite element models for simulating thermal responses to laser irradiation in discrete vessels”, *Physics in Medicine and Biology*, vol. 50, pp. 4075 – 4068, 2005.

- [60] G. Ma, et. al., “Comparison of simplified Monte Carlo simulation and diffusion approximation for the fluorescence signal from phantoms with typical mouse tissue optical properties”, *Applied Optics*, vol. 46, no. 10, pp. 1686 – 1692, 2007.
- [61] M. Cascella, et. al., “Comparison of Monte Carlo simulaitons based on Geant4 and Fluka to the TileCal test beam data” *Nuovo Cimento C*, vol. 32, pp. 425 – 428, 2009.
- [62] S. A. Prahl, et. al., “A Monte Carlo Model of Light Propagation in Tissue”, *Dosimetry of laser radiation in medicine and biology*, vol. 5, pp. 102 – 111, 1989.
- [63] N. S. Zolek, et. al., “Optimization of the Monte Carlo code for modeling of photon migration in tissue”, *Computer methods and programs in biomedicine*, vol. 84, no. 1, pp. 50 – 57, 2006.
- [64] Q. Fang and D. A. Boas, “Monte Carlo simulation of photon migration in 3D turbid media accelerated by graphics processing units”, *Optics Express*, vol. 17, no. 22, pp. 20178 – 20190, 2009.
- [65] X. Lu and P. Hsu, “Reverse Monte Carlo method for transient radiative transfer in participating media”, *J. Heat Transfer*, vol. 126, no. 4, pp. 621 – 627, 2004.
- [66] E. Margallo-Balbas and P. J. French, “Shape based Monte Carlo code for light transport in complex heterogeneous tissues”, *Optics Express*, vol. 15, no. 21, pp. 14086 – 14098, 2007.
- [67] N. Ren, et. al., “GPU-based Monte Carlo simulation for light propagation in complex heterogeneous tissues”, *Optics Express*, vol. 18, no. 17, pp. 6811 – 6823, 2010.

- [68] L. Wang and S. Jacques, “Monte Carlo modeling of light transport in mutli-layered tissues in standard C”, Ph. D. Dissertation, Univ. Texas, MD Anderson Cancer Center, Houston, U.S.A., 1992.
- [69] D. A. Boas, et. al., “Three dimensional Monte Carlo code for photon migration through complex heterogeneous media including the adult human head”, *Optics Express*, vol. 10, no. 3, pp. 159 – 170, 2002.
- [70] S. L. Jacques, “Light distributions from point, line and plane sources for photochemical reactions and fluorecence in turbid biological tissues”, *Photochemistry and Photobiology*, vol. 61, no. 1, pp. 23 – 32, 1998.
- [71] L. Wang, et. al., “MCML – Monte Carlo modeling of light transport in multi-layered tissues”, *Computer methods and programs in biomedicine*, vol. 47, no. 2, pp. 131 – 146, 1995.
- [72] L. V. Wang and H. Wu, “Monte Carlo Modelling of Photon Transport in Biological Tissue”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 3, sec. 3.2, pp. 37.
- [73] S. J. Jacques, “Monte Carlo Modeling of Light Transport in Tissue (Steady State and Time of Flight)”, in *Optical-Thermal Response of Laser-Irradiated Tissue*, 2<sup>nd</sup> ed. New York: Springer, 2011, chp. 5, sec. 5.3, pp. 115 – 129.

- [74] L. V. Wang and H. Wu, “Monte Carlo Modelling of Photon Transport in Biological Tissue”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 3, sec. 3.2, pp. 50 – 54.
- [75] L. V. Wang and H. Wu, “Hybrid Model of Monte Carlo Method and Diffusion Theory”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 6, sec. 6.4, pp. 122 – 124.
- [76] B. Luo and S. He, “An improved Monte Carlo diffusion hybrid model for light reflectance by turbid media”, *Optics Express*, vol. 15, no. 10, pp. 5905 – 5918, 2007.
- [77] L. Wang and S. L. Jacques, “Hybrid model of Monte Carlo simulation and diffusion theory for light reflectance by turbid media”, *J. Opt. Soc. Am. A*, vol. 10, no. 8, pp. 1746 – 1752, 1993.
- [78] T. Hayashi, et. al., “Hybrid Monte Carlo-diffusion method for light propagation in tissue with a low-scattering region”, *Applied Optics*, vol. 42, no. 16, pp. 2888 – 2896, 2003.
- [79] G. Alexandrakis, et. al., “Monte Carlo diffusion hybrid model for photon migration in a two-layer turbid medium in the frequency domain”, *Applied Optics*, vol. 39, no. 13, pp. 2235 – 2244, 2000.
- [80] R. Habel, et. al., “Photon beam diffusion: A hybrid Monte Carlo method for subsurface scattering”, *Computer Graphics Forum*, vol. 32, no. 4, pp. 27 – 37, 2013.

- [81] Joseph, D., et. al., “Application of the discrete ordinates method to grey media in complex geometries using unstructured meshes”, Proc. Eurotherm73 on Computational Thermal Radiation in Participating Media, Mons, Belgium, 2003, vol. 11, pp. 97 – 106.
- [82] F. C. Lockwood and N. G. Shah, “A new radiation solution method for incorporation in general combustion prediction procedures”, Symposium (international) on combustion, vol. 18, no. 1, pp. 1405 – 1414, 1981.
- [83] J. S. Truelove, “Discrete-Ordinate Solutions of the Radiative Transfer Equation”, J. Heat Transfer, vol. 109, pp. 1048 – 1051, 1987.
- [84] K. A. Jensen, et. al., “On various modelling approaches to radiative heat transfer in pool fires”, Combustion and Flame, vol. 148, pp. 263 – 279, 2007.
- [85] J. S. Truelove, “Three-Dimensional Radiation in absorbing-emitting-scattering media using the discrete ordinates approximation”, J. Quantitative Spectroscopy and Radiative Transfer, vol. 39, no. 1, pp. 27 – 31, 1988.
- [86] D. Joseph, et. al., “Discrete Ordinates and Monte Carlo Methods for Radiative transfer Simulation applied to CFD combustion modelling”, ASME J. Heat Transfer, vol. 131, no. 5, (9 pages), 2009.
- [87] J. Liu, et. al., “Development of an unstructured radiation model applicable for two-dimensional planar, axisymmetric, and three-dimensional geometries”, J. Quantitative Spectroscopy and Radiative Transfer, vol. 66, no. 1, pp. 17 – 23, 2000.

- [88] D. Joseph, et. al., “Comparison of three spatial differencing schemes in discrete ordinates method using three-dimensional unstructured meshes”, *International journal of thermal sciences*, vol. 44, no. 8, pp. 851 – 864, 2005.
- [89] K. Peng, et. al., “Graphics processing unit parallel accelerated solution of the discrete ordinates for photon transport in biological tissues”, *Applied Optics*, vol. 50, no. 21, pp. 3808 – 3823, 2011.
- [90] K. Stamnes, et. al., “Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media”, *Applied Optics*, vol. 27, no. 12, pp. 2502 – 2509, 1988.
- [91] K. F. Evans, “The Spherical Harmonics Discrete Ordinate Method for Three-Dimensional Atmospheric Radiative Transfer”, *J. Atmospheric Sciences*, vol. 55, pp. 429 – 446, 1998.
- [92] S. Chandrasekhar, *Radiative transfer*, USA: Dover Publications, Inc., 2013.
- [93] E. Hecht, “Interference” in *Optics*, San Fransico, California: Adison Wesley, 2002, chp. 9, sec. 9.6.1, pp. 422.
- [94] L. V. Wang and H. Wu, “Introduction”, in *Biomedical Optics: Principles and Imaging*, Hoboken, New Jersey: John Wiley & Sons, 2007, chp. 1, sec. 1.8, pp. 14.
- [95] T. Binzoni, et. al., “The use of the Henyey-Greenstein phase function in Monte Carlo simulations in biomedical optics”, *Physics in Medicine and Biology*, vol. 51, no. 17, N313, 2006.

- [96] L. G. Henyey and J. L. Greenstein, “Diffuse radiation in the galaxy”, *Annales d’Astrophysique*, vol. 3, pp. 117 – 137, 1940.
- [97] P. J. Davis, “Circulant Matrices” in *Circulant Matrices*, New York: Chelsea Publishing, 1994, chp. 3, sec. 3.5, pp. 99 – 101.
- [98] R. M. Gray, “Toeplitz and Circulant Matrices: A review”, [Online]. Available: <http://www-ee.stanford.edu/~gray/toeplitz.pdf>
- [99] A. J. Welch and M. J. C. van Gemert, “Overview of Optical and Thermal Laser-Tissue Interaction and Nomenclature” in *Optical-Thermal Response of Laser-Irradiated Tissue*, 2<sup>nd</sup> ed. New York: Springer, 2011, chp. 1, sec. 1.2, pp. 8.
- [100] A. J. Welch, et. al., “Basic Interactions of Light with Tissue” in *Optical-Thermal Response of Laser-Irradiated Tissue*, 2<sup>nd</sup> ed. New York: Springer, 2011, chp. 2, sec. 2.3.1, pp. 16 – 19.
- [101] D. G. Altman and J. M. Bland, “Standard deviations and standard errors”, *British Medical Journal*, vol. 331, no. 7521, pp. 903, 2005.
- [102] *SPSS Statistics 17.0 Algorithms*, SPSS Inc., Chicago, IL, 2007, pp. 300 – 301.



# Appendix

## Summary of Code

As the Matrix Method is a novel technique for solving the fluence generated by a source, it was deemed necessary to include the actual code within this thesis. The code for the isotropic point source in an infinite turbid media from the Matrix Method's code will be included. The changes necessary to create the code to simulate the pencil beam within a semi-infinite turbid media will also be included. This should give future researchers the necessary guidance to set forth and find new ways to apply the Matrix Method to other problems, regardless of whether or not their research is in another field of research.

## A.1 Code for Matrix Method

Chapter 3 outlines the necessary steps in the theory behind the Matrix Method, however it does not outline in any detail the code used for the calculations. This is done intentionally, as the fundamental concepts can be carried over from one programming language to another. The sections within the appendix will serve to illustrate the code written to generate the results within this thesis. The code here is written for the MATLAB 2013b simulation software and uses the assumption that the number of pixels in both directions is odd and that the number of solid angle segments is even. The code is as follows.

### A.1.1 Initialize Variables

In this section of the program we initialize all of the variables within the code.

---

```
% POINT SOURCE IN INFINITE TURBID MEDIA AND
% PENCIL BEAM IN SEMI-INFINITE TURBID MEDIA.

% This m-file was created to calculate the point source or the Pencil Beam
% within a finite turbid media using the Matrix Method. The values are
% those used within this thesis.

% Clearing the Command Window.
clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%          Setting up Initial Parameters          %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Total Number of Iterations
Num_Iter = 3000;

% Absorption Coefficient (cm-1)
ua = 0.1;
```

```

% Scattering Coefficient (cm-1)
us = 100;

% Extinction Coefficient (cm-1)
ut = ua + us;

% Anisotropy
g = 0.9;

% Anisotropy Squared
g2 = g^2;

% Width of each pixel (40 μm x 40 μm)
dx = 0.004; % cm
dy = dx;

dxm = dx/100; % m
dym = dxm;

% Number of Solid Angles per pixel
Number_Solid_Angle_Segments_Max = 60;
NSASM = Number_Solid_Angle_Segments_Max;

% Number of pixels per row
Nx = 501;

% Number of pixels per column
Ny = 501;

% Center of the Grid
Nx_c = Nx*dx/2;
Ny_c = Ny*dy/2;

% Initial Radiance
Radiance = zeros (Ny, Nx, NSASM );

% Initial Absorbance
Absorbance = zeros (Ny, Nx);

% Number of pixels per row in each Convolution Masks
Nxm = 51;

% Number of pixels per column in each Convolution Masks
Nym = 51;

```

% Initial Convolution Masks definition.

CMF\_MC\_Flip = zeros (Nym, Nxm, NSASM);  
CMF\_High\_Acc = zeros (Nym, Nxm, NSASM);  
CMF = zeros (Nym, Nxm, NSASM);

% Center of the Convolution Masks

Nxm\_c = Nxm/2;  
Nym\_c = Nym/2;

% Recording the MSE and MAPE.

MSE\_Pt\_S = zeros (1,Num\_Iter);  
MAPE\_Pt\_S = zeros (1,Num\_Iter);

%%  
%%%%%%%%% Initializing the System Radiance %%%%%%%%%  
%%

% ISOTROPIC POINT SOURCE

% The initial radiance will be set to a value of 1. It will be emitted  
% equally across all solid angle segments.

Radiance (Nyc, Nxc, :) = 1/ NSASM;

% PENCIL BEAM

% The initial ballistic radiance will be set by using Beer's Law.

% Test variable for remaining in the media. 1 = true, 0 = false.

In\_media = 1;

% Initial Intensity of Beam.

Io = 1;

% As this is propagated perpendicular to the x-axis, the step size will be  
stepy = dy/500;

% Initial location of beam.

yi = 0;

% Initial pixel beam is in.

NYP = floor(yi/dy);

```

% Solid angle segment corresponding to 90°.
SA_90 = (NSASM /4) + 1;

% Running a loop for the ballistic intensity.

while In_media == 1

    % Step forward along beam axis.
    yi = yi + stepy;

    % Check if beam has reached a new pixel.
    test = floor(yi/dy)-NYP;

    % If the beam has reached a new pixel.
    if (test>0)

        % Determine fraction that has scattered.
        scat = Io*(1-exp(-ut*dy));

        % Note: We skip the  $\mu_s/\mu_t$  factor since this will be included with
        % the scattering matrices.

        % Determine remaining ballistic intensity.
        Io = Io*exp(-ut*dy);

        % Save scattered portion of the ballistic intensity.
        Radiance (NYP+1, Nxc, SA_90) = Radiance (NYP+1, Nxc, SA_90) + scat;

        % Save the absorbed portion of the ballistic intensity.
        Absorbance (NYP+1, Nxc) = Absorbance (NYP+1, Nxc) + (ua/ut)*scat;

        % Record new pixel which beam is currently in.
        NYP = floor (yi/dy);

        % Check to see if pixel beam is still in media.
        if ((NYP)>=Ny)

            % Set variable to show that we are no longer in the medium.
            In_media = 0;

        end

    end

end

end

```

## A.1.2 Create the Scattering Matrices

In this section the scattering matrices needed for the simulations will be created.

---

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Scattering Matrix Creation Section %%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% We will use a large number angles to calculate the scattering matrix.
```

```
dtheta = 0.0001;
```

```
Theta = -180:dtheta:180;
```

```
% Converting them to radians
```

```
Theta_r = Theta*pi/180;
```

```
% Obtaining the Henyey-Greenstein phase function. Since there are positive
% and negative values for Theta the magnitude of the phase function will be
% divided by 2.
```

```
P_Theta = 0.5*(0.5*(1-g2)*abs(sin(Theta_r))./(1+g2-2*g*cos(Theta_r)).^1.5);
```

```
% Now the triangle will be made. This represents a single solid angle
```

```
% segment centered at zero degrees.
```

```
Triangle = (360/NSASM-abs(Theta)).*((360/NSASM-abs(Theta))>0)/(360/NSASM);
```

```
% For the scattering matrix two will be created. These are for the cases
% where the entire solid angle segment is being scattered, and for when
% only the center angle is scattered.
```

```
% Entire solid angle segment scattered.
```

```
Scattering_SAS = cconv (Triangle, P_Theta);
```

```
% Center angle scattered.
```

```
Scattering_point = P_Theta;
```

```
% Reducing the length of the convolution from 2N+1 to N+1.
```

```
% As well, the central values will be taken.
```

```
index = (length(Scattering_SAS)-1)/4;
```

```
Scattering_SAS = Scattering_SAS((1+index):(end-index));
```

```
% Now the magnitude of the Scattering SAS variable will be scaled down.
```

```
magnitude = dtheta*trapz(Scattering_SAS);
```

```
Scattering_SAS = Scattering_SAS/magnitude;
```

```

% Now the magnitude of the Scattering point variable will be scaled down.
magnitude      = dtheta*trapz(Scattering_point);
Scattering_point = Scattering_point/magnitude;

% Difference between each value
d_values      = (length(Scattering_SAS)-1)/NSASM;

% Down sampling the convolved distribution.
S_SAS_down    = Scattering_SAS(1:d_values:end);

% Down sampling the single angle distribution.
S_pnt_down    = Scattering_point(1:d_values:end);

% Down sampling the angles.
Theta_down    = Theta(1:d_values:end);

% Shifting the convolved distribution around and removing the first point.
S_SAS_down    = [S_SAS_down((NSASM/2+1):end), S_SAS_down(2:(NSASM/2))];

% Shifting the initial distribution around and removing the first point.
S_pnt_down    = [S_pnt_down((NSASM/2+1):end), S_pnt_down(2:(NSASM/2))];

% Shifting the angles around and removing the first point.
Theta_down    = [Theta_down((NSASM/2+1):end), Theta_down(2:(NSASM/2))];

% Now the amplitudes of the sampled values are reduced so they sum to one.
% This ensures that conservation of energy is followed.
S_SAS_down    = S_SAS_down/sum(S_SAS_down);
S_pnt_down    = S_pnt_down/sum(S_pnt_down);

% And now the Scattering matrices for Matrix Method are created.
S = toeplitz(S_Phase_down);
S2= toeplitz(S_point_down);

% And as this is an isotropic medium we can multiply S by the scattering
% ratio. This saves calculation time.

% The solid angle scattering matrix
S_tri = (us/ut)*S;
% The center angle scattering matrix
S_HG  = (us/ut)*S2;

```

## A.1.3 Create the Projection Matrices

In this section the projection matrices needed for the simulations will be initialized. This portion of the code can be broken into two subsections. The first entails calculating some of the solid angle segments.

---

```
if (rem(NSASM,2)==1)
    % If number of solid angle segments odd, we calculate this many solid angle segments.
    Nmax = ceil(NSASM /2);
else
    % If number of solid angle segments even.
    if (rem(NSASM,4)>0)
        % If the number of solid angle segments is not divisible by four, we calculate this many
        solid angle segments.
        Nmax = ceil(NSASM /4);
    else
        % If the number of solid angle segments is divisible by four, we calculate this many solid
        angle segments.
        Nmax = floor(NSASM /8)+1;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creation of Convolution Masks Section
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initial location of the center of the Convolution mask.
x_cen = dx*Nxm_c/2;
y_cen = dy*Nym_c/2;
```



```

% Step distance
dzs = sqrt((dx)^2 + (dy)^2)/10;

% The step distance is smaller than the distance between each pixel because
% this step is used to tests if photons land inside the same pixel or in the next one.

% Number of Projections per Solid Angle.
NPPSA = 1001; % Odd as the center value (501) is at the tip of the triangle

% We cycle through each solid angle segment.
for J_ang = 1:Nmax

    % Create a temporary matrix to store the convolution mask for this
    % solid angle.
    Convolution_Mask = zeros (Nym, Nxm);

    % Initialize a count variable for the total intensity projected from
    % this solid angle.
    Total_Intensity = 0;

    % We cycle through each projection.
    for proj = 1:NPPSA

        % Determining the angle of the projection.
        angle = (360/NSASM)*((J_ang-1) + (-1 + 2*proj/(NPPSA+1)));

        % Convert this angle to radians
        angle = angle*pi/180;

        % Determine the scale factor. This scales the initial intensity
        % based on how far it is away this angle is from the center
        % of the solid angle segment. It is the weighing function.
        scale = 1 - abs(-1 + 2*proj/(NPPSA+1));

        % Saving the starting Intensity.
        Inten = scale;

        % Adding the new intensity to the total intensity.
        Total_Intensity = Total_Intensity + Inten;

        % Create the steps for each direction, and saving them.

        % X-step.
        step_x = dzs*cos(angle);
        step_x_save = step_x;

```

```

% Y-step.
step_y = dzs*sin(angle);
step_y_save = step_y;

% Setting and saving the initial location.

x = x_cen; NXP = floor(x_cen/dx); xold = x_cen;
y = y_cen; NYP = floor(y_cen/dy); yold = y_cen;

% Initializing scale tester.
test2 = 0;

% Initialize a determination variable.
finish = 0;
% finish becomes one when the angle reaches the edge of the mask.

% Run the loop to determine the stopping locations.
while finish == 0

    % Move forward a step.
    x = x + step_x;
    y = y + step_y;

    % Check to see if it is still in the same pixel
    NX = floor(x/dx); NY = floor(y/dy);

    test = abs(NX-NXP) + abs(NY-NYP);

    % If we have moved into a new voxel by a small enough distance.
    if (test2 == 3);

        % If we have moved by a small enough value
        if (test > 0)

            % Calculate the distance from the last crossing.
            dist = sqrt((x-xold)^2 + (y-yold)^2);

            % Determine the fraction of photon packets scattered in this
            % pixel along the chosen line.
            scat = Inten*(1-exp(-dist*ut));

            % Add that amount to the temporary matrix.
            Convolution_Mask(NYP,NXP) = Convolution_Mask(NYP,NXP) + scat;

```

```

    % Update the intensity for the remaining photon packets
    Inten = Inten*exp(-dist*ut);

    % Save this new location as the initial spot.
    yold = y; NYP = NY;
    xold = x; NXP = NX;

    % Run a test system to determine if the line crosses a
    % border of the convolution mask.
    finish = (NX>Nxm)+(NY>Nym)+(NX<1)+(NY<1);

    % Rescale the step sizes; Reset test2
    step_x = step_x_save;
    step_y = step_y_save;

    test2 = 0;

end

else % (If the step size wasn't small enough).

    % Count the case
    test2 = test2 + 1;

    % Move backwards by a step.
    x = x - step_x;
    y = y - step_y;

    % Reduce the step size by a factor of 10.
    step_x = step_x/10;
    step_y = step_y/10;

end
end
end

% Scale the Mask to the value of one. This allows the projection to
% obey conservation of energy.
Convolution_Mask = (Convolution_Mask/Total_Intensity);

% Store the Convolution Mask Matrix in the overall Mask Matrix.
CMF(:,J_ang) = Convolution_Mask;

end

```

The remainder of the code represents the second part of the creation of the projection matrices. Here the copies of the calculated solid angle segments are made and pasted to other solid angle segments. This reduces calculation time.

---

## % COPYING AND PASTING SOLID ANGLE SEGMENT PROJECTIONS.

% Cycling through all of the solid angle segments.

for J\_sang = 1:Nmax

% Saving the solid angle segment.

Convolution\_Mask2 = CMF(:, :, J\_sang);

% Recording the solid angle segment in the CMF Flip variable.

CMF\_MC\_Flip(:, :, J\_sang) = Convolution\_Mask2;

% Flipping the saved solid angle segment upside down.

Convolution\_Mask\_ud = flipud(Convolution\_Mask2);

% Recording the flipped solid angle segment in the appropriate spot.

if (J\_sang > 1)

CMF\_MC\_Flip(:, :, (NSASM - J\_sang + 2)) = Convolution\_Mask\_ud;

end

% If the solid angle segments are even (ie. divisible by 2) then there

% are 4 masks which can be found from calculating one value.

if (rem(NSASM, 2) == 0)

% Flipping the solid angle segment left-right and saving it.

Convolution\_Mask\_lr = fliplr(Convolution\_Mask2);

CMF\_MC\_Flip(:, :, (NSASM / 2 - J\_sang + 2)) = Convolution\_Mask\_lr;

% Flipping the upside down solid angle segment left-right and saving it.

Convolution\_Mask\_ud\_lr = fliplr(Convolution\_Mask\_ud);

CMF\_MC\_Flip(:, :, (NSASM / 2 + J\_sang)) = Convolution\_Mask\_ud\_lr;

% If in fact the number of solid angle segments are divisible by 4,  
% then there are 8 masks which can be determined from one mask.

if (rem(NSASM,4)==0)

% Rotating each of the masks by 90 degrees.

Convolution\_Mask2 = rot90(Convolution\_Mask2,3);

Convolution\_Mask\_lr = rot90(Convolution\_Mask\_lr,3);

Convolution\_Mask\_ud = rot90(Convolution\_Mask\_ud,3);

Convolution\_Mask\_ud\_lr = rot90(Convolution\_Mask\_ud\_lr,3);

% Saving each of the rotated masks.

CMF\_MC\_Flip(:,:(NSASM /4+J\_sang)) = Convolution\_Mask2;

CMF\_MC\_Flip(:,:(NSASM /4-J\_sang+2)) = Convolution\_Mask\_ud;

CMF\_MC\_Flip(:,:(3\*NSASM /4+J\_sang)) = Convolution\_Mask\_ud\_lr;

CMF\_MC\_Flip(:,:(3\*NSASM /4-J\_sang+2)) = Convolution\_Mask\_lr;

end

end

% Saving the final masks.

CMF\_High\_Acc = CMF\_MC\_Flip;

end



```

% FOR BALLISTIC PENCIL BEAM

% Rearranging the Radiance to improve computational time

% Cycling through each solid angle segment
for j = 1: NSASM

    % Obtaining the radiance within solid angle j
    RI = Radiance(:,j);

    % And saving the remaining radiance for scattering.
    M(j,:) = (RI(:))';

end

% We no longer need the variable Radiance as it is now saved matrix M.
clear Radiance

% And scattering the ballistic portion.
M = S_HG*M;

% FOR BOTH SIMULATIONS
ua_m = ua*100;

% This measures the MSE and MAPE. (FMC_S can be the MC solution to either the Pencil
Beam or the Point Source)
Elem = Absorbance/(ua_m*dxm*dym);
Scale = (size(Elem,1)*size(Elem,2));
MSE_Pt_S(1) = sum(sum(((Elem-FMC_S).^2)))/Scale;
MAPE_Pt_S(1) = 100*sum(sum(abs((Elem-FMC_S)./FMC_S)))/Scale;

% From here the process is repeated until all of the iterations are done.

for iterat = 2:Num_Iter

    for j = 1: NSASM

        % Obtaining the radiance within solid angle j
        RI = M(j,:);

        % Reshaping the Vector into a Matrix.
        RI = reshape(RI', Ny, Nx);

        % Obtaining the mask for solid angle j.
        Ms = CMF_High_Acc(:, :, j);

```

```
% Conducting the Convolution.
```

```
Rl = conv2(Rl,Ms,'same');
```

```
% We save the amount absorbed.
```

```
Absorbance = Absorbance + (ua/ut)*Rl;
```

```
% And from here we save the rest for scattering.
```

```
M(j,:) = (Rl(:))';
```

```
end
```

```
% The scattering portion is easily done.
```

```
M = S_tri*M;
```

```
% This measures the MSE and MAPE.
```

```
Elem = Absorbance/(ua_m*dxm*dym);;
```

```
MSE_Pt_bound(iterat) = sum(sum(((Elem-FMC_S).^2))/Scale;
```

```
MAPE_Pt_bound(iterat) = 100*sum(sum( abs((Elem-FMC_S)./FMC_S)))/ Scale;
```

```
end
```

```
Fluence_Final = Absorbance/(ua_m*dxm*dym);
```



## A.2 Code for Monte Carlo Method

The Monte Carlo code here is written for the MATLAB 2013b simulation software based off of current literature. The code is as follows.

---

```
% No averaging in code.

clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Setting up Initial Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Total Number of Photons for Point Source
kkkmax = 12500000;

% Total Number of Photons for Pencil Beam
kkkmax = 50000000;

% Medium Coefficients
g = 0.9;

% Length of each side of each pixel (cm)
dx = 0.004;
dy = 0.004;
dx_m = dx/100; dy_m = dy/100; % (m)

% Absorption Coefficient of each layer (cm-1)
Ua = 0.1;
Ua_m = 10; % (m-1)

% Scattering Coefficient of each layer (cm-1)
Us = 100;

% Total Interaction Coefficient
Ut = Ua + Us;

% Critical weight for the albedo
Wcrit = 0.001;

% Number of pixels in the medium.
Ny = 501;
Nx = Ny;
```

```

y = Ny*dy; % The depth of the grid
x = y; % The width of the grid.

% establishing the Grid which represents the medium.
Grid = zeros (Ny, Nx);

% Initializing the absorbance.
A = Grid;

% Once the photon reaches a critical weight we check to see if it survives.
% These are simply parameters which allow for the testing of the photon's survival.
% Below we have initialized m, where 1/m is the chance the photon will survive the roulette.
m = 10;

% These are simply parameters which allow for estimating the time it takes for the simulation to
run.

Last_Photon = 10000; % The number of photons per projection subset.
kkkmax = kkkmax/Last_Photon; % The number of projection subsets.

% Displaying the simulation start time for the user.
c1=clock; Time_start = datestr (c1);

fprintf('This program started at %s and will run %d Photons\n', Time_start,
kkkmax*Last_Photon)

% Adding a vector to count the number of times a photon scatters prior to terminating.
Scatter_Counts = zeros (1, 500000);

% Iterating through the projection subsets
for kkk = 1:kkkmax

    % Iterating through each photon within a subset
    for Pht = 1:Last_Photon

        % The photon is represented as a structure variable. In this case the
        % position, momentum, weight, life, the step size, and the number of
        % scattering events.

        % Point source initial parameters.
        Initial_location = [dx*Nx/2,dy*Ny/2]; % Located at x=0, y=0
        a = 2*pi*rand;
        Initial_momentum = [cos(a),sin(a)]; % Initially direction of travel.
    end
end

```

```

% Pencil Beam initial parameters.
Initial_location = [dx*Nx/2, 0]; % Located at x=0, y=0
Initial_momentum = [0,1]; % Initially direction of travel.

% Parameters for both simulations.
Initial_weight = 1; % Set to unity for scaling.
Initial_alive = 1; % 1 means traveling, 0 means dead.
Initial_scatters = 0; % Number of scattering events experienced.
Initial_stepsize = 0; % Initial Step size for the photon.

% The Photon is set to an initial value.
Ph = struct; % Creating a structure
Ph.location = Initial_location; % Initial location
Ph.test = Initial_location; % For finding the boundary
Ph.momentum = Initial_momentum; % Initial momentum
Ph.weight = Initial_weight; % Initial weight
Ph.life = Initial_alive; % Starts at 1 - when dead becomes 0
Ph.scatter = Initial_scatters; % Number of scattering events
Ph.stepsize = Initial_stepsize; % Distance the photon travels

% From here we have the photon in the medium!!!!!!
while Ph.life == 1

    % As the photon is in the medium we now have to determine the step size
    % it will take

    if Ph.stepsize == 0

        % determining the new step size
        Ph.stepsize = -1*log(rand)/Ut;

    end

    % Now the photon moves using the step size.
    Ph.test = Ph.location + Ph.momentum*Ph.stepsize;

    % Now we need to determine if the photon will cross into the next boundary.

    if ((Ph.test(2)<=0)||((Ph.test(2)>=y)||((Ph.test(1)<=0)||((Ph.test(1)>=x)))

        % This section runs if the step size is bigger than the distance to
        % the next boundary. As it crossed the boundary the photon terminates.
        Scatter_Counts(Ph.scatter) = Scatter_Counts(Ph.scatter)+1;
        break;

    else

```

```

% This section runs if the step size is smaller than the distance to
% the next boundary. We move the remainder of the step size, change
% the step size to a value of zero, and determine the absorbance and
% scattering components.

% Step one is to move the remainder of the step size
Ph.location = Ph.location + Ph.momentum*Ph.stepsize;

% Step two is to change the step size to zero
Ph.stepsize = 0;

% Step three is to analyze the absorption and scattering components

% For Absorption we modify the weight with a simple equation, then
% we simply record the weight and setup the photon for scattering

% Calculate the drop in weight
weight_drop = Ph.weight*(Ua/Ut);

% Get and Save the Photon's new weight
new_weight = Ph.weight - weight_drop;
Ph.weight = new_weight;

% Save the fraction absorbed
xa = ceil(Ph.location(2)/dx);
ya = ceil(Ph.location(1)/dy);
A (xa, ya) = A (xa, ya) + weight_drop;

% Testing to ensure the photon is within the medium and not on the edge.
test = sum([(Ph.location(1)<dx*(Nx+1)), (Ph.location(2)<dy*(Ny+1)),
(Ph.location(2)>0), (Ph.location(1)>0)]);

if ( test < 4 )

    % Kill Photon - its beyond the boundaries.
    Ph.weight = 0;
    Ph.life = 0;
    Scatter_Counts(Ph.scatter) = Scatter_Counts(Ph.scatter)+1;

end

% For Scattering we just run the scattering function shown below.

% This is the random factor for the Theta coordinates
E = rand;

```

```

% If g does not equal 0 - in this case g = 0.9
cosTheta = (1/(2*g))*(1 + g^2 - ((1-g^2)/(1-g+2*g*E))^2);

% and to get the sin of theta we use a trig identity sin^2 + cos^2 = 1.
% Since theta ranges from 0 to pi sin theta is always positive.
sinTheta = sqrt(1 - cosTheta^2);

% Keeping the original momentums as u factors
ux = Ph.momentum(1);
uy = Ph.momentum(2);

% Determining if the scattering direction is to the left or right.
F = 2*((rand>0.5)-0.5);

% Determining the new direction of photon propagation.
mx = cosTheta*ux + F*sinTheta*uy;
my = cosTheta*uy - F*sinTheta*ux;

% Scaling the momentum to unity.
mag = sqrt(mx^2 + my^2);
mx = mx/mag;
my = my/mag;

% Now we update the momentums.
Ph.momentum = [mx, my];

% And we add 1 to the number of scattering events the photon has undergone.
Ph.scatter = Ph.scatter + 1;

end

% Now we need to perform the death section of the Ph. In this part
% we will first run a simple check to see if the photon is
% dead - ie left the area of interest or the weight is too small. If
% the photon has left it is already listed as dead - otherwise we check
% the weight to make sure the photon is still worth following.

if Ph.life == 1 % The photon is still alive so run the weight check

% Now we check and see if the photon is too small to care about
if (Ph.weight <= Wcrit)

% Run the survival roulette.
% We start by generating a pseudo-random number.

```

```

E = rand;

% From here we determine the fate of the Ph. The photon is given a 1 in
% m chance of surviving. If it survives its weight increases by m fold.

if E < 1/m

    % Here the photon survives.
    Ph.weight = m*Ph.weight;

else

    % Here the photon dies.
    Ph.weight = 0;
    Ph.life = 0;
    Scatter_Counts (Ph.scatter) = Scatter_Counts (Ph.scatter) + 1;

end
end
end
end
end

```

% Determining the time it takes to run one subset, and from it estimating the time to run all of the remaining subsets.

```

if kkk == 1
    c = clock;

    % Determines the time for a subset in hours, minutes and seconds. This is then converted to
    % minutes and seconds, and displayed to the user.
    Rate = c(4:6)-c1(4:6);
    Total = 60*60*Rate(1)+60*Rate(2)+Rate(3);
    minutes = floor(Total/60);
    seconds = round(Total-60*minutes);
    fprintf('It has taken %d minutes and %d seconds to run %d Photons\n\n', minutes, seconds,
Last_Photon)

    % Determines the time for the entire simulation in hours, minutes and seconds. This is then
    % converted to minutes and seconds, and displayed to the user.
    TotalL = Total*(kkkmax-1);
    hours = floor(TotalL/3600);
    minutes = floor((TotalL-3600*hours)/60);
    seconds = round(TotalL-60*minutes-3600*hours);
    fprintf('I will estimate that this program will be done in %d hours, %d minutes and %d
seconds.\n\n', hours, minutes, seconds)

```

```

c    = clock;
Time2 = datestr(c);
fprintf('This time now is %s\n', Time2)

end

fprintf('We are now completing run %d out of %d.\n\n', kkk, kkkmax)

end

% Informing the user the program has finished, and how long it took.
c1 = clock;
Time_finish = datestr (c1);
fprintf ('This program started at %s, finished at %s, and ran %d Photons.\n', Time_start,
Time_finish, kkkmax*Last_Photon)

% Fluence is the fraction absorbed divided by the number of photons,
% divided by the area per pixel, divided by the absorption coefficient.

Fluence_MC = A/(Ua_m*kkkmax*Last_Photon*dx_m*dy_m);

% Saving the Fluence as a text file and as a matlab file.

fName = 'Fluence_MC_raw_data.txt'; fid = fopen (fName,'w');
fprintf (fid,'%e\n',Fluence_MC); fclose (fid);

save('Fluence_MC_raw_data_photons', 'Fluence_MC', 'Scatter_Counts')

% To read the variable back from text file.
% fName = 'Fluence_MC_raw_data.txt'; fid = fopen (fName,'r');
% Fluence_MC = fscanf (fid, '%e'); fclose (fid);
% Fluence_MC = reshape (Fluence_MC, 501, 501);

```