2007-06-07

Computationally Efficient Modeling of Transient Radiation in a Purely Scattering Foam Layer

Rudolph Scott Larson
Brigham Young University - Provo

COMPUTATIONALLY EFFICIENT MODELING OF TRANSIENT

RADIATION IN A PURELY SCATTERING FOAM LAYER

by

R. Scott Larson

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

Brigham Young University

August 2007

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

R. Scott Larson

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Matthew R. Jones, Chair

Date	Vladimir P. Solovjov

Date	Brent W. Webb

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of R. Scott Larson in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

_____   _____
Date   Matthew R. Jones
                                                Chair, Graduate Committee

Accepted for the Department

_____   _____
Date   Larry L. Howell
                                                Department Chair

Accepted for the College

_____   _____
Date   Alan R. Parkinson
                                                Dean, Ira A. Fulton College of Engineering
                                                and Technology

ABSTRACT

COMPUTATIONALLY EFFICIENT MODELING OF TRANSIENT

RADIATION IN A PURELY SCATTERING FOAM LAYER

R. Scott Larson

Department of Mechanical Engineering

Master of Science

An efficient solution method for evaluating radiative transport in a foam layer is a valuable tool for predicting the properties of the layer. Two different solution methods have been investigated.

First, a reverse Monte Carlo (RMC) simulation has been developed. In the RMC simulation photon bundles are traced backwards from a detector to the source where they were emitted. The RMC method takes advantage of time reflection symmetry, allowing the photons to be traced backwards in the same manner they are tracked in a standard forward Monte Carlo scheme.

Second, a reduced order model based on the singular value decomposition (ROM) has been developed. ROM uses solutions of the reflectance-time profiles found for specific values of the governing parameters to form a solution basis that can be used to generate the profile for any arbitrary values of the parameter set.

The governing parameters that were used in this study include the foam layer thickness, the asymmetry parameter, and the scattering coefficient. Layer thicknesses between 4 cm and 20 cm were considered. Values of the asymmetry parameter varied between 0.2 and .08, while the scattering coefficient ranged from 2800 m^{-1} to 14000 m^{-1}.

Ten blind test cases with parameters chosen randomly from these ranges were run and compared to an established forward Monte Carlo (FMC) solution to determine the accuracy and efficiency of both methods. For both RMC and ROM methods the agreement with FMC is good. The average difference in areas under the curves relative to the FMC curve for the ten cases of RMC is 7.1% and for ROM is 7.6%. One of the ten cases causes ROM to extrapolate outside of its data set. If this case is excluded the average error for the remaining nine cases is 5.3%. While the efficiency of RMC for this case is not much greater than that of FMC, it is advantageous in that a solution over a specified time range can be found, as apposed to the FMC where the entire profile must be found. ROM is a very efficient solution method. After a library of solutions is developed, a separated solution with different parameters can be found essentially in real-time. Because of the efficiency of this ROM it is a very promising solution technique for property analysis using inverse methods.

ACKNOWLEDGMENTS

There are few things that we can really take credit for accomplishing on our own. I have had a great deal of help with this project and I would like to acknowledge the many people who have made it possible for me to complete this work.

First, I would like to acknowledge Dr. Matthew Jones for his willingness to work with me on this project, for his support, his friendship, his counsel, and his funding. I acknowledge Dr. Vladimir Solovjov for his friendship and his willingness to answer all of my questions in Russian, and Dr. Brent Webb for being willing to be part of my committee despite his extremely busy schedule.

Next, I would like to mention the support of my sweet wife Lesley. Without her I would not have been able to finish this work. She demonstrated extreme amounts of patience with me as I struggled through this research.

Lastly, and most importantly, I would like to recognize all of the little miracles and answered prayers that made this work possible.

TABLE OF CONTENTS

LIST OF TABLES

x

LIST OF FIGURES

1   Introduction

1.1   Motivation

Foams are formed during a wide variety of industrial processes, and the presence of a foam layer significantly impacts the outcome of these processes. Due to the widespread existence and importance of foam, techniques for characterizing industrial foams are of interest. Knowledge of the structural, radiative and thermophysical properties of foams will lead to greater ability to model the formation and stability of foams and to model the heat and mass transfer in foams. Clarification of the physical mechanisms involved in the production of foam and transport phenomena occurring in foam will lead to the ability to mitigate their undesirable effects and enhance their desirable characteristics in numerous industrial applications.

A spectroscopic method of characterizing industrial foams has been developed [1]. An essential element of this method is a model of the propagation of a laser pulse through a foam layer. The propagation of the laser pulse can be modeled with a high degree of accuracy using Monte Carlo (MC) simulations. However, MC simulations are computationally expensive. Approximate models based on diffusion theory have also been developed. Simulations based on diffusion theory can be performed rapidly, but these simulations are accurate for a limited range of properties. The objective of this

research has been to develop an accurate and computationally efficient method for modeling the propagation of a laser pulse through a foam layer. Ultimately, this model will be used as part of an inverse algorithm capable of characterizing the properties of foam layers which form during industrial processes.

A reverse Monte Carlo (RMC) [2] simulation method and a reduced order model (ROM) [3] based on the singular value decomposition (SVD) have been developed as alternatives to the standard Monte Carlo method. The accuracy and computational efficiency of these alternative methods were assessed by comparing results obtained using these methods with results obtained from a previously developed forward Monte Carlo code [1].

1.2   Glass Foam

Liquid foam is part of our everyday life. We see it in soaps, cleaning agents, shaving products, and beverages. Its properties are exploited in such applications as enhanced oil recovery, fire fighting, insulation products, the production of chemicals, and it has even been suggested as a means to control civil disorder [4].

However, in many applications the formation of foam is a very costly and undesirable side-effect. Foam that forms in chemical reactors and food processing applications insulates the batch materials, preventing efficient heat transfer. The formation of foam is particularly detrimental in the glass manufacturing industry.

The cost and quality of almost all commercial glass products are determined by the performance of the glass melting and delivery systems (furnaces), which, in turn depend on the efficiency of the heat transfer from the furnace to the raw materials and the

2

molten glass. In a typical glassmelt about one-third of the surface of the molten glass is covered by foam of varying thickness [5]. It is estimated that as much as 60% of the radiative heat flux is blocked or scattered by this foam layer [6]. This insulating effect results in a significant reduction in energy efficiency of the furnace.

A foam layer in glassmelts also can reduce the quality of the glass. The loss of heat from the furnace reduces the temperature of the melt. A lower temperature means lower rates of refinement. Unwanted gas bubbles are left in the melt, decreasing the quality of the glass. This decrease in temperature also leads to a decrease in productivity because the amount of time required to achieve a satisfactory quality of glass is increased.

In addition to economic impacts on the glass manufacturing industry, glass foam leads to harmful environmental effects. To overcome the additional resistance to heat transfer created by a foam layer, glass melting furnaces are operated at elevated temperatures. In addition to increasing the amount of fuel required, higher temperature operation leads to an increase in NO_x gas emissions.

1.3   Spectroscopic Methods

Much effort has been devoted to mathematically modeling heat and mass transfer in glass furnaces in order to increase the efficiency and decrease the negative environmental impact associated with glass manufacturing [7]. Methods of characterizing industrial foams are an essential element in these modeling efforts.

Present methods of characterizing foams generally require carefully controlled laboratory conditions [4]. Due to the delicate nature of foams, properties of samples

3

extracted from a process and examined in a lab are unlikely to be representative of the properties of the foam as it is formed in the furnace. Therefore, further improvements in heat and mass transfer modeling depends largely on being able to measure the structural, radiative, and thermophysical properties of glass foam in situ [1]. The ultimate goal of this research will be to improve a method for characterizing the properties of industrial foams.

A common method of non-invasively determining properties of a substance is spectroscopy. Spectroscopy involves probing a material with energy (in the form of x-rays, visible light, infrared radiation, ultrasound, heat, etc.). A spectroscopic method of measuring the properties of foam in a glass furnace has been proposed by previous researchers at BYU [1]. In this method a pulsed laser beam is used to illuminate the foam layer. The laser radiation is absorbed and scattered as it travels through the foam layer. A portion of the radiation passes through the foam and into the glass melt. Some of the radiation is reflected out of the foam (see Fig. 1-1). This reflected radiation is measured at different radial locations away from the laser. Using the reflectance time profile obtained from this measurement technique an inverse method can be used to obtain the properties of the foam. Because the inverse method requires multiple solutions of the forward model of the radiation process a computationally efficient model is essential. The crux of this research is to develop a more efficient model of the radiative transport in the foam layer.
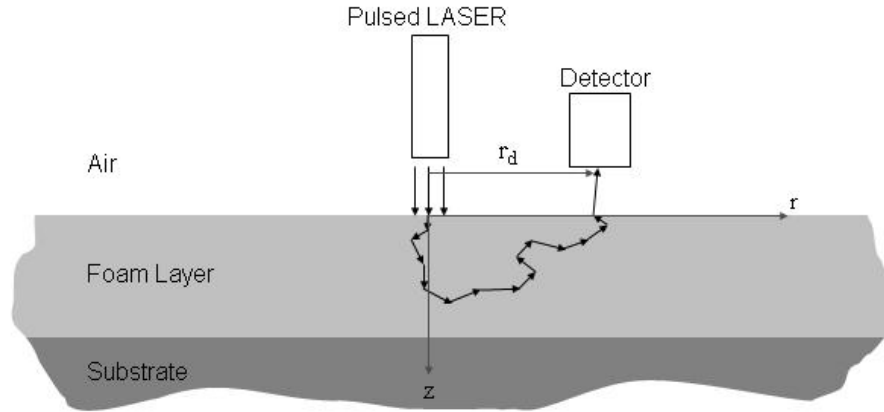
Figure 1-1: Spectroscopic investigation of a foam layer.

Propagation of the laser beam through a non-absorbing foam layer has been modeled using a forward Monte Carlo (FMC) approach [1]. However, use of FMC simulations to solve the inverse problem is infeasible due to the computational expense. Depending on the properties of the foam layer (thickness, scattering coefficient and asymmetry parameter), one MC simulation will require anywhere from 20 to 200 hours on a SGI Origin supercomputer with 64 MIPS processors. Computationally efficient approximate methods based on the diffusion approximation have also been explored, but these methods are not sufficiently accurate in all cases of interest [1]. Therefore, the need for a robust, computationally efficient method for predicting the time-dependent reflectance profiles exits.

Currently, the most commonly used methods for the analysis of radiative heat transfer are: integral equation models, the discrete ordinates method or finite volume method, the diffusion approximation, and Monte Carlo methods [8]. All of these methods involve solving the radiative transfer equation for the intensity field throughout the

5

system. In this research a computationally efficient reverse Monte Carlo (RMC) algorithm has been developed. RMC takes advantage of the principle of reciprocity in radiative transport to calculate the reflected flux at the detector by backward tracing the photons that reach the detector, eliminating the need to calculate the entire field. Also, a simulation method based on reduced order or spectral methods has been employed. This method is based on the singular value decomposition (SVD) of a matrix of reflectance profiles that where found using another technique (e.g. FMC or RMC). The combination of the RMC model and the reduced order model (ROM) are useful tools that can be used to solve the forward problem in near real-time, opening the door to the use of inverse methods to solve for the sought after foam properties.

1.4   Thesis Outline

This thesis will discuss the development of a reverse Monte Carlo model and the reduced order model based on SVD that are used to model the reflected flux exiting a purely anisotropic scattering medium that is irradiated by a pulsed laser. The solutions are dependent on the following foam properties: foam layer thickness (L), scattering coefficient (μ_s), and the asymmetry parameter (g).

Chapter 2 contains a summary of previous work done on modeling radiative transport in participating medium used to characterize material properties as well as a discussion on the significant literature pertaining to RMC and reduced order modeling based on SVD. In Chapter 3 the development of the reverse Monte Carlo model is discussed in detail. Chapter 4 contains a detailed description of the reduced order model and Chapter 5 contains the results of the comparisons of the two models with an

6

established FMC model. Chapter 6 concludes the body of the thesis with a summary of

the research, conclusions made, and comments on future research. Following Chapter 6 is

a list of works sited in this thesis. Appendix A and B contain a full catalog of results for

cases that were run using the two models developed. Appendix C contains the source

code for RMC written in C++ and Appendix D contains the source code for ROM written

in Matlab.

2   Previous Work

2.1   Introduction

The study of radiative transport in participating medium, particularly in glass

foams has been the subject of research in the past years. Aided by the development of

computing resources and new and improved solution methods, a number of radiative

transfer problems that were previously unsolvable have now been solved. This chapter

will discuss the main methods of solution and summarize the research that has been done.

2.2   Current Models for Transient Radiative Processes

Efforts have been made to develop computationally efficient radiation models of

transient radiation in participating medium. This section describes some of the main

methods used including: the discrete ordinate method, finite volume method, integral

equation models, diffusion approximation, and Monte Carlo schemes.

2.2.1   Discrete Ordinates Method

The discrete ordinate method is a descendant of the *Schuster-Schwarzschild* or

two-flux approximation [9, 10]. In this method the intensity is discretized and assumed to

be constant over specific solid angles. Although relatively efficient computationally,

there is a large memory requirement and communication overhead that are major drawbacks for implementing this method [11, 12].

2.2.2 Finite Volume Method

The finite volume method is similar to the discrete ordinates method except it uses finite volumes not only for the spatial discretization but for the directional discretization as well. This approach was first implemented by Briggs et al. [13]. Raithby gives a good review of this method [14]. Lu and Hsu applied finite volumes for the solution of light pulse propagation in three-dimensional scattering medium [15]. This method suffers from the same shortcomings as the discrete ordinates method because of the large memory requirement.

2.2.3 Integral Equation Models

Typically integral formulations are used to develop an analytical model of the fundamental transport phenomenon. The analytical model is then solved using numerical quadrature or similar numerical schemes. Tan and Hsu solved the transient problem using an integral formulation [16]. Integral equation models are well suited for implementation in parallel computing schemes and are fairly accurate, but are difficult to use in the cases with reflective boundaries [17]

2.2.4 Diffusion Approximation

For optically thick medium a simplifying approximation can be made to the RTE so that a closed form solution can be found. Salisbury used a diffusion approximation to

solve the problem of a pulsed laser incident on a participating foam layer with a detector measuring the reflected radiative flux [1]. This diffusion theory approximation was successfully implemented in an inverse algorithm to predict the foam layer thickness and reduced scattering coefficient. Although the diffusion approximation is computationally efficient, it is only accurate when the detector is located a sufficient distance from the source. Since the signal to noise ratio decreases rapidly as the source-detector separation increases. Limitations on the proximity of the detector to the source severely restrict the use of the diffusion approximation in inverse techniques.

2.2.5   Monte Carlo Simulations

Monte Carlo schemes are the widely used as benchmark solutions and have countless applications [8]. They can be used to stochastically solve problems that may not have a deterministic solution. In radiative transfer problems, often there is no closed form solution and there is no known solution technique. Monte Carlo schemes can be used to solve complex radiation problems that may include complicated geometries, varying radiative properties, scattering, or other complications that may make the solution extremely difficult or even impossible to find.

The main idea behind Monte Carlo solutions to radiative transport problems is to follow the random path of photons as they propagate through a medium encountering scattering and or absorbing events. Random numbers and probabilities based on the properties of the medium are used to determine the location, direction, and length of scattering events. Salisbury developed a forward Monte Carlo scheme to solve the problem at hand [1].

Although reliable, Monte Carlo schemes are not computationally efficient. In order to obtain statistically meaningful results hundreds of thousands of photon bundles have to be tracked and recorded. To improve the efficiency of traditional Monte Carlo schemes, a number of techniques have been introduced including reverse or backward Monte Carlo.

2.3   Reverse Monte Carlo

Reverse Monte Carlo takes advantage of the principle of reciprocity in radiative transport to calculate the reflected flux at the detector by backward tracing the photons that reach the detector, eliminating the need to track bundles that never intercept the detector. The idea of RMC has been applied in several studies [18-24]. Collins et al. [19] reported the earliest work on RMC relevant to radiative heat transfer. All of these studies dealt with large light sources, which lead to a straightforward implementation of RMC. A method for implementing RMC to cases with a small energy source is described by Modest [2]. One of the challenges of this study is to develop a method for applying RMC to cases with a small energy source and a small detector. The work presented by Modest is for a steady radiative source in a 2-D medium. Lu and Hsu describe a RMC simulation for a pulsed source in a 3-D medium [25, 26]. The work of both of these researches has been referenced in the development of the RMC simulation used in this research.

2.4   Reduced Order Modeling

Often it is advantageous to obtain a low-dimensional approximation of a high-dimensional process. The main idea of these methods is to obtain an ordered set of basis

12

vectors that represent the dominant features of the data. There are a number of modeling techniques that rely on the singular value decomposition to solve for the basis. One such technique that is referred to by different names but is essentially equivalent is known as Proper Orthogonal Decomposition (POD), Principal Component Analysis (PCA), or as the Karhunen-Loeve Decomposition (KLD) [3, 27-30]. For purposes of this thesis we will refer to the reduced order model based on the singular value decomposition as the reduced order model or ROM although in some references it may be referred to by one of above mentioned titles.

ROM was first developed about a century ago as a tool for processing statistical data [27, 28]. It is a powerful method of data analysis that is used to obtain a low-dimensional approximate description of a high-dimensional process. ROM is used in signal processing, pattern recognition, control theory, fluid flow, and dynamics [29, 30]. Recently ROM has been demonstrated in heat transfer applications. Ostrowski et al. applied a ROM to an inverse method in order to estimate thermal conductivities and convection heat transfer coefficients [3].

These methods are valuable because they allow one to solve the problem offline for a set of specified values of input parameters using any available solution method. Once there is a sufficiently large library of solutions gathered, the solution to the problem for a separate set of parameters can be found using a basis derived from the library of solutions. This enables near real-time solutions of highly complex problems.

3   Reverse Monte Carlo

3.1   Introduction to Reverse Monte Carlo

Monte Carlo methods solve complicated mathematical problems using statistical probabilities and random numbers (hence, the name Monte Carlo). A traditional or forward Monte Carlo (FMC) method tracks a "bundle" of photons as it randomly moves through a participating medium. A random number generator and predetermined probability density functions, which are based on the properties of the medium, are used to determine how far the bundle will travel before it encounters either a scattering or an absorbing event, the scattering angle, and the amount of energy absorbed. This process is followed until the bundle is either absorbed or leaves the system. Repeating this procedure for a very large number of photon bundles results in a complete picture of the intensity field.

As an example, consider the implementation of a FMC method to model the propagation of the laser beam through a foam layer illustrated in Fig.1-1. One would emit photon bundles and trace the path of each bundle, even though a very small fraction of them would actually make their way to the detector and provide useful information. The vast majority of the photon bundles are absorbed or leave the medium in locations where there is no detector, so they do not provide any useful information. This is clearly an

inefficient approach that will require an enormous number of bundles to obtain a statistically meaningful result [2].

A more desirable method would be to trace the photons that actually arrive at the detector back to their source. This is the fundamental operating principle for RMC methods [8]. The RMC method is based on the principle of reciprocity in radiative transfer developed by Case [31]. One of the most basic reciprocity principles states that

$$I(\mathbf{r}_1, -\hat{\mathbf{s}}_1; \mathbf{r}_2, \hat{\mathbf{s}}_2) = I(\mathbf{r}_2, -\hat{\mathbf{s}}_2; \mathbf{r}_1, \hat{\mathbf{s}}_1) \qquad (3-1)$$

That is, the intensity at location $\mathbf{r}_1$ and in direction $-\hat{\mathbf{s}}_1$ due to the point source at $\mathbf{r}_2$ in the direction $\hat{\mathbf{s}}_2$ equals the intensity at location $\mathbf{r}_2$ in the direction $-\hat{\mathbf{s}}_2$ due to the point source at $\mathbf{r}_1$ in the direction $\hat{\mathbf{s}}_1$. Also implied in reciprocity is time reflection symmetry, i.e., the scattering angle of a photon moving one direction in time is the same as if the photons where following the same path backwards in time. This means that the bundles reaching the detector can just as well be traced backwards in time from the detector to the source.

In the case where the detector is small and located such that it is not subject to direct radiation from the source, the number of photons that would reach the detector compared to the number of photons actually emitted is extremely small. This explains the advantage of using a reverse scheme when the detector is small and the source is large— far fewer bundles are required to obtain a statistically meaningful result. On the other hand, when the source is small and the detector large a forward scheme may be more efficient.

If both the detector and the source are small both methods become inefficient, or even fail [2]. For the case of collimated radiation, RMC can be made more efficient by separating the intensity into a two parts, direct (collimated) and scattered.

$$I(\mathbf{r},\hat{\mathbf{s}},t) = I_d(\mathbf{r},\hat{\mathbf{s}},t) + I_s(\mathbf{r},\hat{\mathbf{s}},t) \qquad (3\text{-}2)$$

Modest demonstrated the efficiency of this method for the case of a steady radiative source passing through an absorbing and scattering medium [2]. For the problem being studied here a pulsed laser source is used. The added time dimension significantly increases the complexity of tracking the photons to the source. Lu and Hsu demonstrated a RMC simulation for a light pulse in both a two-dimensional [25] and three-dimensional [26] participating medium. The algorithm developed for this thesis which is described in detail below is based on the work of Modest and Lu and Hsu.

3.2   Theoretical Development

For this research we are interested in computing the reflectance-time profile at a detector located at a radial distance r_d from the laser source. The reflectance is defined as the radiative flux incident on the detector location divided by the output flux of the flat-beam laser.

$$\frac{q''_d}{q''_{laser}} = \frac{q_d / A_d}{q_{laser} / A_{laser}} \qquad (3\text{-}3)$$

Following the development of Modest [2] the radiative flux at the detector surface that is included by the detector's acceptance angle can be computed from the intensity at the detector.

$$
\begin{aligned}
q''_d &= \int_0^{2\pi} \int_0^{\theta_{max}} I_d(\theta,\phi,t)\cos\theta\sin\theta\, d\theta\, d\phi \\
&= \frac{1}{2}\int_0^{2\pi} \int_{\cos^2(\theta)}^{1} I_d(\theta,\phi,t)\, d(\cos^2\theta)\, d\phi \\
&\cong \pi\left(1-\cos^2\theta_{max}\right)\sum_{n=1}^{N} I_n(\mathbf{r}_d,-\hat{\mathbf{s}}_d,t)
\end{aligned}
\tag{3-4}
$$

It is assumed that the intensity at the detector is diffuse. This approximation introduces certain limitations on this method. In order for the diffuse approximation to be valid the detector either should be located sufficiently far enough away from the source so that the radiation is diffuse, or a small detector acceptance angle should be modeled. The RMC model used in this research uses an acceptance angle of ten degrees. The total intensity at the detector can be calculated as the sum the individual intensities I_n of N bundles in the direction of $-\hat{\mathbf{s}}_d$ as illustrated in Fig. 3-1.
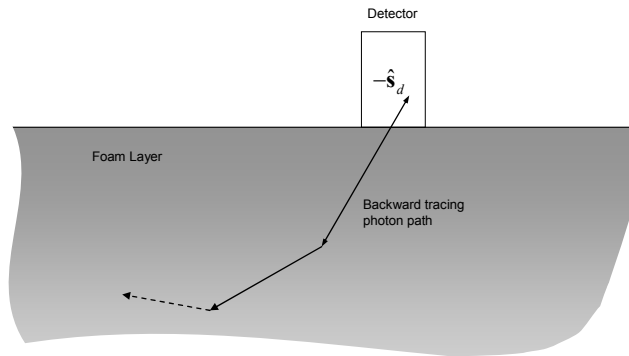
Figure 3-1: Photon trajectory at detector location.

Because the detector is in a location where it is not subject to direct irradiation for the collimated beam there is no direct intensity incident on the detector. The intensity at the detector therefore comes only from scattered radiation and can be found by integrating the source over path lengths where the source and the photon bundles intersect.

$$I_n(\mathbf{r}_d, -\hat{\mathbf{s}}_d, t) = \int_0^l S_1(\mathbf{r}', -\hat{\mathbf{s}}', t) dl' \tag{3-5}$$

where source due to the first scattering of the beam can be expressed as

$$S_1(\mathbf{r}, \hat{\mathbf{s}}, t) = q''_{laser}\, \mu_s \exp(-\mu_s l_c) \frac{\Phi(\mathbf{r}, \hat{\mathbf{s}}_0, \hat{\mathbf{s}})}{4\pi} \tag{3-6}$$

where $\Phi(\mathbf{r}, \hat{\mathbf{s}}_0, \hat{\mathbf{s}})$ is the scattering phase function and $\hat{\mathbf{s}}_0$ is the direction of the collimated beam. The Henyey-Greenstein phase function [32] is used to approximate the scattering phase function with the asymmetry parameter g

$$\Phi(\theta) = \frac{1 - g^2}{(1 + g^2 - 2g\cos\theta)^{3/2}} \tag{3-7}$$

Combining Eqs. (3-5) and (3-6) and noting that $dl' = dz'/\cos\theta$, the intensity at the detector due to a single photon bundle can be expressed as

19

$$I_n(\mathbf{r}_d, -\hat{\mathbf{s}}_d, t) = \frac{\mu_s \, q_{laser}}{4\pi^2 R^2} \sum_j \Phi_j(\mathbf{r}, \hat{\mathbf{s}}_0, \mathbf{s}) \int_{z_{Lj}}^{z_{Uj}} e^{-\mu_s \, z} \frac{dz}{s_{zj}}$$

$$= \frac{q_{laser}}{4\pi^2 R^2} \sum_j \Phi_j(\mathbf{r}, \hat{\mathbf{s}}_0, \mathbf{s}) \frac{e^{-\mu_s \, z_{Lj}} - e^{-\mu_s \, z_{Uj}}}{s_{zj}}$$

(3-8)

where z_U and z_L are the upper and lower z locations where the bundle and pulse cross. The subscript j denotes the individual lengths of the path of a photon between scattering events. A detailed description of how to calculate the crossing locations is given Section 3.3.3. The total intensity at the detector is taken as the statistical average of N photon bundles.

These equations are implemented in an algorithm that is described in detail in the next section.

3.3   Detailed Description of RMC Algorithm

The algorithm developed for this thesis is described in this section. Fig. 3-2 is a flow chart outlining the basic steps in the algorithm. Those steps are described in more detail in the numbered steps given below. Also each step is detailed in the corresponding subsection that follows.
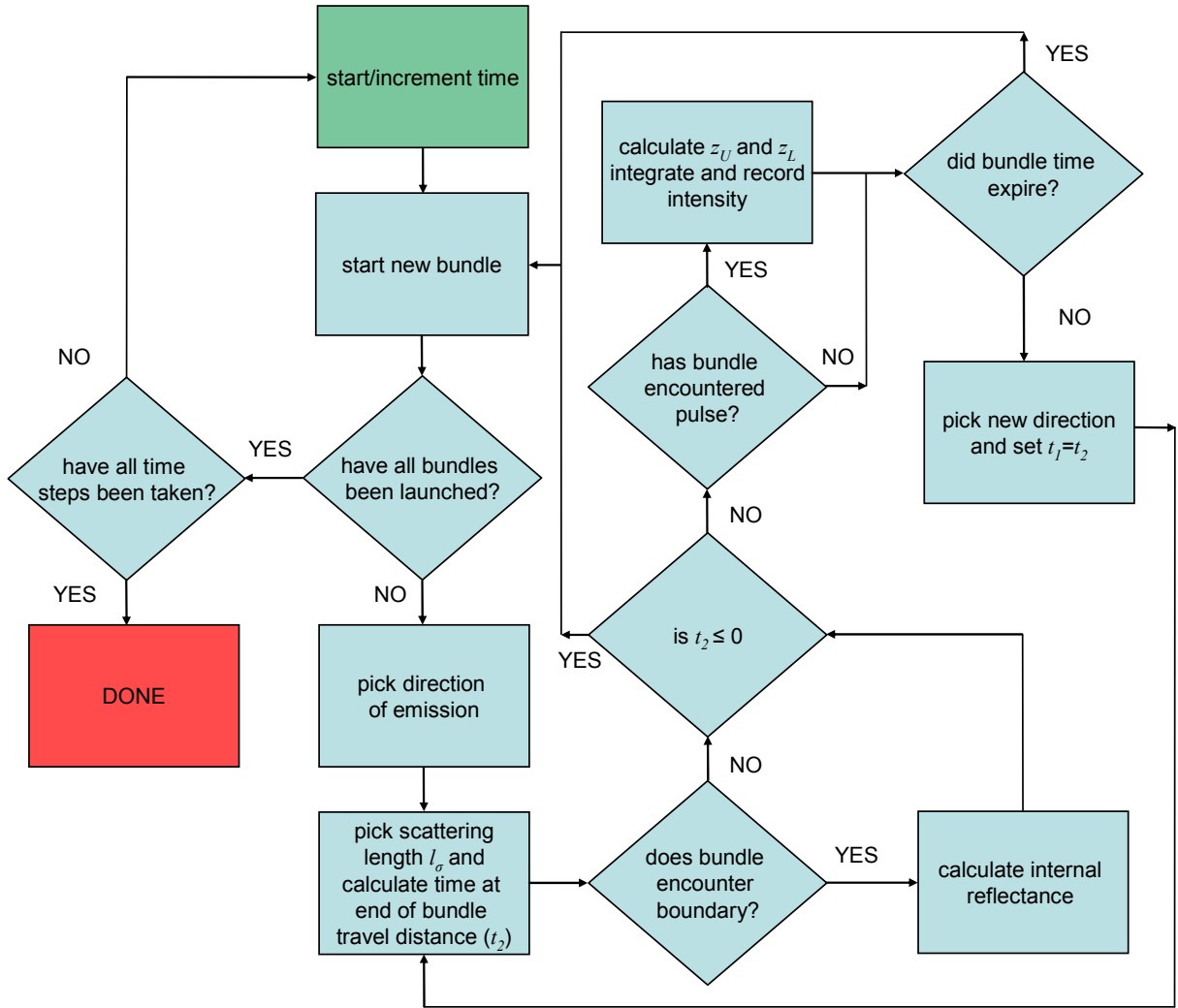
Figure 3-2: Simplified flow chart of RMC algorithm.

1. Select a time t_1 which is the time that the photons reach the detector.

2. Assume that N bundles of photons reach the detector at t_1. Launch one of the N bundles. If all the bundles have been launched for the current time step, go to step one and increment time until the upper limit of the time range is reached.

3. Pick an emission location and direction. The bundles are emitted from some location that is randomly selected from a uniform distribution on the detector. The emission

21

direction (θ, ϕ) is also chosen from random distributions. The method of choosing

emission directions is outlined in Section 3.3.1.

4. Pick the path length to next scattering event, l_σ. The method for choosing scattering

   path length is outlined in Section 3.3.2.

5. Calculate location and time $t_2 (= t_1 - l_\sigma/c)$ of the scattering event. Check to see if $t_2 \le$

   0. If t_2 is less than zero it means that the photon bundle would not have arrived at the

   detector in the time interval selected.

6. Check if the bundle encountered the boundary of the domain. If it has, calculate the

   internal reflectance and decrease the intensity of the bundle accordingly.

7. Check to see if photon has encountered the pulse. If so calculate the upper and lower

   z locations where the bundle and pulse intersected. The procedure for determining z_L

   and z_U is outlined in Section 3.3.3.

8. Pick a new scattering direction based on the procedure in Section 3.3.4. Update time

   $(t_1=t_2)$ and let the end location of the bundle become the starting location for the next

   path length. Go to step 4.

9. Go to step 2 and complete the tally of all N bundles.

3.3.1   Selecting Emission Direction

An emission direction is chosen exactly as it would be in the forward method. As

described by Modest [2] the random number relation for the polar angle of emission θ_e is

$$R_\theta = \frac{\int_{\cos^2 \theta_e}^{1} d\varsigma}{\int_{\cos^2 \theta_{max}}^{1} d\varsigma} = \frac{1-\cos^2 \theta_e}{1-\cos^2 \theta_{max}} = \frac{\sin^2 \theta_e}{\sin^2 \theta_{max}},$$

22

$$\text{or } \theta_e = \sin^{-1}(\sqrt{R_\theta} \sin \theta_{max}) \tag{3-9}$$

The azimuthal angle ϕ_e is chosen uniformly by $\phi_e = 2\pi R_\phi$. R_θ and R_ϕ are random numbers chosen between 0 and 1 and θ_{max} is the detector acceptance angle.

3.3.2 Scattering Length

The scattering length is chosen from random number relations based on the definition of the scattering coefficient given by Modest [8] and is given as

$$l_\sigma = \frac{1}{\mu_s} \ln \frac{1}{R_\sigma} \tag{3-10}$$

Where μ_s is the scattering coefficient and R_σ is a random number uniformly distributed between 0 and 1. It is important to note that travel distance (or path length) and time are not independent. Because photons propagate at the speed of light in the medium the distance traveled is related to time by the constant speed of light, $dl=cdt$. Therefore, knowing the initial location (x_i, y_i, z_i), initial time t_1, the direction of photon propagation $\hat{s} = (s_x, s_y, s_z)$, and now the path length l_σ, the location of the scattering event (x_e, y_e, z_e) and the time of the scattering t_2 event can be calculated as

$$x_e = x_i + l_\sigma s_x$$
$$y_e = y_i + l_\sigma s_y,$$
$$z_e = z_i + l_\sigma s_z,$$

23

and

$$t_2 = t_1 - l_\sigma / c_1 \qquad (3\text{-}11)$$

Noting that t_2 occurs before t_1 or $t_1 > t_2$, c_1 is the speed of light in the medium, and $\hat{\mathbf{s}}$ is a unit direction vector.

3.3.3 Determining Photon-Bundle Interactions

Determining if and where the bundle and pulse intersect is complicated by the fact that the bundle and pulse are moving in both time and space. In the three-dimensional medium the location where the bundle path intersects the cylindrical volume created by the path of the beam must be determined first. There are seven possibilities illustrated in Fig. 3-3.
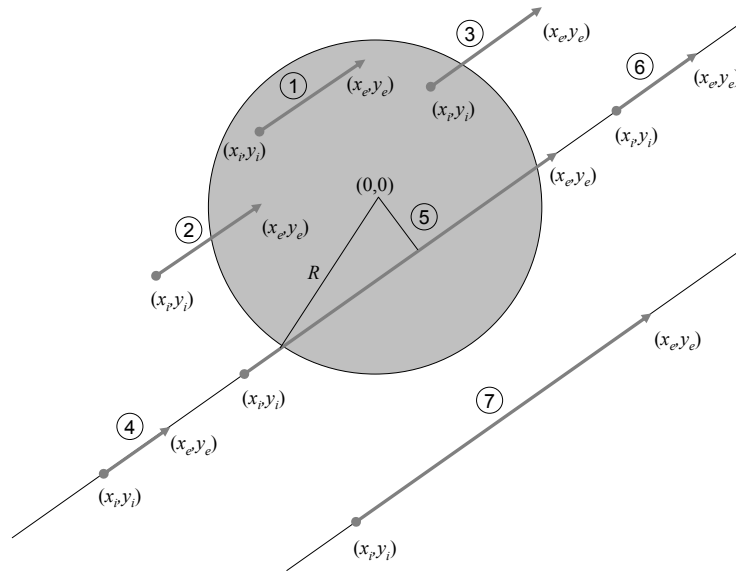
Figure 3-3: Schematic diagram of x-y plane showing 7 different possibilities for bundle-pulse interaction.

The path length to the location where the bundle crosses the beam's cylindrical

path l_c can be found by solving the three equations

$$
\begin{aligned}
x_i + l_c s_x &= x_c, \\
y_i + l_c s_y &= y_c, \\
x_c^2 + y_c^2 &= R^2
\end{aligned}
$$
(3-12)

By squaring and adding the first two equations and substituting the third, the solution can

be found by solving the quadratic equation.

$$
l_{1,2} = \frac{-b \pm \sqrt{d}}{c}
$$
(3-13)

with

$$
\begin{aligned}
a &= x_i^2 + y_i^2 - R^2 \\
b &= x_i s_x + y_i s_y \\
c &= s_x^2 + s_y^2 \\
d &= b^2 - ac
\end{aligned}
$$

The seven possible scenarios depicted in Fig. 3-3 are described as follows:

1. The bundle remains inside the beam's cylindrical path ($r_i < R$ and $r_e < R$).

2. The beginning position in located outside of the beams path, the end position is inside

   ($r_i > R$ and $r_e < R$) and $l_c = l_2$.

3. The beginning position is located inside of the beam's path, the end position is

   outside ($r_i < R$ and $r_e > R$) and $l_c = l_1$.

4. The direction of the bundle path intersects the beam path, but the bundle is scattered

   before reaching beam path ($r_i, r_e > R$). This occurs if $l_2 > l_\sigma$.

25

5. Both endpoints are outside of the beam path ($r_i, r_e > R$), but the bundle intersects the beam path. The path length to the first intersection point is $l_c = l_2$ and to the point where the bundle exits the beam is $l_c = l_1$.

6. The direction of the bundle path intersects the beam path, but the bundle is moving away from the beam path ($r_i, r_e > R$). This is the case if $l_1 < 0$.

7. The path of the bundle completely misses the beam path. This only happens when there are complex roots, i.e. $d < 0$.

Determining where and when the bundle crosses the beam's collimated path is not all. Because the beam is a pulse with temporal width t_p, the interaction location in the z direction must also be found. Lu and Hsu describe a method for finding these locations in [25, 26]. Their method is adopted and described below.

There are numerous possibilities depending on whether the photon is initially inside or outside of the beam path, is inside or outside at the time of the scattering event, is initially within the pulse width, is within the pulse width at the time of the scattering event, leaves or enters the cylinder before, after or during the pulse width, or passes through the entire cylinder. A diagram is helpful to picture some of the possible scenarios.
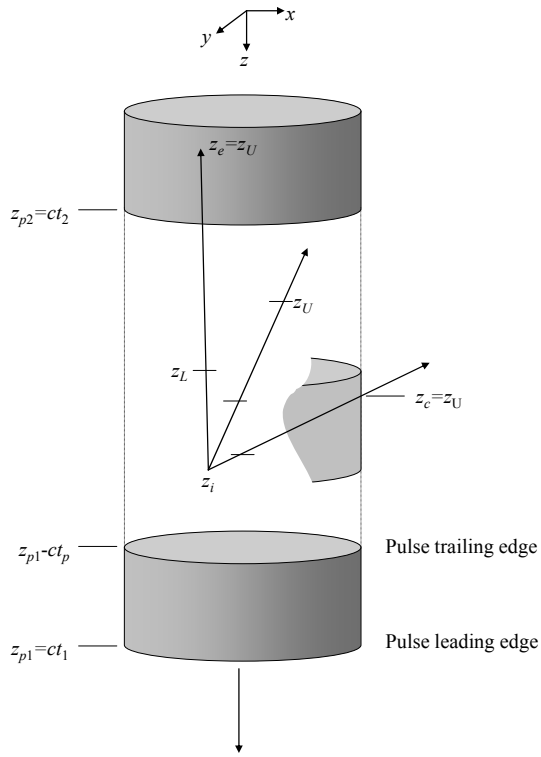
Figure 3-4: Three possible scenarios for photon bundles interacting with pulsed beam. In total there are sixteen different ways the photon bundle and beam can interact.

Depending on where, when, and if the bundle crosses in or out of the beam path, the upper and lower intersection locations can take on any of the following forms.

$$
z_L = \begin{cases} z_i \\ z_c \\ \dfrac{s_z c_1 (t_1 - t_p) + z_i}{1 + s_z} \end{cases}
$$

$$
z_U = \begin{cases} z_e \\ z_c \\ z_{c2} \\ \dfrac{s_z z_{p1} + z_i}{1 + s_z} \end{cases}
$$

(3-14)

Here z_c and z_{c2} are the z locations where the photon bundle crosses into and out of the cylinder. The z locations of the crossing can be found from the knowledge of the distance to the crossing found from Eq. (3-13) and $z_c = z_i + l_{1,2} s_z$. If the photon is moving from outside of the beam path to a location inside the beam path $l_c = l_2$, if it is moving from in to out $l_c = l_1$, and if it moves completely across the beam $l_c = l_2$ and $l_{c2} = l_1$.

Substituting these z locations found in Eq. (3-14) for all the beam-photon interactions into Eq. (3-8) the intensity at the detector location can be found as the statistical average over all the bundles emitted.

$$I_d\left(\mathbf{r}_d, -\hat{\mathbf{s}}_d, t\right) = \frac{1}{N} \sum_{n=1}^{N} I_n\left(\mathbf{r}_d, -\hat{\mathbf{s}}_d, t\right) \tag{3-15}$$

Because it is a statistically random process, there is variation inherent in the solution. Tracking more photons will decrease the amount of statistical noise in the solution. The algorithm used for this research allows the user to specify the maximum standard deviation desired. The program will track a specified set of N photons and calculate the standard deviation using that set. If the standard deviation is greater than the maximum specified, N is doubled until the standard deviation is lower than the required level.

Results using this RMC algorithm are presented in Chapter 5. The actual algorithm coded in C++ is included in appendix B.

4 Reduced Order Modeling Based on SVD

4.1 Introduction to ROM Based on SVD

Most radiative transport problems do not have closed form analytical solutions and the numerical techniques used to solve them are often not efficient enough to allow for real-time solution needed in inverse analysis. Using reduced order modeling (ROM) based on singular value decomposition (SVD) one can solve a complex problem efficiently and accurately. Using information that is obtained off-line using a standard modeling technique (such as MC, RMC, diffusion approximation, etc.), with all the solutions based on a different combination of parameters, a basis can be found for the solution of the problem based on an arbitrary parameter set.

The implementation of reduced order modeling is relatively simple once the parameters governing the phenomenon of interest are identified. Data sets describing the phenomenon at specified values of the governing parameters are obtained by experiment or by analysis [3]. The data sets are arranged in a matrix and an orthonormal basis for the column space of this matrix is obtained from its SVD. The basis vectors are ordered according to the magnitude of their associated characteristic values, and the first few basis vectors contain the majority of the information required to describe the phenomenon. Therefore, a truncated set of the basis vectors can be used to estimate the

phenomenon of interest for an arbitrary set of the governing parameters. The expansion

coefficients, which are functions of the set of governing parameters, are determined by

curve fitting the original data set.

The following paragraphs describe this method in more detail while focusing on

the application of interest in this thesis which is predicting the time-dependent reflectance

profile produced when a pulsed laser beam is normally incident on a non-absorbing,

scattering plane layer. The properties of the layer which form the set of governing

parameters are the scattering coefficient μ_s, the asymmetry parameter g, and the layer

thickness L. The values of these parameters are stored in vector $\mathbf{k}$.

$$\mathbf{k} = \left[\mu_s, g, L\right]^T \tag{4-1}$$

The fundamental concept underlying the reduced order model is that the reflectance

profile for an arbitrary parameter set can be approximated by a linear combination of a

set of orthogonal basis vectors.

$$\mathbf{R}_\mathbf{k} \approx \mathbf{\Phi b(k)} \tag{4-2}$$

In this equation $\mathbf{R}_\mathbf{k}$ is a vector containing a reflectance profile based on the parameters in

the vector $\mathbf{k}$, $\mathbf{\Phi}$ is a matrix of basis vectors, and the vector $\mathbf{b}$ represents the expansion

coefficients, which are functions of the set of governing parameters. The major tasks

required in the development of the reduced order model are the identification of the basis

vectors and the functional relationship between the expansion coefficients and the set of governing parameters.

4.2   Calculation of the Truncated Set of Basis Vectors

In the present study, MC simulations were used to generate reflectance profiles for each possible combination of the following values of the governing parameters: μ_s = (2800, 5600, 8400, 11200, 14000) m^{-1}, g = (0.2, 0.4, 0.6, 0.8) and L = (0.04, 0.08, 0.12, 0.16, 0.2) m. The detector was located a distance of r_d = 5 mm from the center line of the laser beam, and the reflected flux was measured as a function of time for 16 ns with a time resolution of 20 ps. Therefore, this process resulted in m = 100 reflectance profiles, each of which consists of n = 800 time-dependent reflectance measurements. Each reflectance profile forms a column in an n x m matrix $\mathbf{A}$.

Using the SVD, $\mathbf{A}$ can be factored into an n by n orthogonal matrix $\mathbf{U}$, a n by m diagonal matrix Σ and a m by m orthogonal matrix $\mathbf{V}$ as shown in Eq. (4-3) [33].

$$\mathbf{A} = \mathbf{U}_A \Sigma_A \mathbf{V}_A^{\mathrm{T}} \qquad (4\text{-}3)$$

The elements of Σ_A are termed the singular values. The number of nonzero singular values is equal to the rank of $\mathbf{A}$ and is given the symbol r_A. The singular values are represented by σ_{Ai}, i = 1, .., r_A [33].

The first r_A columns of $\mathbf{U}_A$ form an orthogonal basis for the column space of $\mathbf{A}$. However, it is unnecessary to use the entire basis to accurately represent the reflectance profiles, so only the first M columns of $\mathbf{U}_A$ are used to produce the truncated set of basis

vectors, $\mathbf{\Phi}$, required by Eq. (4-2). As shown in Fig. 4-1, the singular values of **A** decrease

rapidly, and based on Fig. 4-1 *M* is set equal to 6 in this study.
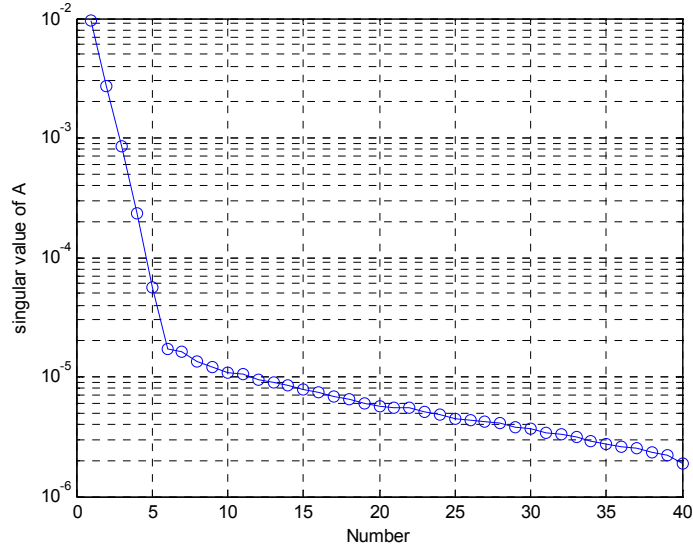
Figure 4-1: Singular values of A.

The ith column of **A**, $\mathbf{a}_i$, represents the time-dependent reflectance profile obtained when

the layer has properties corresponding to the ith set of governing parameters. This

reflectance profile can now be approximated using the truncated set of basis vectors as

shown in Eq. (4-4).

$$\mathbf{a}_i \approx \mathbf{\Phi}\mathbf{b}\left(\mathbf{k}^i\right) \qquad (4\text{-}4)$$

4.3   Calculation of the Expansion Coefficients

Equation (4-4) can be rewritten in matrix form as

$$\mathbf{A} = \mathbf{\Phi B} \qquad (4\text{-}5)$$

where $\mathbf{B}$ is an M by m matrix whose i^{th} column contains the expansion coefficients required to reconstruct the reflectance profile corresponding to the ith parameter set. Because of the orthogonality of $\mathbf{\Phi}$, $\mathbf{B}$ is readily calculated as shown in Eq. (4-6).

$$\mathbf{B} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{A} \qquad (4\text{-}6)$$

Equation (4-5) gives the expansion coefficients for each of the specified parameter sets used to create $\mathbf{A}$. The ability to calculate the expansion coefficients for an arbitrary parameter set, $\mathbf{b(k)}$, is required. Estimates of expansion coefficients may be obtained by interpolation of the results given in Eq. (4-6). This is done by defining a coefficient matrix $\mathbf{C}$ such that

$$\mathbf{B} \equiv \mathbf{CF} \qquad (4\text{-}7)$$

where $\mathbf{F}$ is an m by m matrix of interpolating functions. While any interpolating functions may be used, previous studies have found that inverse multiquadric functions are useful when interpolating data in more than one dimension [34, 35]. The ith column of $\mathbf{F}$ is

calculated by evaluating the multiquadric interpolation function defined in Eq. (4-8) for

each of the parameter sets used to generate **A**.

$$\mathbf{f}_i = \begin{bmatrix} \dfrac{1}{\sqrt{\left(\dfrac{\mu_s^i - \mu_s^1}{\mu_{max}}\right)^2 + \left(\dfrac{g^i - g^1}{g_{max}}\right)^2 + \left(\dfrac{L^i - L^1}{L_{max}}\right)^2 + 1}} \\ \vdots \\ \dfrac{1}{\sqrt{\left(\dfrac{\mu_s^i - \mu_s^j}{\mu_{max}}\right)^2 + \left(\dfrac{g^i - g^j}{g_{max}}\right)^2 + \left(\dfrac{L^i - L^j}{L_{max}}\right)^2 + 1}} \\ \vdots \\ \dfrac{1}{\sqrt{\left(\dfrac{\mu_s^i - \mu_s^m}{\mu_{max}}\right)^2 + \left(\dfrac{g^i - g^m}{g_{max}}\right)^2 + \left(\dfrac{L^i - L^m}{L_{max}}\right)^2 + 1}} \end{bmatrix} \tag{4-8}$$

Since **F** may be a singular matrix, Eq. (4-7) must be solved for the coefficient matrix **C** in

a least squares sense using the Moore-Penrose pseudoinverse, $\mathbf{F}^+$ [33, 36]. Calculation of

the pseudoinverse is also based on the SVD of the interpolation matrix.

$$\mathbf{F} = \mathbf{U}_F \mathbf{\Sigma}_F \mathbf{V}_F^T \tag{4-9}$$

$\mathbf{U}_F$ and $\mathbf{V}_F$ are orthonormal matrices and $\mathbf{\Sigma}_F$ is a diagonal matrix containing the singular

values of **F**, $(\sigma_{Fi}, i = 1, .., r_F)$ the pseudoinverse is found as illustrated in Eq. (4-10).

$$\mathbf{F}\mathbf{V}_F\mathbf{S}_F\mathbf{U}_F^T = \mathbf{U}_F\mathbf{\Sigma}_F\mathbf{V}_F^T\mathbf{V}_F\mathbf{S}_F\mathbf{U}_F^T = \mathbf{I} \tag{4-10}$$

where $\mathbf{S}_F$ is a diagonal matrix that is defined as

$$\mathbf{S}_F = \text{diag}\left\{1/\sigma_{F,i}\right\} \tag{4-11}$$

for $\sigma_{F,i} > 0$. In practice, it is necessary to specify a minimum allowable value for

$\sigma_{F,i}$ and to set the elements of $\mathbf{S}_F$ equal to zero if $\sigma_{F,i}$ is less than this specified minimum

value. The singular values of the interpolation matrix used in this study are shown in Fig.

4-2. The results presented in the following section were obtained with a minimum

allowable singular value of 10^{-4}.
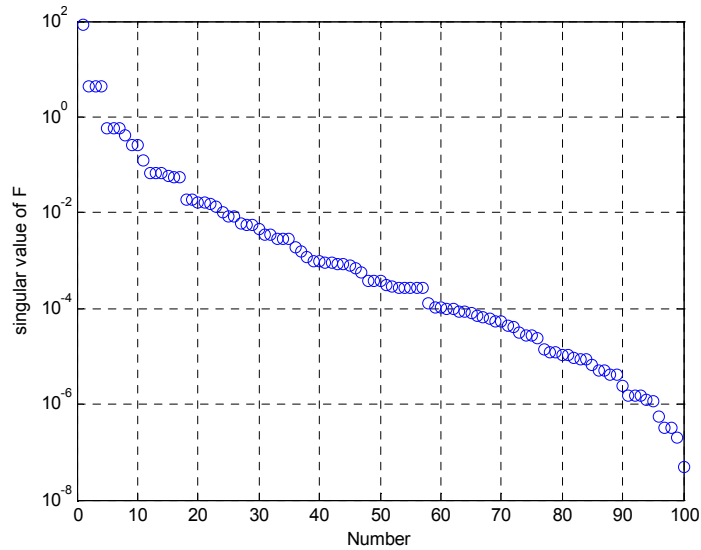
Figure 4-2: Singular values of F.

From Eq. (4-10), it is clear that the pseduoinverse of $\mathbf{F}$ is given by

$$\mathbf{F}^+ = \mathbf{V}_F \mathbf{S}_F \mathbf{U}_F^T \qquad (4\text{-}12)$$

Once the pseudoinverse of $\mathbf{F}$ is known, the coefficient matrix is obtained by post multiplying Eq. (4-7) by $\mathbf{F}^+$.

$$\mathbf{C} = \mathbf{B}\mathbf{F}^+ \qquad (4\text{-}13)$$

It is assumed that this coefficient matrix is valid for any arbitrary parameter set. Therefore, expansion coefficients for any arbitrary parameter set may be obtained by evaluating the interpolation function at $\mathbf{k}$, and post multiplying $\mathbf{C}$ with the resulting vector.

$$\mathbf{b}(\mathbf{k}) = \mathbf{C}\mathbf{f}(\mathbf{k}) \qquad (4\text{-}14)$$

where

$$\mathbf{f}(\mathbf{k}) = \begin{bmatrix} \dfrac{1}{\sqrt{\left(\dfrac{\mu_s - \mu_s^{\,1}}{\mu_{max}}\right)^2 + \left(\dfrac{g - g^1}{g_{max}}\right)^2 + \left(\dfrac{L - L^1}{L_{max}}\right)^2 + 1}} \\ \vdots \\ \dfrac{1}{\sqrt{\left(\dfrac{\mu_s - \mu_s^{\,j}}{\mu_{max}}\right)^2 + \left(\dfrac{g - g^j}{g_{max}}\right)^2 + \left(\dfrac{L - L^j}{L_{max}}\right)^2 + 1}} \\ \vdots \\ \dfrac{1}{\sqrt{\left(\dfrac{\mu_s - \mu_s^{\,m}}{\mu_{max}}\right)^2 + \left(\dfrac{g - g^m}{g_{max}}\right)^2 + \left(\dfrac{L - L^m}{L_{max}}\right)^2 + 1}} \end{bmatrix} \qquad (4\text{-}15)$$

The expansion coefficients given by Eq. (4-14) are then used in Eq. (4-2) to calculate the reflectance profile for any arbitrary parameter set, **k**.

ROM has been applied to the problem at hand. The results for ten test cases with random parameter sets are presented in chapter 5.

5   Results

5.1   Results of Reverse Monte Carlo Simulations

Ten test cases were investigated in order to compare the accuracy and efficiency

of the reverse Monte Carlo algorithm with the forward Monte Carlo predictions. The

parameters for the cases were generated randomly with the asymmetry parameter g

ranging from 0.2 to 0.8, the scattering coefficient μ_s ranging from 2800 to 14000 m^{-1}, and

the layer thickness L ranging from 0.04 to 0.2 m. Table 5-1 shows the parameters used in

each case.

Table 5-1: Parameters used for ten test cases.

Case	*g*	*L* [m]	μ_s [m^{-1}]
1	0.5154	0.1415	11620
2	0.5066	0.1963	13760
3	0.4330	0.0500	7840
4	0.7888	0.0989	12760
5	0.8964	0.0403	3173
6	0.6710	0.1690	3768
7	0.6590	0.1800	13410
8	0.2970	0.0530	4950
9	0.6920	0.1260	9865
10	0.5390	0.1540	9737

The geometry modeled here is that shown in Fig. 1-1. A flat-beam pulsed laser with a radius of 1mm and with a pulse width of 20 ps is normally incident on a foam layer with specified thickness L. A detector with an acceptance angle $\theta_{max} = 10°$ is located at a radius of 5 mm from the center of the beam. The ten cases where all run on an SGI Origin supplied with 64 MIPS processors.

5.1.1 Accuracy of RMC

Results of all ten cases can be found in Appendix A. For the sake of brevity a few of the cases are presented here. Figure 5-1 (a-d) are reflectance-time profiles of purely anisotropically scattering foam layers that is irradiated by a flat-beam pulsed laser. A detector with a ten degree acceptance angle is used for this comparison. The profiles are calculated using both FMC and RMC.

(a)

(b)

(c)

41
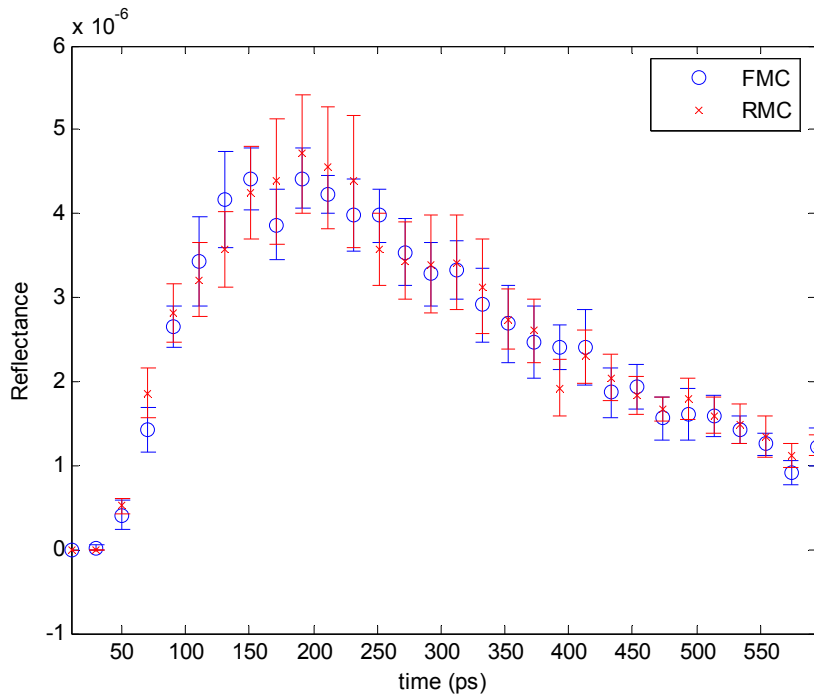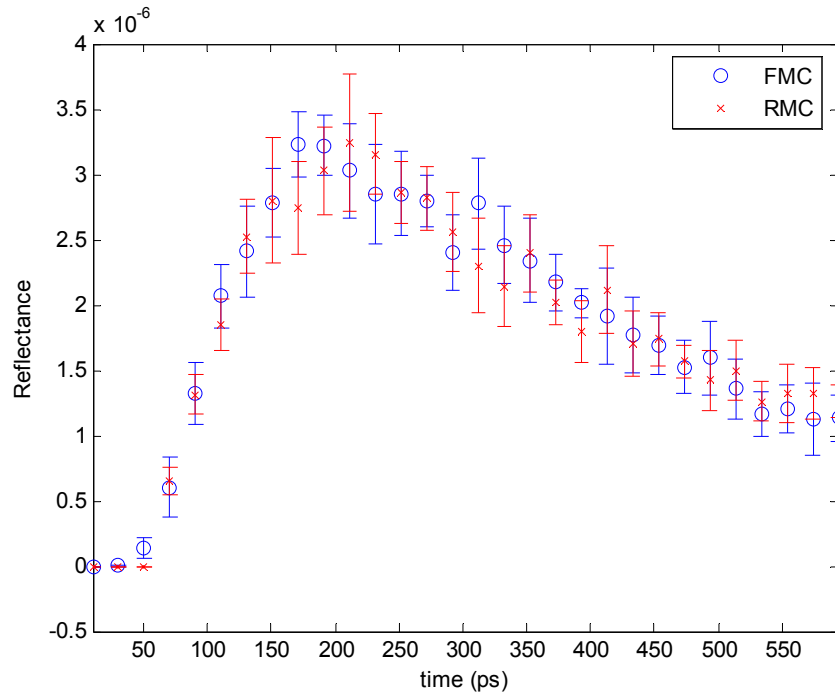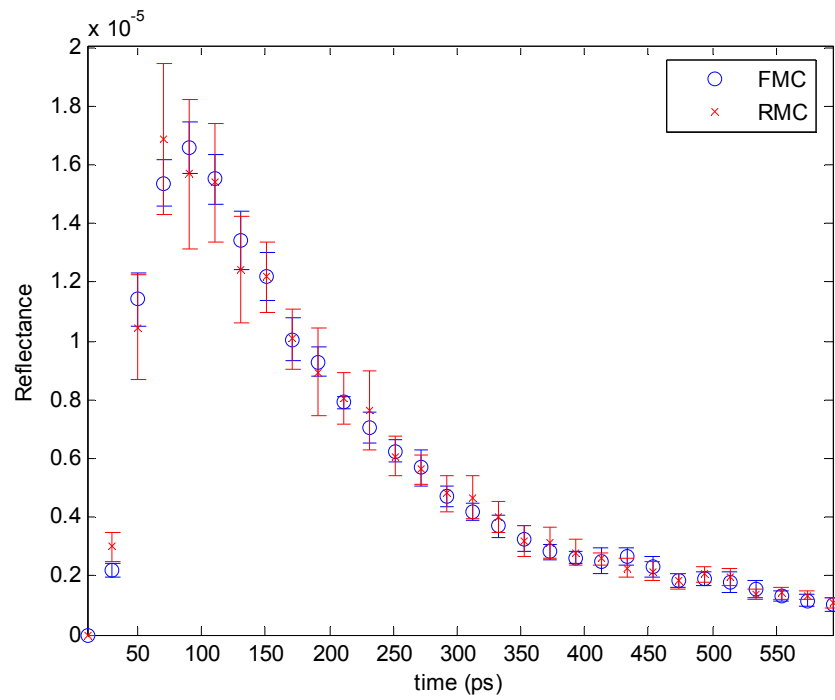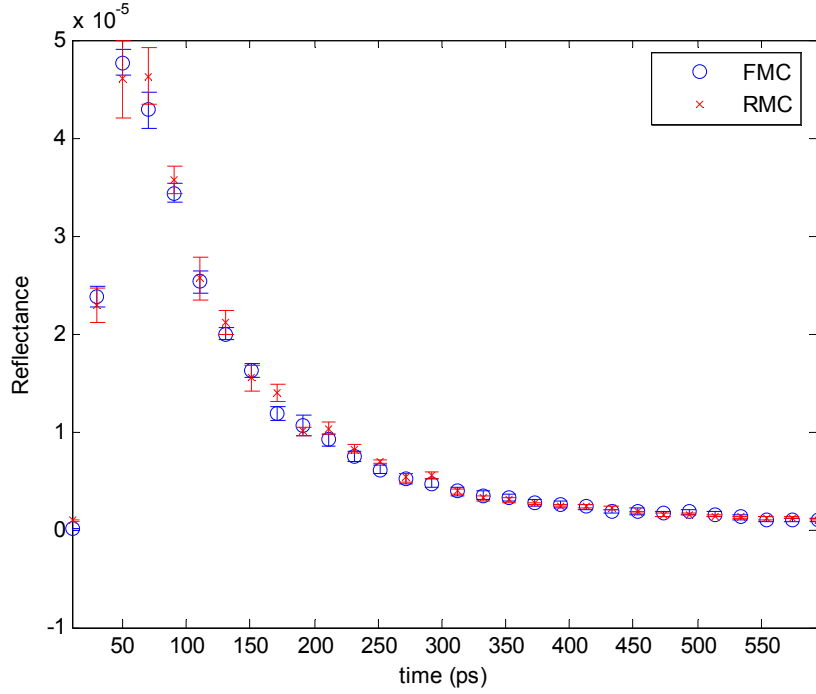
(d)

Figure 5-1: Comparison of reflectance profiles obtained using a reverse Monte Carlo (RMC) with the corresponding FMC simulation for (a) case 1 with parameters $\mu_s = 11{,}620$ m^{-1}, $g = 0.5154$, $L = 0.1415$ m, (b) case 2 with parameters $\mu_s = 13{,}760$ m^{-1}, $g = 0.5066$, $L = 0.1963$ m, (c) case 4 with parameters $\mu_s = 12{,}760$ m^{-1}, $g = 0.7888$, $L = 0.0989$ m and (d) case 6 with parameters $\mu_s = 3{,}768$ m^{-1}, $g = 0.6710$, $L = 0.1690$ m. A flat-beam pulsed laser and a detector with a ten degree acceptance angle are used.

The agreement between the FMC and RMC is good. Listed below in Table 5-2 is the quantification of the difference between the two profiles. The difference is quantified as the difference in the areas under the respective curves normalized by the area under the FMC curve as calculated by Eq. (5-1).

$$\delta_A = \left| \frac{A_{FMC} - A_{RMC}}{A_{FMC}} \right| \qquad (5\text{-}1)$$

Also an *R*-squared value for the relationship is calculated using standard statistical methods.

Table 5-2: Quantification of the difference between FMC and RMC using the relative difference in the area under the curves and an R-squared value.

Case	g	L [m]	μ_s [m^{-1}]	δ_A	R-squared
1	0.5154	0.1415	11620	0.0806	0.9577
2	0.5066	0.1963	13760	0.0780	0.9491
3	0.4330	0.0500	7840	0.0767	0.9618
4	0.7888	0.0989	12760	0.0556	0.9887
5	0.8964	0.0403	3173	0.0659	0.9947
6	0.6710	0.1690	3768	0.0645	0.9935
7	0.6590	0.1800	13410	0.0771	0.9702
8	0.2970	0.0530	4950	0.0686	0.9755
9	0.6920	0.1260	9865	0.0690	0.9776
10	0.5390	0.1540	9737	0.0708	0.9737

5.1.2   Efficiency of RMC

The efficiency of the RMC algorithm can be measured by the number of photon bundles needed in order to obtain a statistically accurate prediction. Since it takes essentially the same amount of CPU time to track a photon backward as it does to track it forward, we can compare the efficiency of both methods by looking at the number of bundles required to achieve the same level of accuracy. In the FMC cases there were 10 sets of 10^6 bundles for a total of 10^7 bundles. To match this level of variance obtained by the FMC with 10^7 bundles the RMC averaged anywhere from 10^6 up to 2 x 10^7 depending on the properties simulated as well as the time step of interest. The actual run time for the two methods would confirm this as both methods required approximately the same amount of computation time.

5.2   Results of Reduced Order Modeling

The results of the ROM are presented in this section. The same ten random test cases from Table 5-1 are used to establish a comparison between the ROM and FMC. For these cases a Gaussian beam laser is modeled and the detector has a ninety degree acceptance angle. It should be noted that this is a different setup than that used in the RMC simulations and therefore the results for the ROM are not comparable to those given for RMC. Recall that the RMC results are for a flat-beam laser with a ten degree acceptance angle. A Gaussian beam laser with a large acceptance angle is used to generate the library for the ROM solution in order to increase the speed at which the solutions were obtained.

5.2.1   Accuracy of ROM

In order to generate the ROM basis vectors a library of reflectance profiles was generated using FMC simulations. Profiles where obtained for each possible combination of the following values of the governing parameters: μ_s = (2800, 5600, 8400, 11200, 14000) m^{-1}, g = (0.2, 0.4, 0.6, 0.8) and L = (0.04, 0.08, 0.12, 0.16, 0.2) m. For the purposes of the ROM basis vector library 10^7 photons where used in the FMC simulations in order to obtain clean profiles. The matrix **A** contains m = 100 column vectors of profiles for each combination of parameters, each of which consists of n = 800 time-dependent reflectance measurements. Using this library, ROM was run for the ten test cases listed in Table 5-1. A few of the profiles obtained from the ROM are compared to FMC simulations below in Fig. 5-2 (a-d).
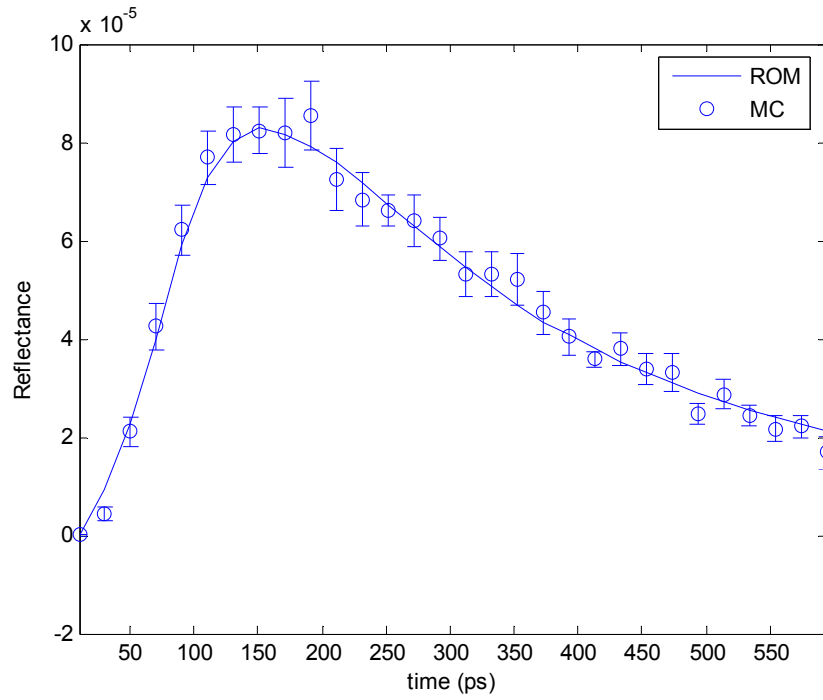
(a)
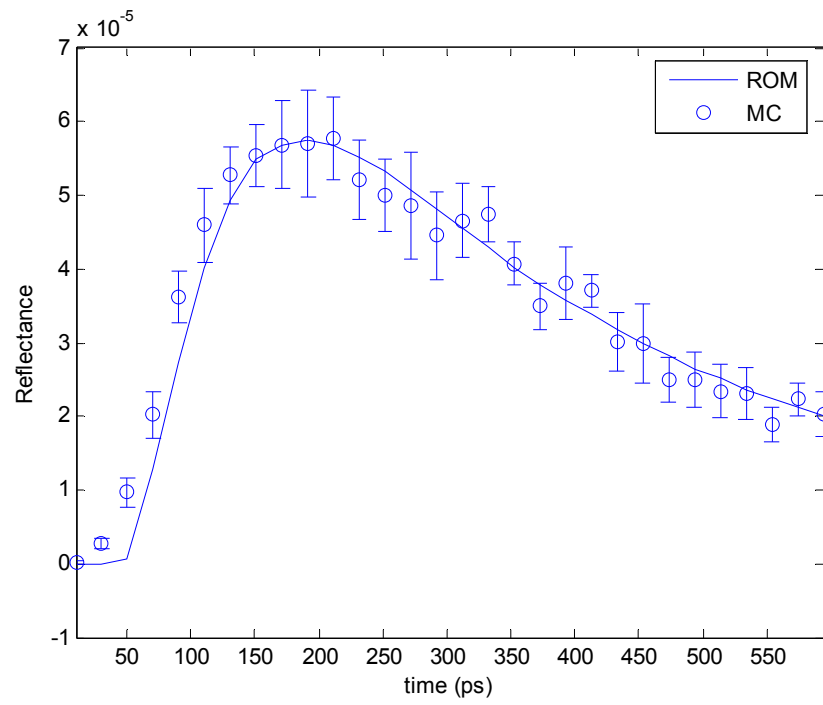
(b)
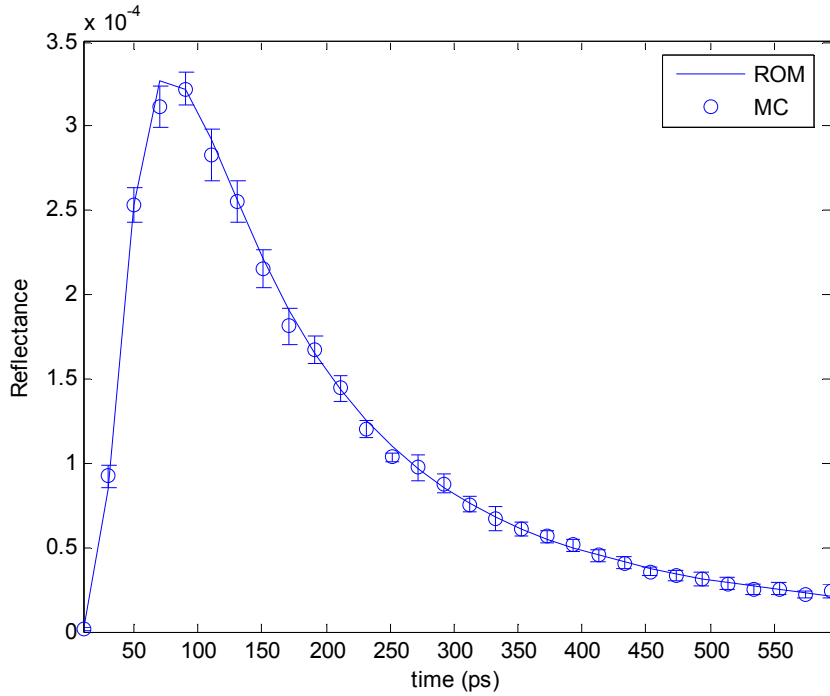
(c)
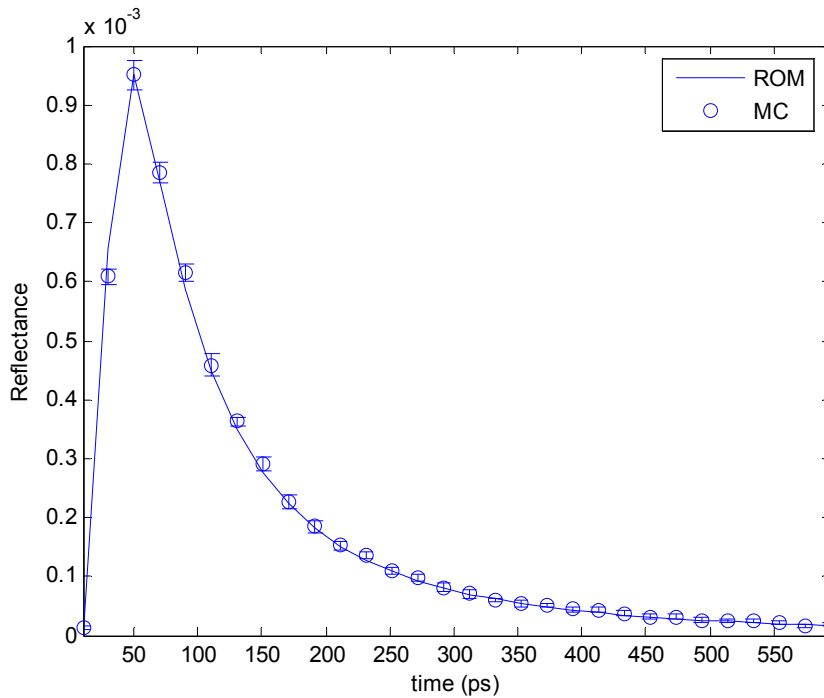
(d)

Figure 5-2: Comparison of reflectance profiles obtained using a reduced order model (ROM) with the corresponding FMC simulation for (a) case 1 with parameters $\mu_s = 11{,}620$ m^{-1}, $g = 0.5154$, $L = 0.1415$ m, (b) case 2 with parameters $\mu_s = 13{,}760$ m^{-1}, $g = 0.5066$, $L = 0.1963$ m, (c) case 4 with parameters $\mu_s = 12{,}760$ m^{-1}, $g = 0.7888$, $L = 0.0989$ m and (d) case 6 with parameters $\mu_s = 3{,}768$ m^{-1}, $g = 0.6710$, $L = 0.1690$ m. A Gaussian beam laser and a detector with a ninety degree acceptance angle are used.

46

Similar to Table 5-2, Table 5-3 contains the results for the comparison of ROM with

FMC.

Table 5-3: Quantification of the difference between FMC and ROM using the relative difference in the area under the curves and an R-squared value.

Case	g	L [m]	μ_s [m^{-1}]	δ_A	R-squared
1	0.5154	0.1415	11620	0.0497	0.9834
2	0.5066	0.1963	13760	0.0754	0.9500
3	0.4330	0.0500	7840	0.0620	0.9685
4	0.7888	0.0989	12760	0.0250	0.9976
5	0.8964	0.0403	3173	0.2860	0.9246
6	0.6710	0.1690	3768	0.0319	0.9976
7	0.6590	0.1800	13410	0.0698	0.9651
8	0.2970	0.0530	4950	0.0523	0.9844
9	0.6920	0.1260	9865	0.0587	0.9763
10	0.5390	0.1540	9737	0.0520	0.9775

5.2.2 Efficiency of ROM

Once the library of solutions is generated the reduced order model based on SVD

is computationally extremely fast. Also, the basis vectors and constant coefficients need

by ROM can be computed offline leaving a few simple matrix multiplications as the only

calculation left in order to obtain a solution. ROM can be used to find a solution in what

is essentially real-time.

6 Discussion of Results

6.1 Discussion of RMC Results

It can be seen from the results of the RMC simulation that the agreement with the FMC simulation is within statistical error. With this formulation good results can only be obtained if the detector acceptance angle is small enough that the approximation introduced in Eq. (3-4) (that the intensity at the detector is diffuse) is valid or if the detector is far enough away for the source that the radiation can be modeled as diffuse. This puts some limitations on this formulation of the RMC.

The reverse Monte Carlo scheme is most efficient for cases where there is a small detector and a large source. The situation modeled here is a small detector and a small source. The transient nature of the source makes it even less efficient. In order for the photon paths to obtain energy they must encounter the moving pulse. Because the pulse width is relatively small, many photons must be used in order to obtain a meaningful result.

For the results presented here the allowable standard deviation for the RMC was 8%. The average standard deviation was about 6%. To achieve that level of accuracy 10^7 photons where required by the FMC simulations on average for all ten cases. Case ten

was the fastest case, requiring 50 hours to complete 30 data points. Case four was the slowest, requiring about 90 hours on the supercomputer.

The points that are around the peak of the profile all took less photons and were calculated faster than the other points. These times are when the signal-to-noise ratio is the highest, therefore fewer photons are needed to obtain a statistically meaningful result. Conversely, the points farther away from the peak take more time to calculate. This is because the signal-to-noise ratio gets lower as time passes. Also the photons are allowed to travel longer, requiring more computations.

The RMC scheme presented here has some benefits compared to the FMC. Although it may not be much faster computationally, it has the advantage of being able to calculate discrete time intervals. That is, if one is interested in information at one specific time, or over a specific time interval, the RMC scheme allows one to calculate just that information. In FMC the entire profile must be calculated. If information around the peak or information from early time steps is needed, RMC can calculate that relatively quickly.

It is noted that the simulation is faster for smaller scattering coefficients and larger asymmetry parameters. If the medium is highly scattering, as characterized by the scattering coefficient, the photon bundles must be traced through many scattering events—requiring many more calculations than would be required in the case of lower a scattering coefficient. Also, the solution is not very sensitive to the layer thickness for the range that is used in this study. The glass-foam interface is rarely encountered by the photons that are traveling at the times investigated.

6.2   Discussion of ROM Results

The results of the study of ROM show that it can be used as a powerful tool for accurately and efficiently modeling a parameterized process. It has been show that ROM can be used for any parameterized solution of equations if a solution method is available to build the library. Once the library is in place, a solution based on different values of the governing parameters can be found in real-time. The error reported in Table 5-2 can be misleading. The FMC data that the ROM is being compared to has significant statistical variation due to the random nature of the solution technique. For the error to be below 10% is sufficient to say that the solution is accurate. The figures with the error bars show that the ROM model predicts the solution within statistical uncertainties.

The only downside to ROM is the expensive upfront cost of creating the library of solution. For the results presented in this research a library of 100 solutions is used, based on the combinations of the three parameters—two with five values and one with four values. The values used are evenly distributed over the range of parameters. If the number of significant parameters is increased the number of library solutions would increase with the number of required solutions. For example, if just one more parameter is added (e.g., absorption coefficient) with five values, then the number of required solutions would go from 100 up to five times 100, or 500 required solutions. Often the original solutions can be very expensive to obtain. The solutions used here come from the FMC simulation, the results of which require 20 to 200 hours on the supercomputers.

We can see that it is important to understand the range of values that solutions will be needed for so that the library can be built to include those values. Extrapolation outside of the library range is ill advised as can be seen in case 5. The asymmetry

51

parameter used in that case was 0.8964. The upper limit used to create the library is 0.800. It is obvious from the high level of error in this case that extrapolation outside of the range of parameters gives poor results. Also, as can be seen in case 2, the error is increased as the value of the parameters reach the bound. Case 2 has a scattering coefficient of 13,760 m^{-1} which is near the bound of 14,000 m^{-1}.

6.3   Conclusions and Recommendations

It has been shown that a RMC simulation can be formulated for transient radiative transport in a purely scattering foam layer with good agreement to the FMC simulation. Although the RMC did not prove to give marked improvement in computational efficiency over the FMC for the situation studied in this work, it does have advantages if results over a specific time interval are needed. If information at early times is needed or over only a small range of times, the RMC algorithm is more efficient. If the entire time domain is required the FMC may be more efficient.

With the development of parallel computing resources, it is anticipated that a RMC scheme could be coded to trace photons simultaneously on multiple processors. Such a parallel processing scheme could prove to dramatically increase the speed of a RMC solution. A skilled programmer could optimize the algorithm to increase the efficiency.

The results of the ROM are very promising. It has been shown that the ROM developed here is a powerful tool for real-time solutions of complex problems. It is both accurate and efficient. The ROM could be implemented in an inverse scheme to evaluate material properties.

52

References

[1] Salisbury, S. M., "A method for characterizing the properties of industrial foams," Masters Thesis, Brigham Young University, (2005).

[2] Modest, M. F., "Backward Monte Carlo Simulations in Radiative Heat Transfer," *Journal of Heat Transfer,* **125**, pp. 57-62 (2003).

[3]Ostrowski, Z., Bialecki, R. A., Kassab, A. J., "Estimation of constant thermal conductivity by use of Proper Orthogonal Decomposition," *Computational Mechanics*, **37**, pp. 52-59, (2005).

[4] Weaire, D. and Hutzler, S., *The Physics of Foams,* Clarendon Press, Oxford (1999).

[5] Fedorov, A. G. and Viskanta, R., "Radiation Characteristics of Glass Foams," *Journal of the American Ceramic Society,* **83** (11), pp. 2769-2776 (2000).

[6] Trier, W., "Wärmeübergang zwischen Flamme und Glasbad in Glasschmelxwannenöfen," *Glastech. Ber.,* **36** (3), pp. 73-86 (1963).

[7] Choudhary, M. K., "Recent advances in mathematical modeling of flow and heat transfer phenomena in glass furnaces," *Journal of the American Ceramic Society,* **85** (5), pp. 1030-1036. (2002).

[8] Modest, M. F., *Radiative Heat Transfer,* Academic Press, New York (1993).

[9] Schuster, A., "Radiation through a foggy atmosphere," *Astrophysical Journal*, **21**, pp. 1-22, (1905).

[10] Schwarzschild, K., "Über das Gleichgewicht der Sonnenatmosphären (Equilibrium of the sun's atmosphere)," *Akad. Wiss. Göttingen, Math.-Phys. Kl. Nachr.*, **195**, pp.41-53, (1906).

[11] Sakami, M., Mitra, K., and Hsu, P.-f., "Transient Radiative Transfer in Anisotropically Scattering Media Using Monotonicity-Preserving Schemes," *Proceedings, the ASME 2000 Int. Mechanical Engineering Congress & Exposition*, Orlando, FL (2000).

[12] Sakami, M., Mitra, K., and Hsu, P.-f., "Analysis of Light-Pulse Transport Through Two-Dimensional Scattering and Absorbing Media," *J. Quant. Spectrosc. Radiat. Transf.*, **73**(2–5), pp. 169–179 (2002).

[13] Briggs, L.L, Miller, W. F. Jr., and Lewis, E. E., "Ray-effect mitigation in discrete ordinate-like angular finite element approximation in neutron transport," *Nuclear Science and Engineering*, **57**, pp.205-217, (1975).

[14] Raithby, G. D., "Discussion of the finite-volume method for radiation, and its application using 3D unstructured meshs," *Numerical Heat Transfer—Part B: Fundamentals*, **35**(4), pp.389-405, (1999).

[15] Lu, X., Hsu, P.-f., and Chai, J. C., "Transient Radiative Transfer of Light Pulse Propagation in Three-Dimensional Scattering Media With Finite Volume Method and Integral Equation Model," ASME Paper No. HT2003-47455, (2003).

[16] Tan, Z.-M., and Hsu, P.-f., "An Integral Formulation of Transient Radiative Transfer," *ASME J. Heat Transfer*, **123**(3), pp. 466–475, (2001).

[17] Lu, X., and Hsu, P. F., "Parallel Computing of an Integral Formulation of Transient Radiation Transport," *Journal of Thermophysics and Heat Transfer*, **17**(4), pp. 425-433, (2003).

[18] Gordon, H. R., "Ship Perturbation of Irradiance Measurements at Sea: 1 Monte Carlo Simulations," *Applied Optics*, **24**, pp. 4172-4182, (1985).

[19] Collins, D. G., Blattner, W. G., Wells, M. B., and Horak, H. G., "Backward Monte Carlo Calculations of the Polarization Characteristics of the Radiation Emerging From Spherical-Shell Atmospheres," *Applied Optics*, **11**, pp. 2684-2696, (1972).

[20] Adams, C. N., and Kattawar, G. W., "Radiative Transfer in Spherical Shell Atmospheres—I. Rayleigh Scattering," *Icarus*, **35**, pp.139-151, (1978).

[21] Nishita, T., Miyawaki, Y., and Makamae, E., "A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources," *Computer Graphics*, **21**, pp. 303-310, (1987).

[22] Sabella, P., "A Rendering Algorithm for Visualizing 3D Scalar Fields," *Computer Graphics*, **22**, pp. 51-58, (1988).

[23] Edwards, D. K., "Numerical Methods in Radiation Heat Transfer," in *Proc. Second National Symposium on Numerical Properties and Methodologies in Heat Transfer*, Shih, T.M., ed., Hemisphere, pp. 479-496, (1983).

[24] Walters, D. V., and Buckius, R. O., "Rigorous Development For Radiation Heat Transfer in nonhomogeneous Absorbing, Emitting And Scattering Media," *Int. J. Heat Mass Transf.*, **35**, pp. 3323–3333, (1992).

[25] Lu, X., and Hsu, P. F., "Reverse Monte Carlo Method for Transient Radiative Transfer in Participating Media," *Journal of Heat Transfer*, **126**, pp. 621-627, (2004).

[26] Lu, X., and Hsu, P. F., "Reverse Monte Carlo Simulations of Light Pulse Propagation Within Three-Dimensional Media," *Journal of Thermophysics and Heat Transfer*, **19**(3), pp. 353-359, (2005).

[27] Pearson, K., "On lines planes of closest fit to system of points in space," *Lond Edinb Dublin Philos Magazine J Sci*, **2**, pp. 559-572, (1901).

[28] Hoteling H., "Analysis of complex statistical variables into principal components," *Journal of Educational Psychology*, **24**, pp. 417-441, (1983).

[29] Atwell, J. A., and King, B. B., "Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations," *Mathematical and Computer Modeling*, **33**, pp. 1-19, (2001).

[30] Berkooz, G., Holmes, P., Lumley, J. L., "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual Review of Fluid Mechanics*, **25**(5), pp. 539-575, (1993).

[31] Case, K. M., "Transfer Problems and the Reciprocity Principle," *Rev. Mod. Phys.*, **29**(4), pp.651-663, (1957).

[32] Jacques, S. L., and Wang, L., "Monte Carlo Modeling of Light Transport in Tissues," *Optical Response of Laser-Irradiated Tissue*, A. J. Welch and M. J. C. van Gemert, editors, Plenum Press, New York, pp. 73-100, (1995).

[33] Strang, Gilbert, *Linear Algebra and its Application*, Thomson, Brooks/Cole, Belmont, CA, (2006).

[34] Hardy, R.L., "Multiquadric equations of topography and other irregular surfaces," *Journal of Geophysics Research*, **76**, pp. 1905-1915, (1971).

[35] Hardy, R.L., "Theory and applications of the multiquadric-biharmonic method: 20 years of discovery 1968-1988," *Computational Math Applications*, **19**, pp. 163-208, (1990).

[36] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., *Numerical Recipes in FORTRAN: the art of scientific computing,* Cambridge University Press, Cambridge, (1992).

Appendix A.   Comparison of RMC with FMC

The contents of this appendix include comparisons between reverse Monte Carlo and forward Monte Carlo simulations for all ten test cases that were run.
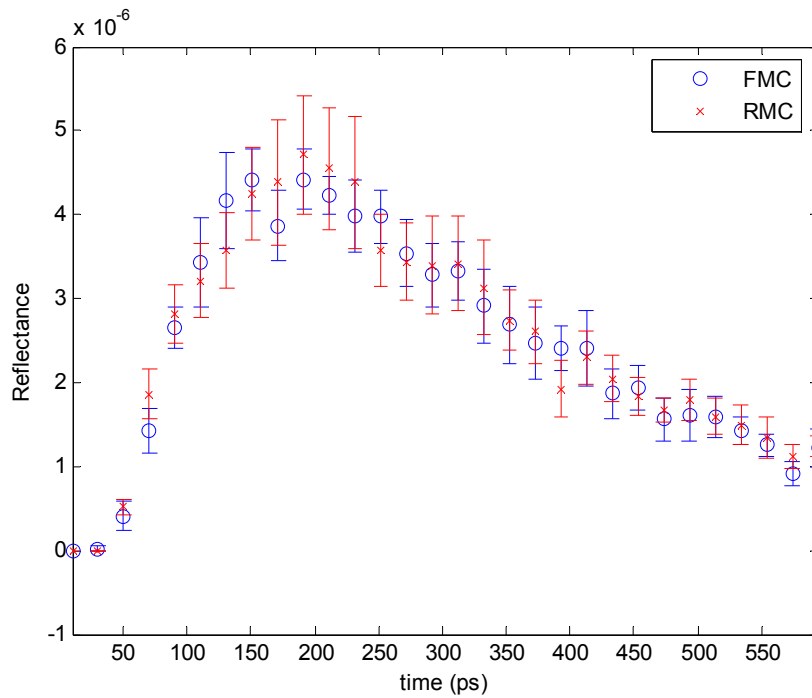
Figure A-1: Comparison between RMC and FMC for values of case 1 (μ_s = 11,620 m^{-1}, g = 0.5154, L = 0.1415 m).
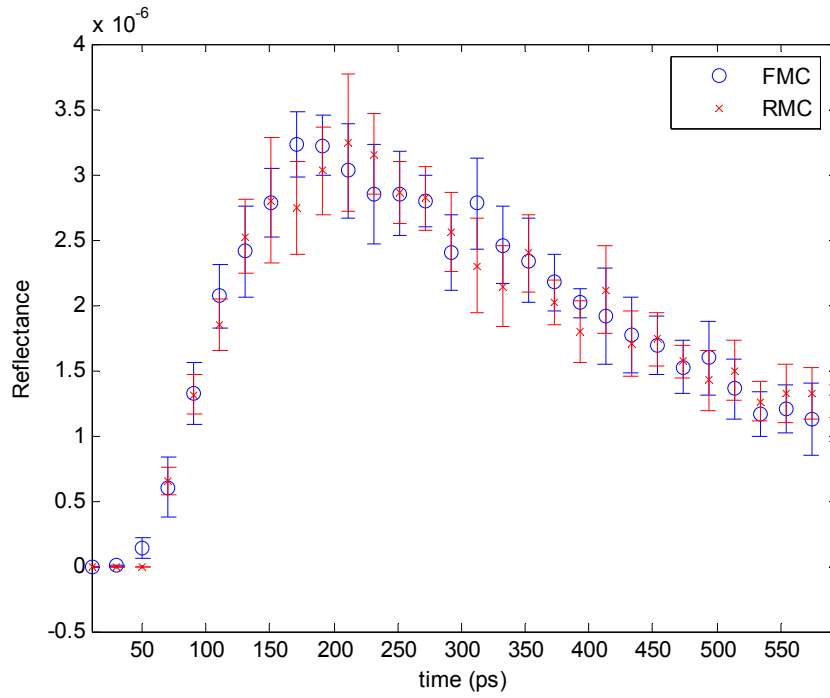
Figure A-2: Comparison between RMC and FMC for values of case 2 ($\mu_s = 13{,}760$ m^{-1}, $g = 0.5066$, $L = 0.1963$ m).

Figure A-3: Comparison between RMC and FMC for values of case 3 ($\mu_s = 7{,}840$ m^{-1}, $g = 0.4330$, $L = 0.0500$ m).

58

Figure A-4: Comparison between RMC and FMC for values of case 4 (μ_s = 12,760 m^{-1}, g = 0.7888, L = 0.0989 m).

Figure A-5: Comparison between RMC and FMC for values of case 5 (μ_s = 3,173 m^{-1}, g = 0.8964, L = 0.0403 m).

Figure A-6: Comparison between RMC and FMC for values of case 6 ($\mu_s = 3{,}768$ m^{-1}, $g = 0.6710$, $L = 0.1690$ m).

Figure A-7: Comparison between RMC and FMC for values of case 7 ($\mu_s = 13{,}410$ m^{-1}, $g = 0.6590$, $L = 0.1800$ m).

Figure A-8: Comparison between RMC and FMC for values of case 8 (μ_s = 4,950 m^{-1}, g = 0.2970, L = 0.0530 m).

Figure A-9: Comparison between RMC and FMC for values of case 9 (μ_s = 9,865 m^{-1}, g = 0.6920, L = 0.1260 m).

Figure A-10: Comparison between RMC and FMC for values of case 10 (μ_s = 9,737 m^{-1}, g = 0.5390, L = 0.1540 m).

Appendix B.   Comparison of ROM with FMC

The contents of this appendix include comparisons between reduced order modeling based on singular value decomposition and forward Monte Carlo simulations for all ten test cases that were run.
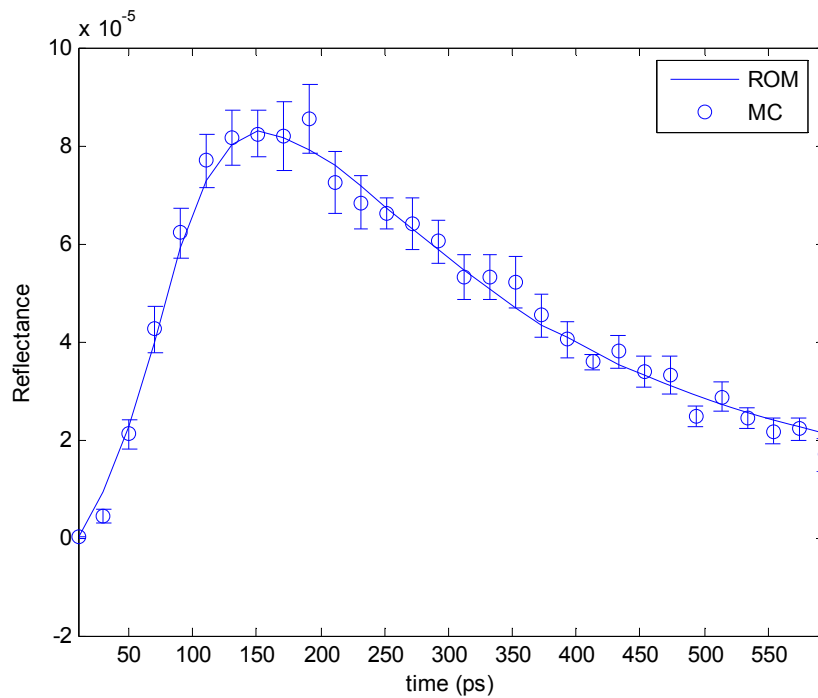
Figure B-1: Comparison between ROM and FMC for values of case 1 (μ_s = 11,620 m^{-1}, g = 0.5154, L = 0.1415 m).
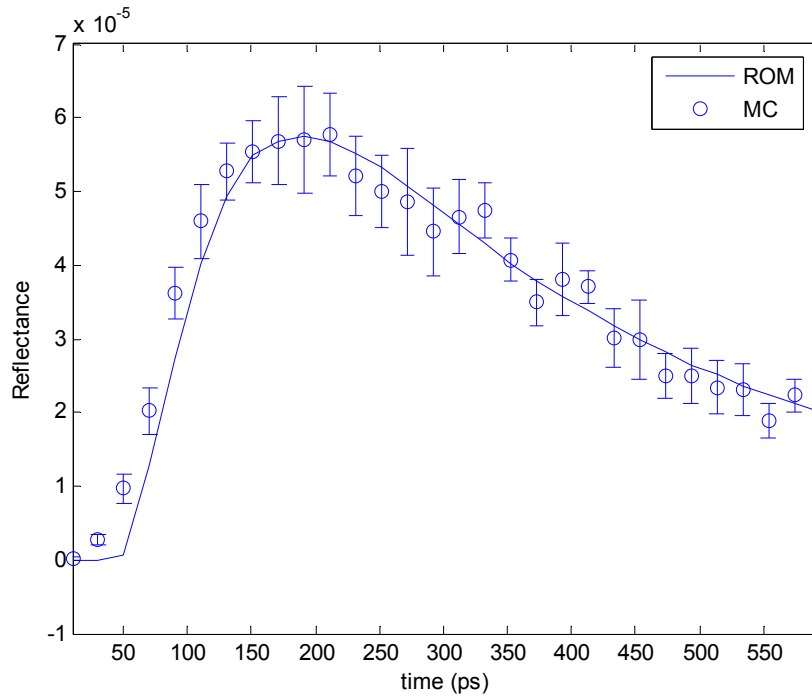
Figure B-2: Comparison between ROM and FMC for values of case 2 (μ_s = 13,760 m^{-1}, g = 0.5066, L = 0.1963 m).

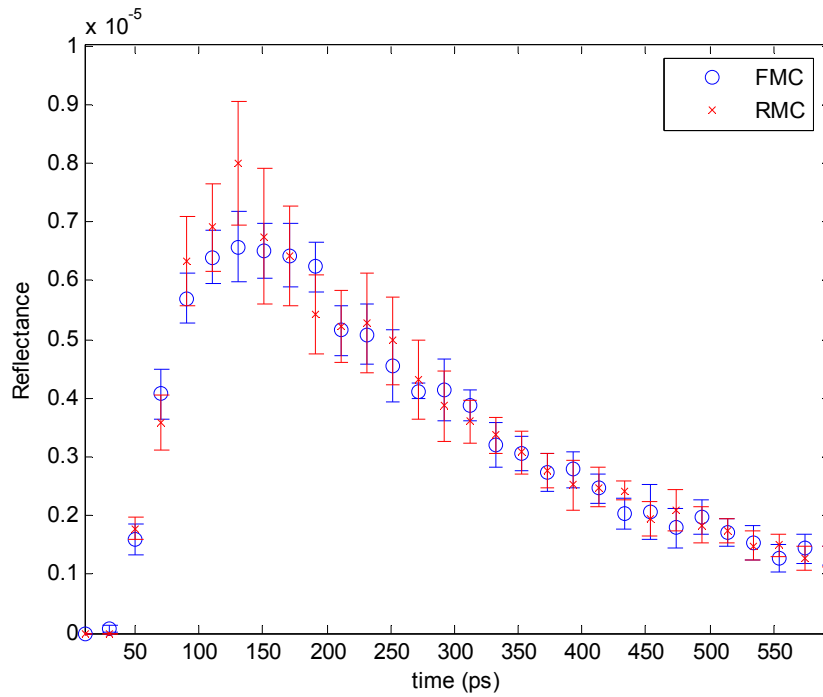**Figure B-3: Comparison between ROM and FMC for values of case 3 (μ_s = 7,840 m^{-1}, g = 0.4330, L = 0.0500 m).**

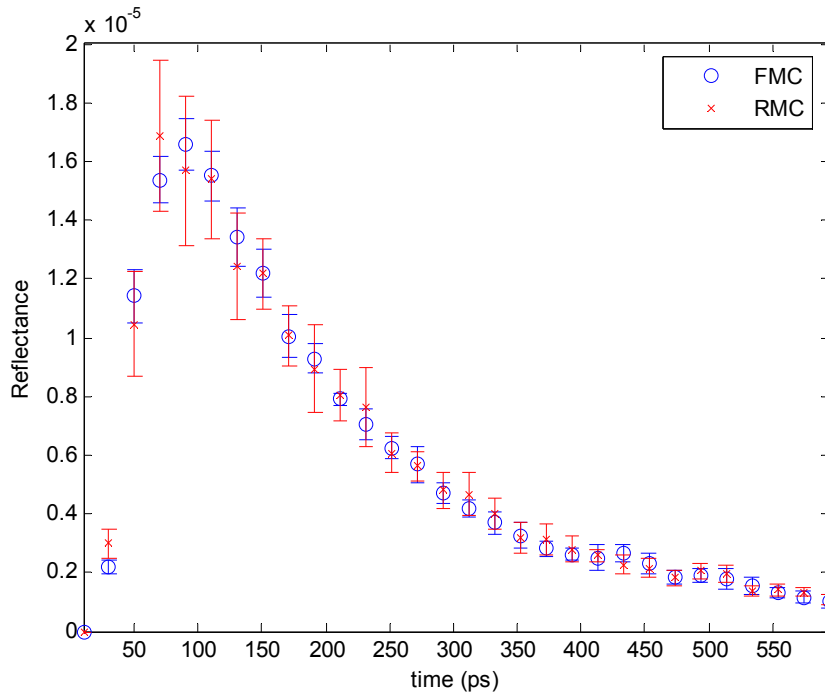**Figure B-4: Comparison between ROM and FMC for values of case 4 (μ_s = 12,760 m^{-1}, g = 0.7888, L = 0.0989 m).**
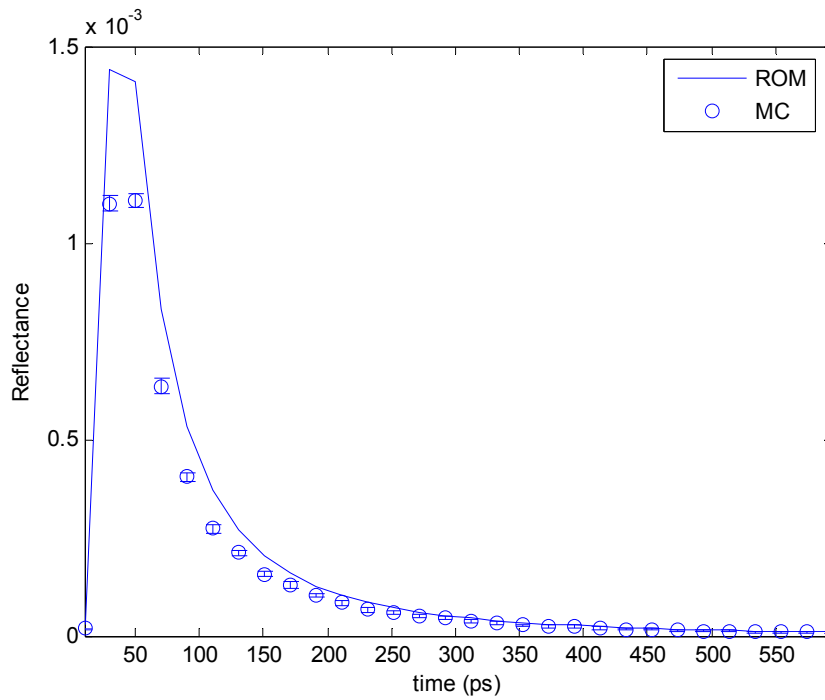
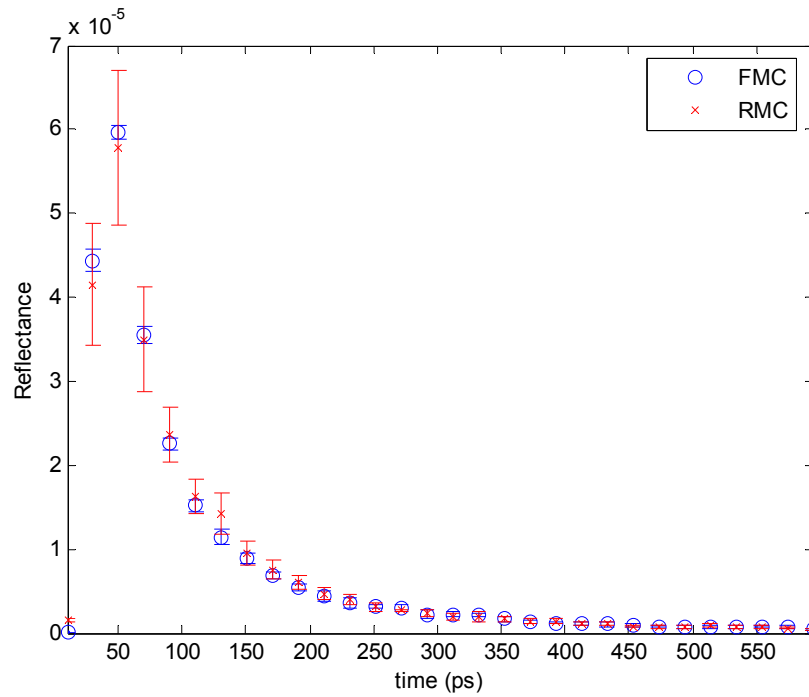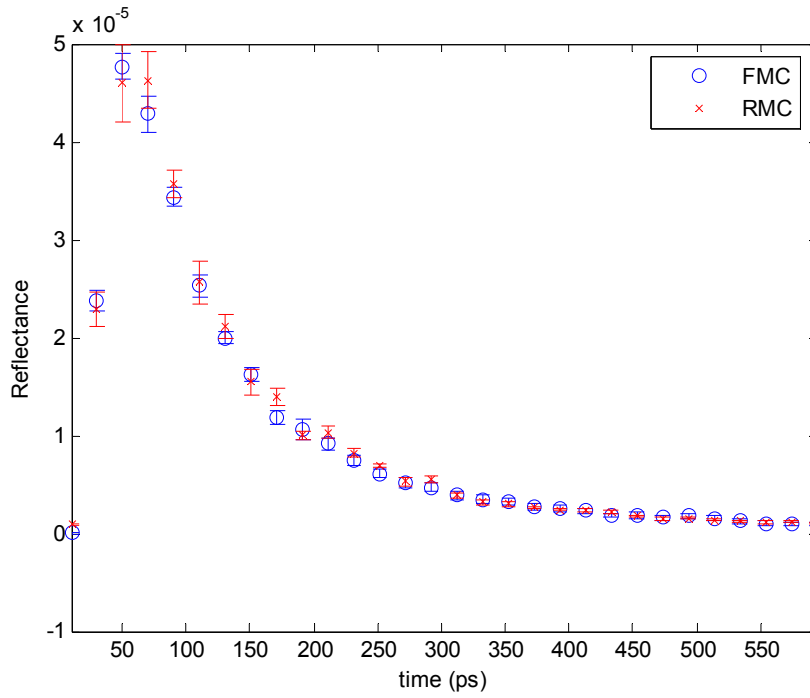**Figure B-5: Comparison between ROM and FMC for values of case 5 (μ_s = 3,173 m^{-1}, g = 0.8964, L = 0.0403 m).**
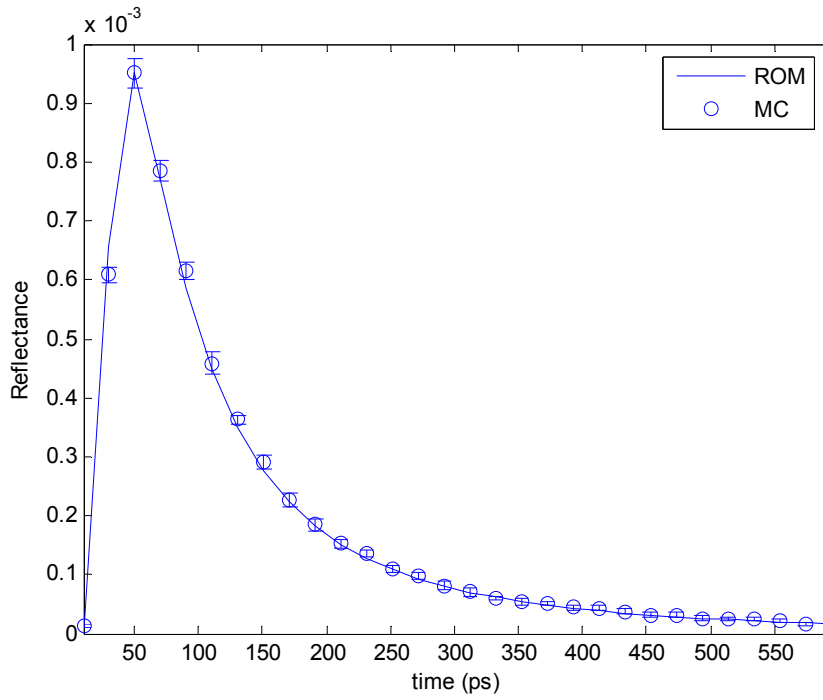
Figure B-6: Comparison between ROM and FMC for values of case 6 (μ_s = 3,768 m^{-1}, g = 0.6710, L = 0.1690 m).

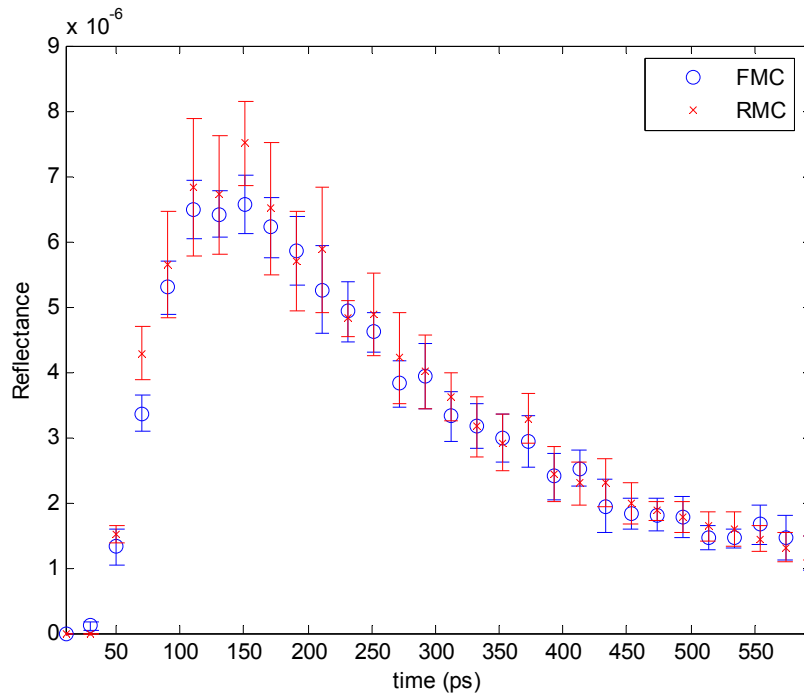**Figure B-7: Comparison between ROM and FMC for values of case 7 (μ_s = 13,410 m^{-1}, g = 0.6590, L = 0.1800 m).**

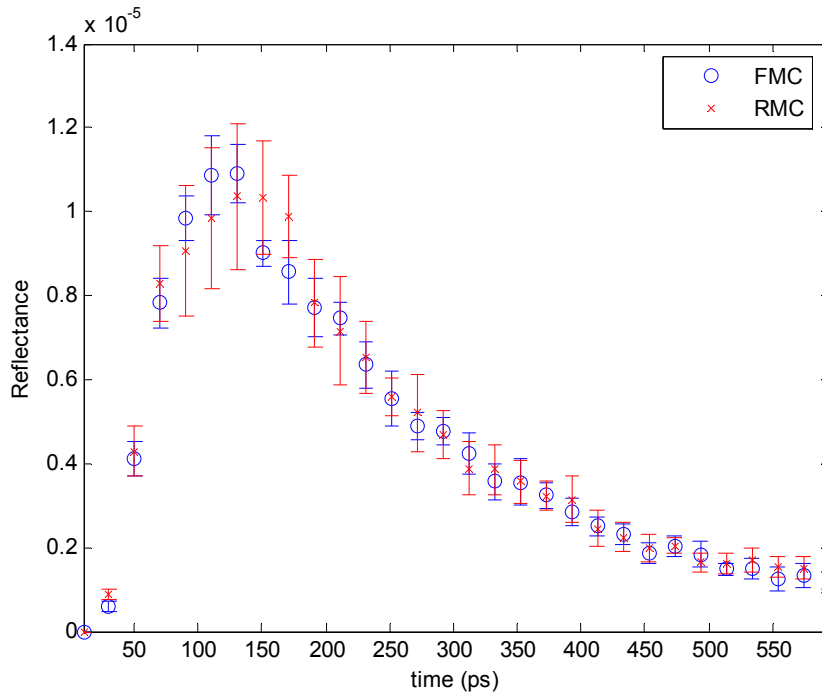**Figure B-8: Comparison between ROM and FMC for values of case 8 (μ_s = 4,950 m^{-1}, g = 0.2970, L = 0.0530 m).**

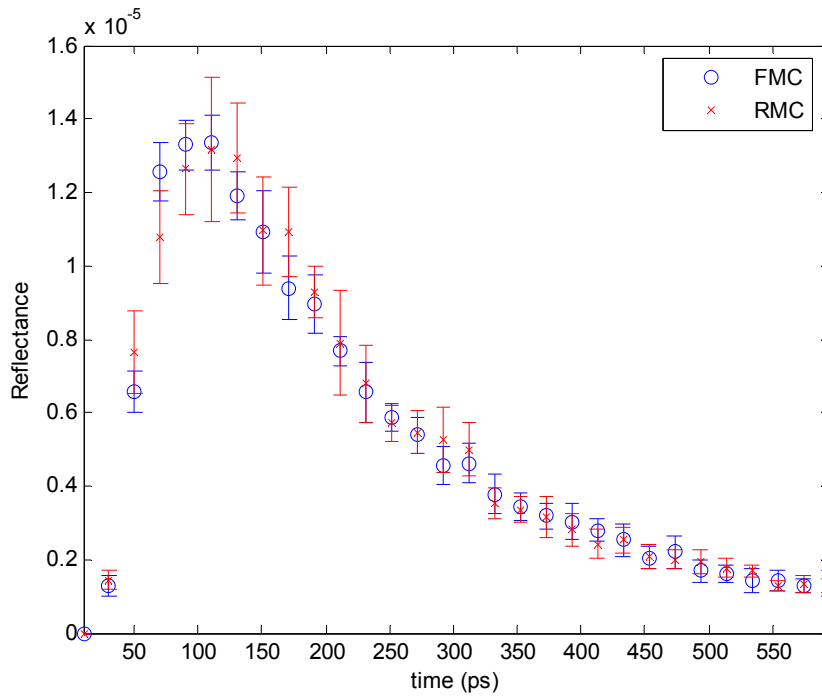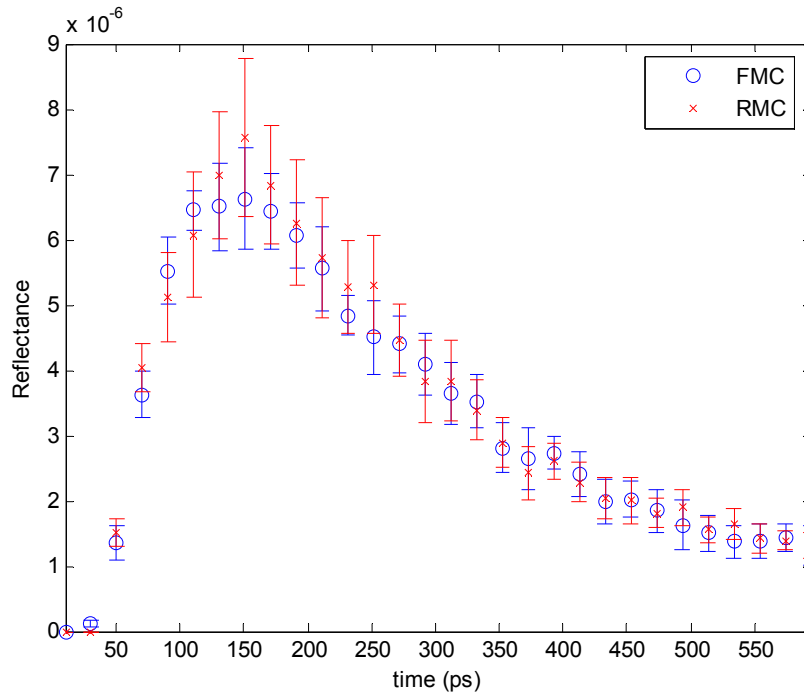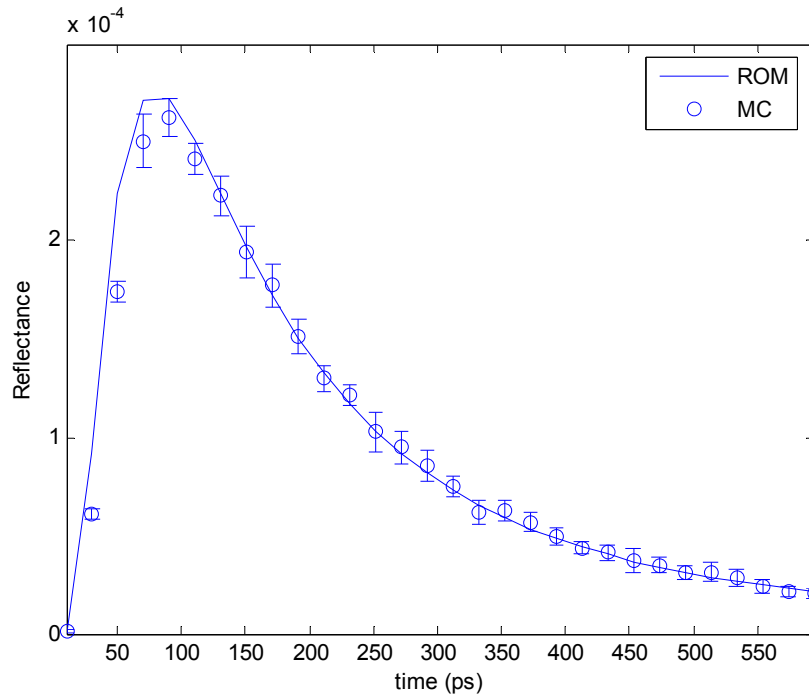**Figure B-9: Comparison between ROM and FMC for values of case 9 (μ_s = 9,865 m^{-1}, g = 0.6920, L = 0.1260 m).**
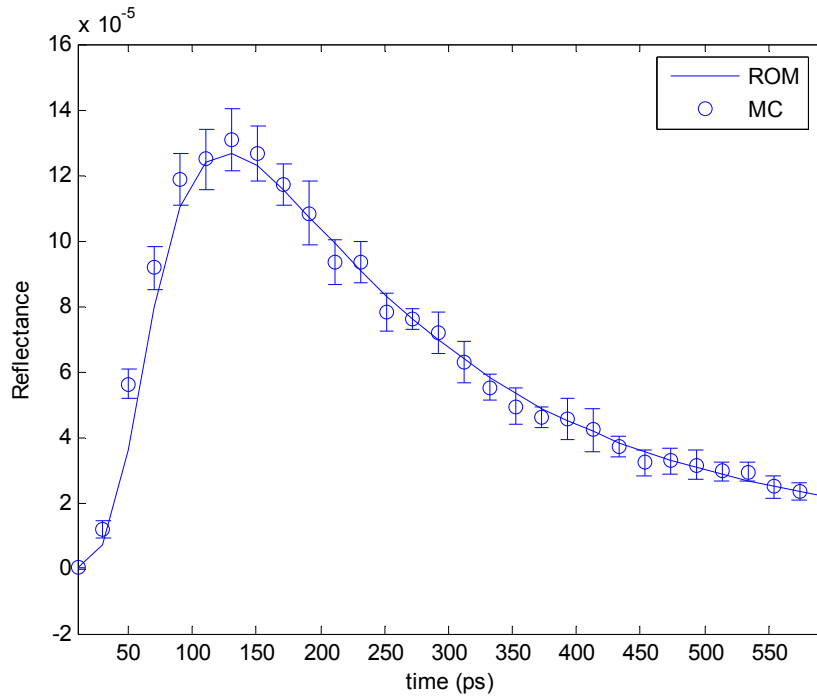
Figure B-10: Comparison between ROM and FMC for values of case 10 (μ_s = 9,737 m^{-1}, g = 0.5390, L = 0.1540 m).

Appendix C.   RMC Source Code

Source code file: rmc.cpp

Compile instructions: use the following command in a UNIX command window
                               in the directory where rmc.cpp is located
                                      g++ rmc.cpp –o *nameofexecutable* –lm –O

Run instruction: run the executable by typing
                                      ./namefoexecutable
                               and then press "Enter."

User inputs: values defining the layer properties

Program output files: revmccs.txt – output columns of time, non-dimensional
                               reflectance data, and standard deviation of means.

Source code:
```
/*This is a reverse Monte Carlo simulation code modeling
reflectance-time profiles of a pulsed laser source traveling through a
glass-foam layer.*/

#include <iostream>
#include <iomanip>
#include <math.h>
#include <fstream>
#include <stdio.h>
#include <stdlib.h>

using namespace std;

#define pi 3.1415926536
#define nf 1.3
#define na 1.0
#define ng 1.5
#define c0 3.0e8
#define c (c0/nf)
#define tmax 16e-9
#define dt 20.1416015625e-12
#define tp dt
#define delztp (c*tp)
#define nt int(tmax/dt)
```

```cpp
#define psrw 0.10
#define irmin 0.001

#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1 - 1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0-EPS)

double ran2(long *idum);

int main ()
{
//define internal variables
long *idum,seed;
int i,N,Np,in,numsmpl,numsmplhf,ns1,it,toggle1,toggle2;
double L,sig,g,QT,DQ,r0,rd,dr,rin,R,qd[50],thd,psi,theta,xi,yi,zi;
double sx,sy,sz,sxnew,synew,sznew,lim,lme,ir,alphai,alphat;
double qdav,stddev,xe,ye,ze,xt,yt,zt,sinth,ri,re,alphac;
double szdet,Ibndl,zU,zL,zc,zo,l1,l2,a,b,c1,d,random;
double Qadj,var,stddevmax,Ref[nt];
double t1,t2,lscat,zp1,zp2;

//Relate variables to their pointers
idum=&seed;
srand((unsigned)time(NULL));
seed=-int(1e6*rand()/RAND_MAX)*2-1;

//user inputs-parameters
g=.5390;                    // assymetry parameter
L=.1540;                    // layer thicknes [m]
sig=9737;                   // scattering coefficient [1/m]  (mus)

QT=1000.;                   // laser power [W]
R=0.001;                    // radius of laser [m]
rd=0.005;                   // radial location of detector on centers [m]
r0=0.0005;                  // detector radius [m]
thd=10;                     // detector acceptance angle [deg]
Np=1000;                    // number of photon bundles
numsmpl=10;                 // number of samples-must be even
stddevmax=0.04;             // maximum standard deviation
dr=2*r0;                    // radial resolution

//output file streams
ofstream data ("revmccs10.txt");
data<<"L="<<L<<endl<<"g="<<g<<endl<<"mus="<<sig<<endl;
data<<"time_(s)"<<setw(25)<<"Refl."<<setw(25)<<"Sx_bar"<<endl;
```

```
numsmplhf=numsmpl/2;
szdet=cos(thd*pi/180);
Qadj=QT/(4.*pi*R*R)*dr*dr*(1.-szdet*szdet);

//loop over time domain
for (it=0;it<30;it++){
N=Np;
ns1=1;
Ref[it]=0;
start:
//loop to generate numsmpl 'samples'
DQ=Qadj/N;
for (i=ns1; i<=numsmpl; i++) {
      qd[i]=0.;
      for (in=1; in<=N; in++) {
            Ibndl=0.;
            t1=(it+0.5)*dt+(ran2(idum))*dt;
            ir=1;

            //point of emmission (square)
            xi=rd+(1.999*ran2(idum)-1)*r0;
            yi=(2*ran2(idum)-1)*r0;
            zi=0;

            //direction of emmision
            sz=sqrt(1-(1-szdet*szdet)*ran2(idum));
            sinth=sqrt(1-sz*sz);
            psi=2*pi*ran2(idum);
            sx=sinth*cos(psi);
            sy=sinth*sin(psi);
scatter:
            //distance to scattering event
            lscat=-log(ran2(idum))/sig;
            toggle1=0;
            toggle2=0;

            //location of scattering event
            xe=xi+lscat*sx;
            ye=yi+lscat*sy;
            ze=zi+lscat*sz;
            t2=t1-lscat/c;
            zp1=c*t1;            //z location of pulse at t1
            zp2=c*t2;            //z location of pulse at t2

            //check if time is up
            if (t2<0) {
                  lscat=c*t1;
                  xe=xi+lscat*sx;
                  ye=yi+lscat*sy;
                  ze=zi+lscat*sz;
                  t2=t1-lscat/c;
                  zp1=c*t1;
                  zp2=c*t2;
            }

            //check whether bundle leaves domain at top and calculate
            internal reflection
```

```
if (ze<0.) {
        xt=xe;
        yt=ye;
        zt=ze;                //location on transmitted side
        alphai=pi-acos(sz);     //incident angle
        alphac=asin(na/nf);     //critical angle
        if (toggle1==0) {       //first half of reflection
                lim=-zi/sz;
                lme=lscat-lim;
                lscat=lim;
                xe=xi+lim*sx;
        ye=yi+lim*sy;
                ze=zi+lim*sz;
                t2=t1-lim/c;
                zp2=c*t2;
                toggle1=1;
        }
        else {                          //second half of reflection
toggle1:
                xi=xe;
                yi=ye;
                zi=ze;
                xe=xt;
                ye=yt;
                ze=-zt;
                sz=-sz;
                lscat=lme;
                t1=t2;
                t2=t1-lme/c;
                zp1=c*t1;
                zp2=c*t2;
                if (alphai>0.05 && alphai<alphac) {
                        alphat=asin(nf*sin(alphai)/na);
                        ir=ir*((pow(sin(alphai-
alphat),2)/pow(sin(alphai+alphat),2)+pow(tan(alphai-
alphat),2)/pow(tan(alphai+alphat),2))/2);
                }
                else if (alphai<=0.05)
                        ir=ir*pow((nf-na),2)/pow((nf+na),2);
                else
                        ir=ir;
                toggle1=0;
        }
}
//check whether bundle leaves domain at bottom and
calculate internal reflection
if (ze>L) {
        xt=xe;
        yt=ye;
        zt=ze;                //location on transmitted side
        alphai=acos(sz);  //incident angle
        if (toggle2==0) { //first half of reflection
                lim=(L-zi)/sz;
                lme=lscat-lim;
                lscat=lim;
                xe=xi+lim*sx;
                ye=yi+lim*sy;
```

```
                        ze=zi+lim*sz;
                        t2=t1-lim/c;
                        zp2=c*t2;
                        toggle2=1;
            }
            else {                      //second half of reflection
toggle2:
                        xi=xe;
                        yi=ye;
                        zi=ze;
                        xe=xt;
                        ye=yt;
                        ze=2*L-zt;
                        sz=-sz;
                        lscat=lme;
                        t1=t2;
                        t2=t1-lme/c;
                        zp1=c*t1;
                        zp2=c*t2;
                        if ((nf*sin(alphai)/ng)>1)
                                ir=ir;
                        else if (alphai<=0.05)
                                ir=ir*pow((nf-ng),2)/pow((nf+ng),2);
                        else {
                                alphat=asin(nf*sin(alphai)/ng);
                                ir=ir*((pow(sin(alphai-
alphat),2)/pow(sin(alphai+alphat),2)+pow(tan(alphai-
alphat),2)/pow(tan(alphai+alphat),2))/2);
                        }
                        toggle2=0;
            }
        }
        //check what part of lscat is inside collimated beam
        ri=sqrt(xi*xi+yi*yi);
        re=sqrt(xe*xe+ye*ye);
        a=ri*ri-R*R;
        b=sx*xi+sy*yi;
        c1=sx*sx+sy*sy+1.e-10;
        d=b*b-a*c1;

        if (d<0.) goto check;

        if (zi>zp1) {
                qd[i]=qd[i]+DQ*Ibndl;
                continue;
        }

        if (ri<R) {
        //initial point is inside
                if(re<R) {
                        if ((zp1-delztp)<=zi && zi<=zp1) {
                                if ((zp2-delztp)<=ze && ze<=zp2){
                                        zL=zi;
                                        zU=ze;
                                }
                                else if (ze>zp2) {
                                        zL=zi;

                                73
```

```
                                        zU=(zi+sz*zp1)/(sz+1.);
                                }
                        }
                        else if (zi<(zp1-delztp)) {
                                if (ze<(zp2-delztp)) goto check;
                                else if ((zp2-delztp)<=ze && ze<=zp2) {

                                        zL=(sz*(zp1-delztp)+zi)/(sz+1);
                                        zU=ze;
                                }
                                else if (ze>zp2) {
                                        zL=(sz*(zp1-delztp)+zi)/(sz+1);
                                        zU=(zi+sz*zp1)/(sz+1.);
                                }
                        }
                }
                else {
                        l1=(-b+sqrt(d))/c1;
                        zc=zi+l1*sz;
                        if ((zp1-delztp)<=zi && zi<=zp1){
                                if ((zp1-delztp-l1)<=zc && zc<=(zp1-l1))
                                {
                                        zL=zi;
                                        zU=zc;
                                }
                                else if (zc>(zp1-l1)) {
                                        zL=zi;
                                        zU=(zi+sz*zp1)/(sz+1.);
                                }
                        }
                        else if (zi<(zp1-delztp)) {
                                if ((zp1-delztp-l1)<=zc && zc<=(zp1-l1))
                                {
                                        zL=(sz*(zp1-delztp)+zi)/(sz+1);
                                        zU=zc;
                                }
                                else if (zc>(zp1-l1)){
                                        zL=(sz*(zp1-delztp)+zi)/(sz+1);
                                        zU=(zi+sz*zp1)/(sz+1.);
                                }
                                else if (zc<(zp1-delztp-l1)) goto check;
                        }
                }
        }
}
else {
        //initial point is outside
        if (re<R) {
                l2=(-b-sqrt(d))/c1;
                zc=zi+l2*sz;
                if ((zp2-delztp)<=ze && ze<=zp2) {
                        if ((zp1-delztp-l2)<=zc && zc<=(zp1-l2))
                        {
                                zL=zc;
                                zU=ze;
                        }
                        else if (zc<(zp1-delztp-l2)) {
                                zL=(sz*(zp1-delztp)+zi)/(sz+1);
```

74

```
                                        zU=ze;
                                }
                        }
                        else if (zp2<ze) {
                                if ((zp1-delztp-l2)<=zc && zc<=(zp1-l2))
                                {
                                        zL=zc;
                                        zU=(zi+sz*zp1)/(sz+1.);
                                }
                                else if (zc<(zp1-delztp-l2)) {
                                        zL=(sz*(zp1-delztp)+zi)/(sz+1);
                                        zU=(zi+sz*zp1)/(sz+1.);
                                }
                                else goto check;
                        }
                        else if (ze<(zp2-delztp)) goto check;
                }
                else {
                l1=(-b+sqrt(d))/c1;
                        l2=(-b-sqrt(d))/c1;
                        if((l2>lscat) || (l1<0.)) goto check;
                if (l1>lscat) l1=lscat;
                        zc=zi+l2*sz;
                        zo=zi+l1*sz;
                        if ((zp1-delztp-l2)<=zc && zc<=(zp1-l2)) {
                                if ((zp1-delztp-l1)<=zo && zo <=(zp1-l1))
                                {
                                        zL=zc;
                                        zU=zo;
                                }
                                else if ((zp1-l1)<zo) {
                                        zL=zc;
                                        zU=(zi+sz*zp1)/(sz+1.);
                                }
                        }
                        else if (zc<(zp1-delztp-l2)) {
                                if (((zp1-delztp)-l1)<=zo && zo <=(zp1-
                                l1)) {
                                        zL=(sz*(zp1-delztp)+zi)/(sz+1);
                                        zU=zo;
                                }
                                else if (zo<(zp1-delztp-l1)) goto check;
                                else if ((zp1-l1)<zo) {
                                        zL=(sz*(zp1-delztp)+zi)/(sz+1);
                                        zU=(zi+sz*zp1)/(sz+1.);
                                }
                        }
                        else if (zc>(zp1-l2)) goto check;
                }
        }
        Ibndl=Ibndl+ir*((1-g*g)/(2*pow((1+g*g+2*g*sz),1.5)))*(exp(-
sig*zL)-exp(-sig*zU))/sz;

check:
        if (toggle1==1) goto toggle1;
        if (toggle2==1) goto toggle2;
        if (ze>zp2) {
```

```
                qd[i]=qd[i]+DQ*Ibndl;
                continue;
        }
        //check to see if time is up
        if (t2<=0) {
                qd[i]=qd[i]+DQ*Ibndl;
                continue;
        }
        //decide wether or not to continue bundle
        random=double(rand())/RAND_MAX;
        if (ir<irmin && random>psrw) {
                qd[i]=qd[i]+DQ*Ibndl;
                continue;
        }

        //endpoint becomes starting point
        xi=xe;
        yi=ye;
        zi=ze;
        t1=t2;

        // new direction after anisotropic scattering
        if (g<0.01) {
                theta=acos(2*ran2(idum)-1);
        }
        else {
                theta=acos((1+g*g-pow(((1-g*g)/(1-
                g+2*g*ran2(idum))),2))/(2*g));
        }

        psi=2*pi*ran2(idum);

        if(fabs(sz)>=0.99999) {
                sxnew=sin(theta)*cos(psi);
                synew=sin(theta)*sin(psi);
                sznew=sz*cos(theta)/fabs(sz);
        }
        else {
                sxnew=sin(theta)*(sx*sz*cos(psi)-sy*sin(psi))/sqrt(1-
sz*sz)+sx*cos(theta);
                synew=sin(theta)*(sy*sz*cos(psi)+sx*sin(psi))/sqrt(1-
sz*sz)+sy*cos(theta);
                sznew=-sin(theta)*cos(psi)*sqrt(1-
sz*sz)+sz*cos(theta);
        }

        sx=sxnew;
        sy=synew;
        sz=sznew;

    goto scatter;

    }
}

//find average value and std dev
qdav=0.;
```

```
for (i=1; i<=numsmpl;i++) {
      qdav=qdav+qd[i];
}
qdav=qdav/numsmpl;
stddev=0.;
for (i=1; i<=numsmpl; i++) {
      stddev=stddev+pow((qd[i]-qdav),2);
}
var=sqrt(stddev/(numsmpl*(numsmpl-1)));
if (qdav==0) stddev=0;
else stddev=var/qdav;

cout<<setw(14)<<N*numsmpl<<"
"<<setw(15)<<qdav*pi*R*R/(QT*dr*dr)<<setw(15)<<stddev*100<<endl;

if (stddev>stddevmax && N<2048000) {
      //combine pairs of samples and double number of bundles
      for (i=1; i<=numsmplhf; i++) {
            qd[i]=0.5*(qd[i]+qd[i+numsmplhf]);
      }
      N=2*N;
      ns1=numsmplhf+1;
      goto start;
}

      //print out reflection data
      Ref[it]=qdav*pi*R*R/(QT*dr*dr);
      cout<<it<<"\t"<<Ref[it]<<endl;
      data<<(it+0.5)*dt<<setw(25)<<Ref[it]<<setw(25)<<var*pi*R*R/(QT*dr
*dr)<<endl;
}
return (0);

}

//Random number generator
double ran2(long *idum)
{
int j;
long k;
static long idum2=123456789;
static long iy=0;
static long iv[NTAB];
double temp;

if(*idum<=0)
{
      if(-(*idum)<1) *idum=1;
      else *idum=-(*idum);
      idum2=(*idum);
      for (j=NTAB+7; j>=0; j--)
      {
            k=(*idum)/IQ1;
            *idum=IA1*(*idum-k*IQ1)-k*IR1;
            if (idum<0) *idum+=IM1;
      if (j<NTAB) iv[j]=*idum;
      }
```

```
        iy=iv[0];
}
k=(*idum)/IQ1;
*idum=IA1*(*idum-k*IQ1)-k*IR1;
if (*idum<0) *idum+=IM1;
k=idum2/IQ2;
idum2=IA2*(idum2-k*IQ2)-k*IR2;
if(idum2 < 0) idum2+=IM2;
j=iy/NDIV;
iy=iv[j]-idum2;
iv[j]=*idum;
if(iy < 1) iy+=IMM1;
temp=AM * iy;
if(temp > RNMX) return RNMX;
else return temp;
}
```

Appendix D.   ROM Source Code

<u>Source code file</u>:	ROM.m
<u>Compile instruction</u>:	In MATLAB enable cells. Once the file is loaded into the editor window, simply click 'run'. Or each cell may be evaluated individually. Once the first cells have been executed the cell titled "Arbitrary Profile" can be evaluated independently.
<u>User inputs</u>:	Radiative parameters for an arbitrary profile
<u>Program input files</u>:	PodData_5mm_Raw.xls – contains demensional reflectance profiles generated using a forward Monte Carlo simulation

<u>Source code</u>:

```
%This is a reduced order modeling simulation based on singular value
%decomposition. It calculates reflectance-time profiles for a set of
%parameters at arbitrary values using previously calculated profiles at
%predetermined values.

clear all; clc;
t1=cputime;

N_photons=10^7;     %number of photons used to create snapshots
r_laser=.001;       %radius of laster
r_location=0.005;   %radial location snapshots are taken at
delta_r=.001;       %radial resolusion

dt=20.1416015625e-12;
t=[dt/2:dt:dt*800]';
%input snapshot matrix A
A =xlsread('PodData_5mm_Raw','a1:cv40');

%nondimensionalize A
A=A*r_laser^2/(N_photons*2*r_location*delta_r);
```

```
[N,M]=size(A);


%parameters used to create A
g = [.2:.2:.8];
L = [.04:.04:.2];
mus=[2800:2800:14000];


gmax=max(g);
Lmax=max(L);
musmax=max(mus);


ng =length(g);
nL =length(L);
nmus =length(mus);


tol=1e-4;    %for singular values


%matrix of parameter sets
is=1;
for i=1:ng
    for j=1:nL
        for k=1:nmus
            K(is,:) = [g(i) L(j) mus(k)];
            is=is+1;
        end
    end
end


%matrix of interpolation functions
for i=1:M
    for j=1:M
            F(i,j)=1/sqrt(((K(i,1)-K(j,1))/gmax)^2+((K(i,2)-
            K(j,2))/Lmax)^2+((K(i,3)-K(j,3))/musmax)^2+1);
    end
end


%number of significant eigenvalues
n_fe=6;


[U_A SIG_A V_A]=svd(A);


Fe=U_A(:,1:n_fe);
%calculate coefficient matrix
B=Fe'*A;


%perform Singular Value Decomposition on F to get Moore-Penrose inverse
[U_F,Sig_F,V_F]=svd(F);
Sa=Sig_F;


%zero small singular values
for i=1:M
    if Sig_F(i,i) > tol
        Sig_F(i,i)=1/Sig_F(i,i);
    else
        Sig_F(i,i)=0;
```

```matlab
    end
end


%get C matrix using singular value decomposition
C=B*V_F*Sig_F*U_F';


%% Arbitrary Profile
%=====================================================================
t2=cputime;
N_photons_2=10^5;    %number of photons for test cases

ga = 0.5154;
La = 0.1415;
musa = 11621;
Ka = [ga La musa];


%interpolation function
for i=1:M
    Fa(i)=1/sqrt(((Ka(1)-K(i,1))/gmax)^2+((Ka(2)-
        K(i,2))/Lmax)^2+((Ka(3)-K(i,3))/musmax)^2+1);
end


Ba=C*Fa';


%arbitrary field
Aa=Fe*Ba;


%input A1 exact for comparison
A1=xlsread(['test' test_num '_ave'],'L4:L104');
E=xlsread(['test' test_num '_ave'], 'O4:O104');
%nondimensionalize A1
A1=A1*r_laser^2/(N_photons_2*2*r_location*delta_r);
E=E*r_laser^2/(N_photons_2*2*r_location*delta_r);


%graph data
figure(2)
hold on
plot(t(1:40),Aa(1:40),'-')
errorbar(t(1:40),A1(1:40),E(1:40),'o')
legend('ROM','MC')
axis ([t(1) t(40) -1 1]); axis 'auto y';grid on;box on
ylabel('Reflectance'); xlabel('time (ps)')
hold off


%calculate relative error
for i=1:40
    error(i)=abs((Aa(i)-A1(i))/A1(i))*100;
end


%display error information
err_sum=sum(error(5:40));
err_max=max(error(5:40));
err_ave=mean(error(2:40));
disp(['average error = ' num2str(err_ave)]);
```

```matlab
Nu=40;      %number of points to calc R-squared
Aa_ave=mean(Aa(1:Nu));
% fitness
sum1=0;
sum2=0;

for i=1:Nu
    sum1=sum1 + (Aa(i)-A1(i))^2;
    sum2=sum2 + (Aa(i)-Aa_ave)^2;
end

m=3;
Syx=sqrt(sum1/(Nu-(m+1)));
Sy=sqrt(sum2/(Nu-1));
R=sqrt(1-Syx^2/Sy^2);

display(['R-squared = ' num2str(R^2)]);

%%
```