

Advanced 3D Monte Carlo Algorithms for Biophotonic and Medical Applications

Lewis McMillan



University of
St Andrews

This thesis is submitted in partial fulfilment for the degree of
PhD
at the
University of St Andrews

March 2019

Declaration

I, Lewis McMillan, hereby certify that this thesis, which is approximately ***** words in length, has been written by me, that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in September 2015 and as a candidate for the degree of PhD in September 2015; the higher study for which this is a record was carried out in the University of St Andrews between 2015 and 2019.

Date Signature of candidate

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date Signature of supervisor

Date Signature of supervisor

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Acknowledgements

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Contents

Declaration	iii
Abstract	v
Acknowledgements	vii
Abbreviations	ix
List of Figures	x
1 Monte Carlo Radiation Transport Technique	1
1.1 Introduction and Background	1
1.2 Monte Carlo Radiation Transport Algorithm	4
1.2.1 Introduction & background	4
1.2.2 Optical Properties	7
1.2.2.1 Derived Parameters	9
1.2.3 MCRT Algorithm	12
1.2.4 Code Details	17
1.3 Validation of MCRT Code	20
1.4 Conclusion	21

Abbreviations

K-M theory Kubelka-Munk Theory.

MCRT Monte Carlo radiation transfer.

MPI Message-passing interface.

PDF probability density function.

RTE radiative transfer equation.

List of Figures

- 1.1 Sample Buffon needle experiment. 100 needles are dropped on a 10×10 cm area with lines spaced 1.5 cm apart. If a needle lands on a line it is recorded and coloured blue, else it is red. This simulation gave a value of $\pi \approx 3.10$.
- 1.2 Illustration of the rejection method for determining π from the area of a circle inscribed within a square. The ratio of the area of the circle to the square is $\frac{\pi}{4}$. Thus the ratio of darts landing in the circle to those that land outside the circle is $\pi \approx \frac{4N_{inner}}{N_{total}}$, where N_{total} is the total number of darts, and N_{inner} is the total number of darts that land in the circle. Using 200 darts gave a value of $\pi \approx 3.12$.
- 1.3 Computer generated imagery using ray tracing. The Monte Carlo method is used to “compute radiance along ray paths between lights and the camera”, in order to generate CGI images [11].
- 1.4 Energy flow through area dA within solid angle $d\Omega$ in a direction \hat{s} . Adapted from [19, 20]
- 1.5 Cylindrical volume element, $ds dA$, with solid angle $d\Omega$ in direction \hat{s} and solid angle $d\Omega'$ in direction \hat{s}' . Energy flowing through this element is used to derive the radiation transfer equation. Adapted from [19, 20].
- 1.6 Figure show the g factor for the Henyey-Greenstein phase function, for various configurations of back, forward or isotropic scattering.
- 1.7 Spectra of some of the more common absorbers found in the skin [31, 35–43].
- 1.8 Flowchart of the Monte Carlo radiation transport algorithm as described in this section.
- 1.9 Example of a possible voxel model, with three different layers, various holes due to ablative pixel beam lasers (see ??). Each voxel can represent a different optical/thermal property of the tissue medium.
- 1.10 Illustration of photon propagation through a 2D grid. d_{x1} , and d_{y1} are the distances to the voxel walls in the x and y directions in the μ_1 voxel. In this case $S_1 = d_{x1}$ as d_{x1} is smaller than d_{y1} , thus the photon hits the voxel wall in the x direction. For the μ_5 voxel, d_y is smaller, thus the photon hits the voxel wall in the y^{th} direction.
- 1.11 Illustration of rotating the the centre of mass frame to the lab frame. \mathbf{n} is the direction vector of the photon before scattering, and \mathbf{n}_s is the scattered direction vector. θ and φ are the scattering angles. z_s is in the same direction as \mathbf{n} .
- 1.12 Source code hierarchy, showing the relationship between different modules. Green is the entry point for the simulation. Red are the data modules, light blue are the routine modules, and grey are the external dependencies.
- 1.13 Performance of the parallelisation of the MCRT code using MPI.
- 1.14 Figure shows the fluence as a function of depth. Figure also shows comparison to the Jacques MCRT simulation and the MCRT as described in this chapter.

Chapter 1

Monte Carlo Radiation Transport Technique

1.1 Introduction and Background

This chapter will provide an overview of the Monte Carlo method and how it is used within the context of Monte Carlo radiation transfer (MCRT). The chapter will then present the details of the MCRT code used as the basis of the subsequent chapters. Validation of this code and details of computational speed up are also presented. Subsequent chapters will expand upon the code for each individual projects needs.

Monte Carlo Method

The Monte Carlo method is a numerical analysis technique based upon random numbers, which are used to calculate unknown variables in problems [1, 2].

The earliest use of the method is in Buffon's needle experiment of the 18th century [3–5]. Buffon asked the question;

“Suppose we have a floor made of parallel strips of wood, each the same width, and we drop a needle onto the floor. What is the probability that the needle will lie across a line between two strips?”

The solution to this question is: for a needle length l , strip separation s , where x is the distance from the needle to the closest line, and θ is the angle of the needle with respect to the wood strips. Then using a simple geometrical argument, a needle crosses a strip if $x \leq \frac{l}{2} \sin\theta$.

x is distributed uniformly in $[0, \frac{s}{2}]$, and θ in $[0, \frac{\pi}{2}]$. Therefore the probability density function for x is $p(x) = \frac{2}{s}$, and θ is $p(\theta) = \frac{2}{\pi}$. The probability density function (PDF), is a function of a variable that gives probability for a variable to take a given value. The PDF is normalised over the whole range of the variable, in this case x , and θ . Thus, as x and θ are independent variables, giving a joint probability of $p(x, \theta) = \frac{4}{s\pi}$. So the probability of a needle of length l ($l < s$) is:

$$P = \int_0^{\frac{\pi}{2}} \int_0^{\frac{l}{2} \sin\theta} \frac{4}{s\pi} dx d\theta = \frac{2l}{s\pi} \quad (1.1)$$

Equation (1.1) can be used to carry out a Monte Carlo estimation of π . A simple rearrangement yields: $\pi = \frac{2l}{sP}$ where P is the ratio of needles crossing the line over total number dropped.

Laplace was the first to suggest that Buffon's needle experiment could be used to estimate π [4]. Figure 1.1 shows an example of simulation of Buffon's needle experiment.

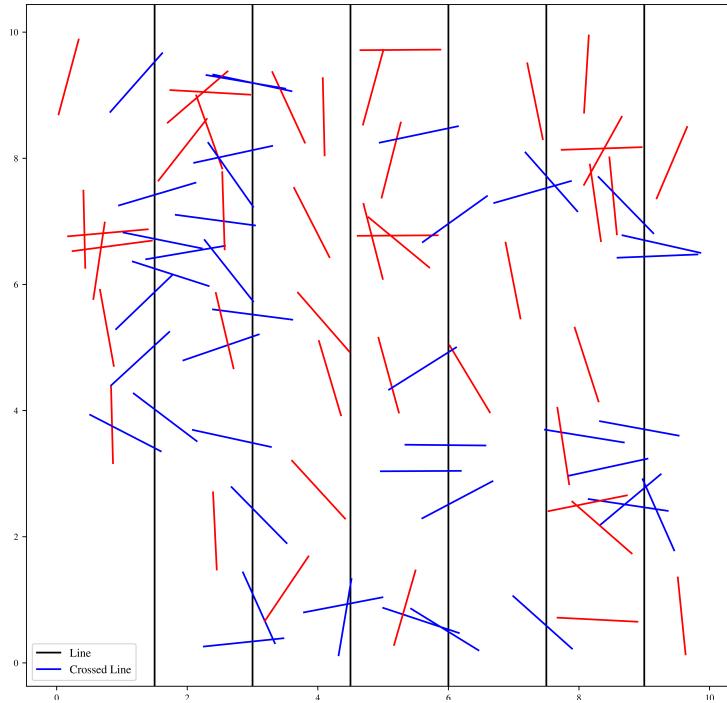


Figure 1.1: Sample Buffon needle experiment. 100 needles are dropped on a 10×10 cm area with lines spaced 1.5 cm apart. If a needle lands on a line it is recorded and coloured blue, else it is red. This simulation gave a value of $\pi \approx 3.10$.

There are various different approaches to using the Monte Carlo method to obtain randomly sampled variables. One analytical way of achieving this is the inverted sampling method. The inverted sampling method can be summarised by the following steps for drawing a sample X_i from an arbitrary PDF $p(x)$:

1. Compute the CDF $P(x) = \int_0^x p(x')dx'$
2. Compute the inverse $P^{-1}(x)$
3. Obtain a uniformly distributed random number ξ
4. Finally, compute $X_i = P^{-1}(\xi)$

If a given problem cannot use the inverted sampling method, as it may not be possible to get a PDF or analytically invert the CDF, then the rejection method can be used. The rejection method is essentially a dart throwing method. This means that points are drawn and compared to the function. If the point lies under the function then the point is accepted, if it lies above the function then it is rejected. For example, if we have a function, $f(x)$, that does not have an analytical PDF, we can use a PDF $p(x)$ such that $f(x) < cp(x)$ where c is a constant. We therefore sample from $p(x)$, and if the sampled point lies under $f(x)$ it is accepted, else it is rejected. Figure 1.2 shows an example of this process.

The Monte Carlo method is used in various different disciplines. Ranging from use in the financial sector to analyse investments and stocks by simulating the sources of uncertainty which affect their values [6, 7], use in statistical analysis [8], and in modern computer generated images

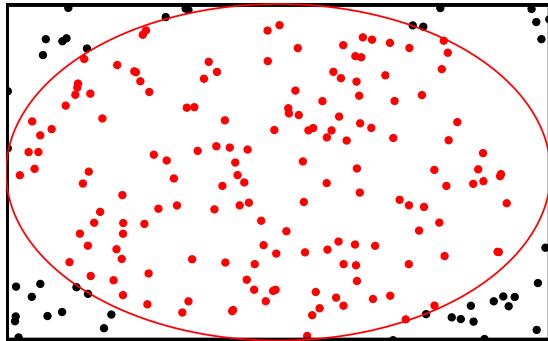


Figure 1.2: Illustration of the rejection method for determining π from the area of a circle inscribed within a square. The ratio of the area of the circle to the square is $\frac{\pi}{4}$. Thus the ratio of darts landing in the circle to those that land outside the circle is $\pi \approx \frac{4N_{inner}}{N_{total}}$, where N_{total} is the total number of darts, and N_{inner} is the total number of darts that land in the circle. Using 200 darts gave a value of $\pi \approx 3.12$

(see Fig. 1.3) [9, 10]. It is also widely used in astronomy and medicine, in order to simulate the propagation of radiation through turbid media. This technique, MCRT, is what makes up the bulk of this thesis and is described in depth in the following sections.

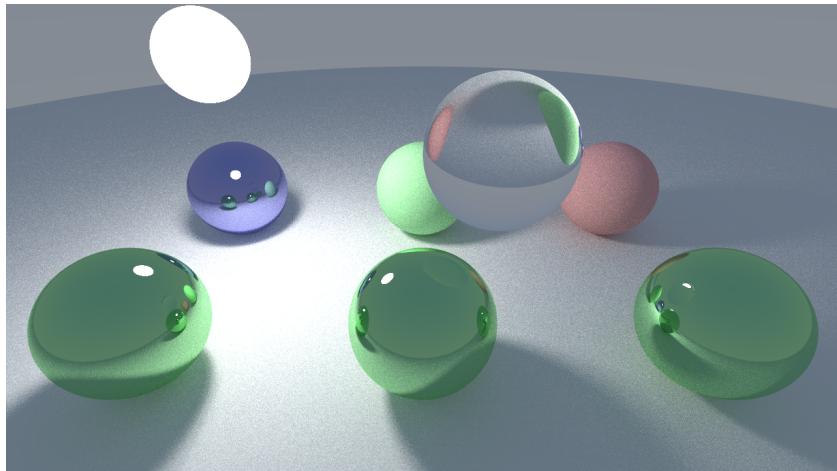


Figure 1.3: Computer generated imagery using ray tracing. The Monte Carlo method is used to “compute radiance along ray paths between lights and the camera”, in order to generate CGI images [11].

1.2 Monte Carlo Radiation Transport Algorithm

1.2.1 Introduction & background

The technique that makes up the bulk of this thesis, is the MCRT technique. This method was developed at the end of the Second World War at the Los Alamos National Laboratory, for the purpose of calculating neutron diffusion through shielding material [12–15]. It has since found a myriad of applications from light transport through dusty galactic clouds [16], calculating doses for radiotherapy [17] to light transport through tissue [18].

Radiative Transfer

Transport of photons through turbid media, can be modelled analytically using the radiative transfer equation (RTE). The RTE models the radiative losses, and gains by a beam of radiation as it travels through a medium, including: loss of energy due to absorption, loss/gain of energy due to scattering, and energy gain due to emission. Before we derive the RTE, we first define some terms and physical quantities.

The first term is spectral irradiance, L_ν . Spectral irradiance is defined as the energy flow in a direction \hat{n} , for a solid angle $d\Omega$, per unit time per unit temporal frequency bandwidth. Irradiance is defined as the spectral irradiance over a small frequency range $[\nu, \nu + \Delta\nu]$:

$$L(\vec{r}, \hat{s}, t) = L_\nu(\vec{r}, \hat{s}, t)\Delta\nu \quad (1.2)$$

Where:

- \vec{r} is the position;
- \hat{s} is the unit normal vector;
- t is the time;
- and $L(\vec{r}, \hat{s}, t)$ is the irradiance [$Wm^{-2}sr^{-1}$].

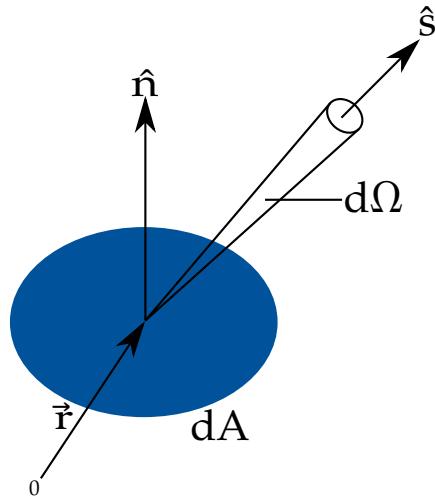


Figure 1.4: Energy flow through area dA within solid angle $d\Omega$ in a direction \hat{s} . Adapted from [19, 20]

The irradiance can be used to determine the energy, dE , transported across an area dA , in a solid angle $d\Omega$ in a time dt (see Fig. 1.4) is:

$$dE = L(\vec{r}, \hat{s}, t) \cdot (\hat{s} \cdot \hat{n}) dA d\Omega dt \quad (1.3)$$

Where:

- \hat{n} is the unit normal to dA ;
- and $\hat{s} \cdot \hat{n}$ is the angle of the solid angle.

Irradiance can also be used to determine the fluence rate, ϕ , which is defined as the energy flow per unit time, independent of the flow direction.

$$\phi(\vec{r}, t) = \int_{4\pi} L(\vec{r}, \hat{s}, t) d\Omega \quad (1.4)$$

Where:

- ϕ is the fluence rate [$W m^{-2}$].

Solving the RTE yields the irradiance which gives the distribution of light in the medium, and gives information on the state of the system and all the physical properties of it.

With the irradiance defined, as well as the other quantities that follow, we can now derive the RTE [19, 20]. We first consider conservation of energy, as shown in Eq. (1.5).

$$dP = -dp_{div} - dp_{ext} + dP_{scatt} + dP_{src} \quad (1.5)$$

Where:

dP is the total change in energy in the volume $dA ds$ within the solid angle, $d\Omega$, per unit time (see Fig. 1.5);

dP_{div} is the energy loss due to the divergence of the radiation beam per unit time;

dP_{ext} is the energy loss due to absorption and scattering within $dA ds d\Omega$;

dP_{scatt} is the energy gain due to scattering from \hat{s}' into $d\Omega$ per unit time;

and dP_{src} is the energy gain due to emission within the medium, per unit time.

The total change in energy in the volume element within the solid angle $d\Omega$, dP , is equal to:

$$dP = \frac{1}{c} \frac{\partial L(\vec{r}, \hat{s}, t)}{\partial t} dA ds d\Omega \quad (1.6)$$

Where c is the speed of light.

The first loss term, dP_{div} , is the energy loss due to divergence of the radiation beam. This is modelled as:

$$dP_{div} = \frac{\partial L}{\partial s} d\Omega dV \quad (1.7)$$

$$= \hat{s} \cdot \nabla L(\vec{r}, \hat{s}, t) d\Omega dV \quad (1.8)$$

dP_{ext} is the second loss term, and accounts for energy loss due to scattering and absorption in the volume element within the solid angle $d\Omega$. This is modelled as:

$$dP_{ext} = \mu_t ds L(\vec{r}, \hat{s}, t) dA d\Omega \quad (1.9)$$

The first energy gain term, dP_{src} , is due to emission in the volume element within the solid angle $d\Omega$.

$$dP_{src} = S(\vec{r}, \hat{s}, t) dV d\Omega \quad (1.10)$$

The second energy gain term, and final term, is due to the incident energy on the volume element within the solid angle $d\Omega$ in direction \hat{s} due to scattering from any direction \hat{s}' .

$$dP_{scatt} = N_s dV \left(\int_{4\pi} L(\vec{r}, \hat{s}', t) P(\hat{s}', \hat{s}) \sigma_s d\Omega' \right) d\Omega \quad (1.11)$$

$$= \mu_s dV \left(\int_{4\pi} L(\vec{r}, \hat{s}', t) P(\hat{s}', \hat{s}) d\Omega' \right) d\Omega \quad (1.12)$$

Where:

N_s is the number density of scatters;
 $P(\hat{s}', \hat{s})$ is the scattering phase function (see Section 1.2.2 for further discussion);
and σ_s is the cross section of the scatters, thus $\mu_s = N_s \sigma_s$ (again see Section 1.2.2 for further discussion).

Finally substituting Eqs. (1.6), (1.8) to (1.10) and (1.12) into Eq. (1.5) yields the RTE:

$$\frac{1}{c} \frac{\partial L(\vec{r}, \hat{s}, t)}{\partial t} + s \cdot \nabla L(\vec{r}, \hat{s}, t) = -\mu_t L(\vec{r}, \hat{s}, t) + \mu_s \int_{4\pi} p(\hat{s}, \hat{s}') L(\vec{r}, \hat{s}', t) d\Omega' + S(\vec{r}, \hat{s}, t) \quad (1.13)$$

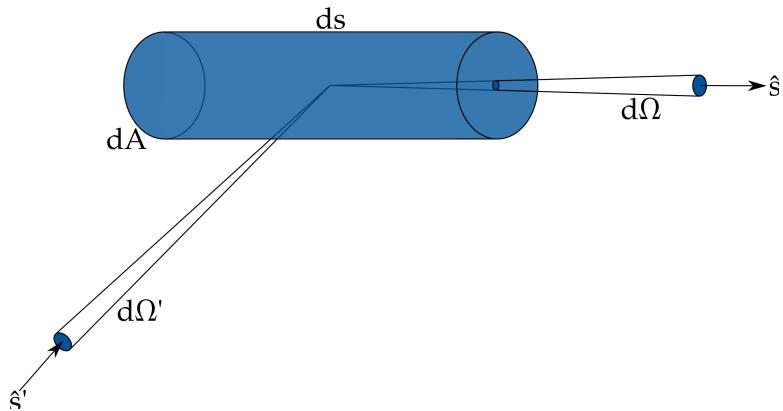


Figure 1.5: Cylindrical volume element, $ds dA$, with solid angle $d\Omega$ in direction \hat{s} and solid angle $d\Omega'$ in direction \hat{s}' . Energy flowing through this element is used to derive the radiation transfer equation. Adapted from [19, 20].

In general, the RTE is hard to solve in arbitrary 3D geometries, however there are a number of approximations, and numerical methods available. Diffusion approximation, Kubelka-Munk Theory (K-M theory), and MCRT are the common methods used to approximate the RTE.

Kubelka-Munk Theory

K-M theory was originally developed in order to calculate the light distribution in thin layered materials, such as paint or paper [21]. The theory is rather simple and makes many assumptions about the medium and the incident light. The main assumptions of K-M theory are: only scattering and absorption take place in the medium, the incident light is already diffuse, the medium is uniform with only isotropic scattering, no external or internal reflections, and the medium is planar and infinitely wide [22–24].

These assumptions make K-M theory poor for modelling light-tissue interactions. This is as in tissue scattering is not isotropic but rather forward biased (see Section 1.2.2). Tissue is rarely, planar and infinitely wide. Tissue also has some reflections at its external and internal boundaries, due to change in refractive indices. Many medical and biophotonic treatments/methods use laser light which is not diffuse. Finally tissue can also exhibit fluorescence, which K-M theory is not able to model, along with polarization. K-M theory does have some positive aspects. It is good at calculating the diffuse reflectance of simple media, and can be used to roughly estimate calculations. Though it is not well suited for modelling light-tissue applications [25].

Diffusion Approximation

The diffusion approximation for the RTE, is where the irradiance is separated into two components:

$$L(\vec{r}, \hat{s}) = L_c(\vec{r}, \hat{s}) + l_d(\vec{r}, \hat{s}) \quad (1.14)$$

Where L_c is the unscattered contribution, which satisfies Beer's law*, and L_d is the diffuse contribution. The L_d component is expanded using Legendre polynomials and truncated. The diffusion approximation also has a number of assumptions and restrictions. The main assumption is that scattering dominates over absorption, and that the scattering is nearly isotropic. This restricts the types of scattering the Diffusion approximation can model, though using similarity relations can partially model scattering in tissue [26, 27].

Diffusion theory is computationally fast, and simple. However it is poor at modelling light-tissue interactions due to its assumptions and restrictions, mainly the inaccurate modelling near the boundaries of the medium and its lack of modelling fluorescence and other microphysics. However it can be used to speed up MCRT in optically thick regions [28, 29].

MCRT

The final method, MCRT, is a method that is numerically equivalent to the RTE [19]. MCRT is a flexible method, it can model arbitrary 3D geometries, various microphysics including fluorescence, and polarisation. It can also model various different light sources, from collimated laser beams, to diffuse light sources. The only downside that is noted in the literature is that the MCRT can be expensive computationally. However with computational power growing faster with each year, this is less of a problem going forward. The next several sections give an in depth description of the MCRT method and its flexibility, along with a description of the code used in this thesis to solve various medical and biophotonic problems.

1.2.2 Optical Properties

Before an in depth description of the MCRT method is outlined, we must discuss optical properties of materials, which the MCRT method needs in order to simulate the transport of photons in a material.

Optical properties of a medium are the properties that describe how light is transported through that medium. Usually the optical properties of a medium are defined by three main parameters: the scattering and absorption coefficients (μ_s and μ_a), and the anisotropy coefficient (g). There are several other optical properties the medium can be defined with, however these in general are only used for specific applications, such as Raman cross-sections for Raman scattering.

*Beer's law (or Beer-Lambert law) states that the transmission, T , is equal to $e^{-\mu L}$, where L is the distance and μ is the attenuation coefficient.

Scattering

The scattering coefficient, along with the anisotropy value (see [Anisotropy](#)), define how light is scattered in a medium. Scattering occurs in skin due to a number of different scatterers, and inhomogeneities found within the skin. The main scatters in the skin are filamentous proteins such as collagen and elastin. These proteins are generally found within the dermis and epidermis [30]. In the upper layers of the skin, the main scatters are keratins and various chromophores such as melanin. The size of the scatters affect how light is scattered and into which direction that light is scattered into.

The scattering of light within tissue is usually defined as μ_s or μ'_s : the scattering coefficient and the reduced scattering coefficient, where $\mu'_s = \mu_s(1 - g)$. The scattering coefficient is defined such that the probability of transmission without scattering in a path length L is:

$$T = e^{-\mu_s L} \quad (1.15)$$

This gives units of inverse length for the scattering coefficient (usually measured in cm^{-1}). The reduced scattering coefficient is quite often given in place of the scattering coefficient, as the reduced coefficient is more easily measured than the “normal” coefficient [31].

Anisotropy

Anisotropy is the degree of deviation that light undergoes at each scattering event. The anisotropy value is taken from the phase function for the medium. The phase function, $\Phi(\theta, \phi)$, is usually normalised over all angles:

$$\int_{\Omega} \Phi(\theta, \phi) d\Omega = 1 \quad (1.16)$$

Where θ , and ϕ are the usual spherical angles. Thus the usual Rayleigh scattering and isotropic scattering, phases function’s are:

$$\Phi_{isotropic}(\theta, \phi) = \frac{1}{4\pi} \quad (1.17)$$

$$\Phi_{Rayleigh}(\theta, \phi) = \frac{3}{8\pi}(1 + \cos^2(\theta)) \quad (1.18)$$

For simplicity, the phase function is usually cast as the anisotropy value g , which is defined as the average angle of deflection:

$$g = \langle \cos(\theta) \rangle = \int_{\Omega} \cos(\theta) \Phi(\theta, \phi) d\Omega \quad (1.19)$$

The anisotropy factor, g , can take on any value from -1 to 1 . Where a value of -1 is highly back scattering, 0 is isotropic scattering, and 1 is highly forward scattering (see Fig. [1.6](#)).

There are many phase functions that can be used to model the anisotropy factor in a medium. The standard phase function in biological tissue is the Henyey-Greenstein phase function. The Henyey-Greenstein phase function, was originally created to modelling diffuse radiation in the galaxy [32, 33]. The Henyey-Greenstein phase function has become the *de-facto* phase function for biological tissue. This is due to the phase functions relative simplicity and due to it being regarded as a “good” phase function for accurately modelling scattering in biological tissue [34]. The Henyey-Greenstein phase function is shown in Eq. [\(1.20\)](#):

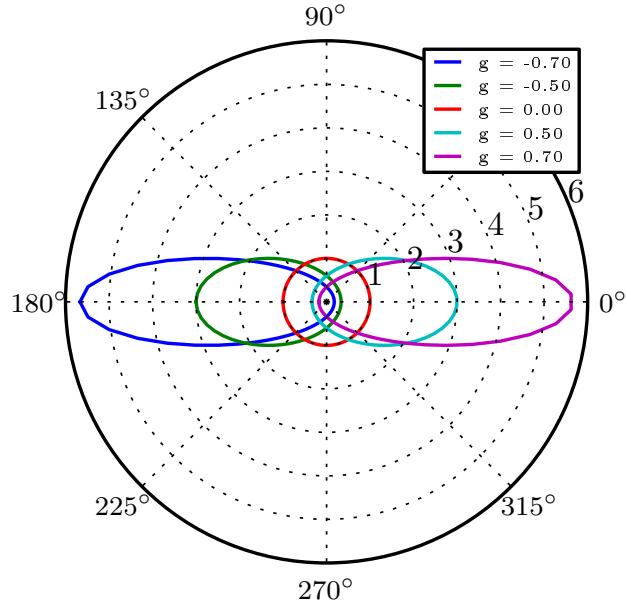


Figure 1.6: Figure show the g factor for the Henyey-Greenstein phase function, for various configurations of back, forward or isotropic scattering.

$$\Phi_{H.G}(\theta, \phi) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g \cos(\theta))^{\frac{3}{2}}} \quad (1.20)$$

Absorption

Absorption of light by a medium is defined by the absorption coefficient μ_a . The absorption coefficient is defined in a similar fashion to the scattering coefficient, by considering the probability of transmission without absorbing in a path length L :

$$T = e^{-\mu_a L} \quad (1.21)$$

This, again like the scattering coefficient, gives inverse distance for the unit of the absorption coefficient (and its is also usually measured in units of cm^{-1}).

There are various sources of absorbers in tissue: blood, water, fat, melanin, β -carotene, bilirubin are among the more absorbing chromophores. These chromophores can all contribute, depending on the probing wavelength, with some more absorbing than others, see Fig. 1.7.

1.2.2.1 Derived Parameters

There are also some derived parameters that are useful to use. These are the albedo and the transmission coefficient.

The transmission coefficient is defined as the sum of the scattering coefficient and the absorption coefficient:

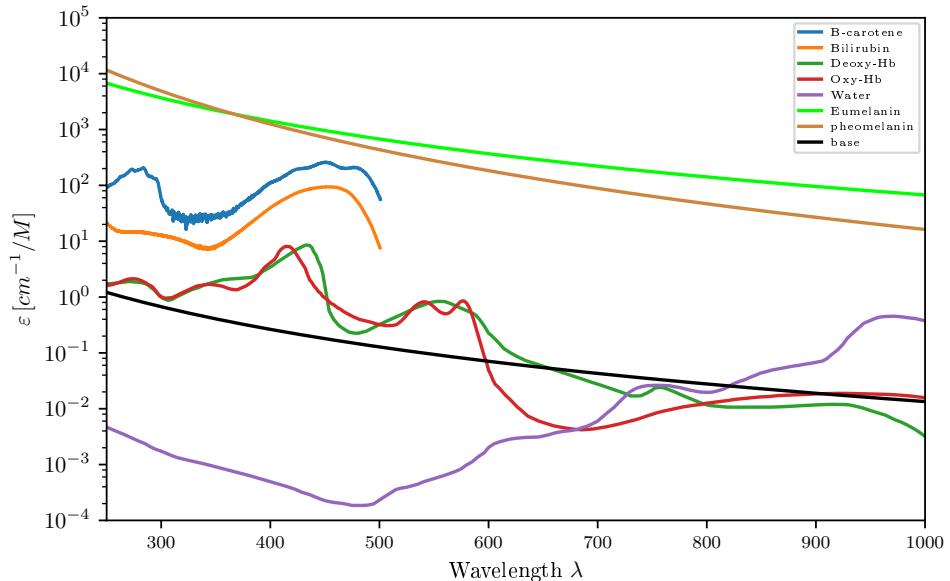


Figure 1.7: Spectra of some of the more common absorbers found in the skin [31, 35–43].

$$\mu_t = \mu_s + \mu_a \quad (1.22)$$

The albedo, or scattering probability, is defined as the ratio of the scattering coefficient to the transmission coefficient:

$$a = \frac{\mu_s}{\mu_a + \mu_s} = \frac{\mu_s}{\mu_t} \quad (1.23)$$

Other Parameters

The preceding subsection, described the optical properties that this thesis will use in every chapter. However there are other optical properties that can be used to define a medium. These other parameters generally are used to model microphysics such as Raman scattering, polarization, fluorescence or reflection/refraction. This section will give a brief overview of these other optical properties.

Refractive Index

The refractive index of a medium, defines how fast light propagates through that medium. Generally, for tissue, the refractive index is given as a bulk refractive index. Meaning that the medium is divided into sections, with each section given a refractive index. For example, skin's refractive indices are divided up by the different layers of skin. Details on how refraction is implemented with the code can be found in ??.

Raman Scattering

Raman scattering is where a photon is scattered inelastically, which excites the molecule the photon scattered off, thus decreasing the energy of the photon and increasing the photons wavelength. The optical property needed to model Raman scattering is the Raman scattering cross section. The cross section, like the absorption or scattering coefficient, is the likelihood of a photon undergoing a Raman scattering event. Raman scattering has been modelled in MCRT in order to simulate spatially offset Raman spectroscopy for breast tumour analysis [44].

Fluorescence

Fluorescence occurs when a photon is absorbed by a fluorescent molecule and re-emitted with a new wavelength. Fluorescence is a reactively common phenomena, and is heavily utilised in biophotonics and medicine, in order to image, or monitor molecules in tissue. Again the optical property that models fluorescence is coefficient that gives the probability of absorption and re-emission of a photon by a certain molecule. Usually this is in the form of an absorption coefficient or extinction coefficient. The extinction coefficient is a measurement of absorption in terms of the concentration of that absorber. Thus if a medium has many fluorophores, then the total absorption coefficient is the bulk absorption of the medium plus the contribution from the fluorophores as in Eq. (1.24):

$$\mu_a = \ln 10 \sum_i C_i \varepsilon_i \quad (1.24)$$

Where C_i is the concentration of the i^{th} fluorophore, and ε_i is the extinction coefficient of the i^{th} fluorophore.

Fluorescence will be described in more depth in ????.

1.2.3 MCRT Algorithm

A MCRT algorithm can range in complexity from a simple 20 line program, to tens of thousands of lines, depending on the application. This section will provide an in depth description of the MCRT algorithm for the propagating photons thorough a spherical medium with optical properties μ_s , and μ_a . The section following this one, will provide details of how the MCRT algorithm is implemented in the Fortran programming language, along with the various code details, such as the parallelisation of the code.

Medium and Grid Set-up

The first step of any MCRT algorithm, is to set-up the medium the photons will propagate through. There are a variety of ways that the medium can be set-up, for this section, we will assume the medium is an isotropic sphere, radius R , and centred at the origin. For simplicity we will only consider one wavelength, λ . As the MCRT algorithm presented here is run on a 3D Cartesian grid, the grid is setup before creating the spherical medium. The grid is composed of $n_x \times n_y \times n_z$ voxels[†], where each voxel can have its own optical properties. The grid is setup by first setting an array that stores the locations of the voxel boundary walls in the x , y , and z directions. The next step is to setup the actual medium. This is achieved by discretising the medium onto a grid. For this example a sphere is inscribed into a cubic volume, by setting the optical properties of a voxel to that of the medium if the sphere encloses that voxel. The voxels out with sphere are set to that of the ambient medium. An example of a voxelised medium can be seen in Fig. 1.9.

Photon Launch and Initialisation

The second step in the MCRT algorithm, is to initialise the photon. Initialisation of the photon involves setting its initial position and direction. Again how this is done depends on the experiments being simulated. Here we initialise the photon to the centre of the sphere. The initial direction is sampled isotropically, and set accordingly:

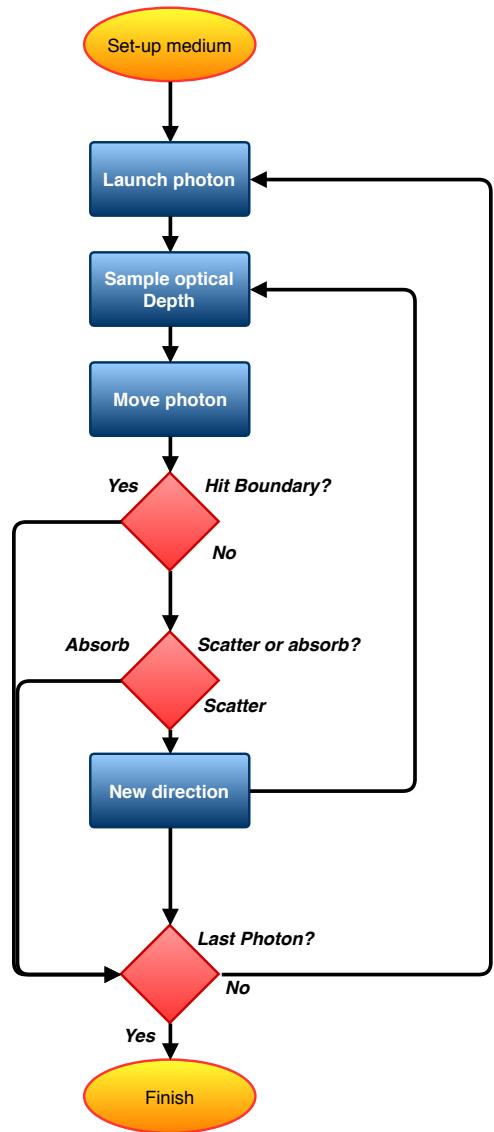


Figure 1.8: Flowchart of the Monte Carlo radiation transport algorithm as described in this section.

[†]A voxel is a 3D pixel

$$n_x = \sin(\theta) \cdot \cos(\phi) \quad (1.25)$$

$$n_y = \sin(\theta) \cdot \sin(\phi) \quad (1.26)$$

$$n_z = \cos(\theta) \quad (1.27)$$

With θ and ϕ sampled uniformly between $[0, \cos^{-1}(2\xi - 1)]$ and $[0, 2\pi\xi]$ respectively.

The next step is to launch a photon packet. Depending on the source of photon packets for a given simulation, this step varies from simulation to simulation. The general idea of launching a photon packet is that the packet is given an initial direction vector and position (which consists of a physical position and a voxel position)[‡]:

$$\text{direction} = \begin{bmatrix} n_{xp} \\ n_{yp} \\ n_{zp} \end{bmatrix} \quad (1.28)$$

$$\text{position} = [x_p, y_p, z_p] \quad (1.29)$$

$$\text{voxel} = [x_{cell}, y_{cell}, z_{cell}] \quad (1.30)$$

To set the direction vectors, the components of the direction vectors must be first set. The packets position is tracked using a Cartesian coordinate system, however for ease of computation for calculating scattering angles (see [Photon Interaction Event](#)), the direction vectors are computed in a spherical system thus the direction vectors are in Eqs. (1.25) to (1.27).

θ and ϕ are generated dependant on the photon source used. The individual sine and cosine terms are saved for use in the scattering routines, see [Photon Interaction Event](#).

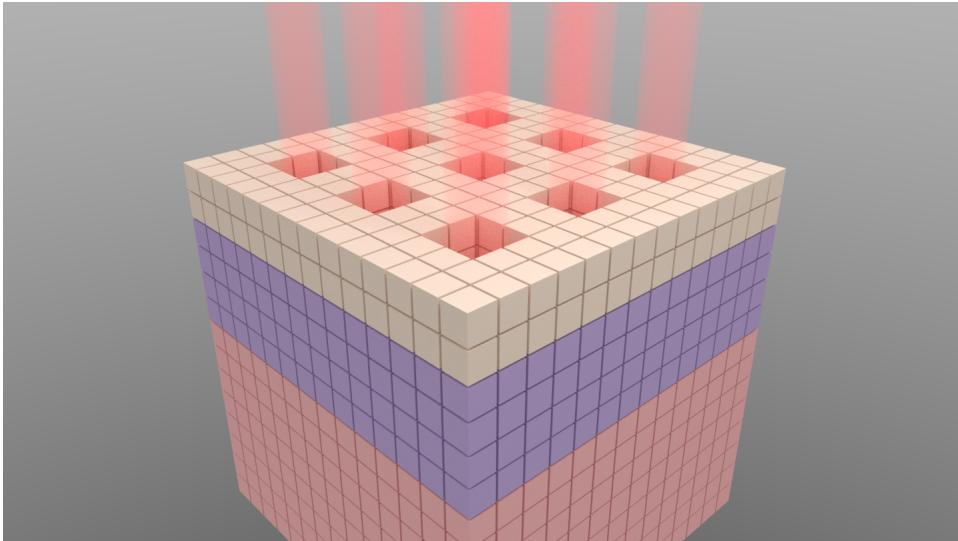


Figure 1.9: Example of a possible voxel model, with three different layers, various holes due to ablative pixel beam lasers (see ??). Each voxel can represent a different optical/thermal property of the tissue medium.

[‡]all variables given in this section are the same as they are in the code.

Photon Move

The next step in the algorithm is moving a packet to the next interaction point. The probability a packet will interact over a distance dL is $\mu_t dL$, where μ_t is the interaction probability (see [Optical Properties](#)). Thus, the probability of travelling dL without any interaction is $1 - \mu_t dL$. Therefore over a distance L , with N segments of length L/N the probability of travelling L before any interaction:

$$P(L) = (1 - \mu_t \frac{L}{N}) \cdot (1 - \mu_t \frac{L}{N}) \dots (1 - \mu_t \frac{L}{N}) = (1 - \mu_t \frac{L}{N})^N \quad (1.31)$$

$$P(L) = \lim_{N \rightarrow \infty} (1 - \mu_t \frac{L}{N})^N = e^{-\mu_t L} = e^{-\tau} \quad (1.32)$$

Where τ is the number of mean free paths over a distance L . We now have a PDF, Eq. (1.32), for the distance a packet will travel before an interaction occurs. For this to be of use we need to be able to sample from this PDF in order to get a random optical depth. Using the Monte Carlo method described in [Section 1.1](#), with ξ as our random variable, we get:

$$\xi = \int_0^\tau e^{-\tau'} = 1 - e^{-\tau} \rightarrow \tau = -\log(1 - \xi) \quad (1.33)$$

As ξ is symmetric about 0.5 we can substitute $1 - \xi$ for ξ yielding:

$$\tau = -\log(\xi) \quad (1.34)$$

We now have an optical distance, however we need to convert this into a physical distance so that we can move our photon packet. From our definition of τ we know that $\tau = \int_0^L \mu_t dS$, and if we have a smooth, homogeneous medium (i.e not a gridded medium) thus

$$L = \frac{\tau}{\mu_t} \quad (1.35)$$

Therefore in order to update the packets position is simply:

$$x_p = x_p + L \cdot n_{xp} \quad (1.36)$$

$$y_p = y_p + L \cdot n_{yp} \quad (1.37)$$

$$z_p = z_p + L \cdot n_{zp} \quad (1.38)$$

However as the code in this thesis is a 3D gridded Cartesian code, we have to slightly adjust how we move and update the packets position. As stated in [Medium and Grid Set-up](#), the medium has been discretised onto a grid, so that each voxel can have a different μ_t , thus Eq. (1.35) becomes:

$$L = \frac{\tau}{\mu_{t,\zeta}} \quad \zeta = (x, y, z) \quad (1.39)$$

with $\mu_{t,\zeta}$ the μ_t for the ζ^{th} voxel.

Moving the photon through a voxelised medium is more involved than propagating a photon through a non voxelised medium. This is mainly due to the fact the “book keeping” needed in tracking where the photon is and in what voxel it is in. The first step of moving the photon through a voxelised medium is drawing a random optical depth. This optical depth will be the full optical depth the photon travels before an interaction event. The generation of a random

optical depth is as outlined above. As the photon travels through the voxel grid, a running total of the current optical distance travelled is kept. This is then compared to the randomly generated optical depth. When the running total optical depth equals the randomly generated optical depth the photon propagation is stopped, and the photon undergoes an interaction.

The next step is to calculate the distance to the nearest wall in the x , y , and z directions. The distance is calculated for each direction. Equation (1.40) shows for the x direction:

$$d_x = \frac{x_{face} - x_{cur}}{n_{xp}} \quad (1.40)$$

Where d_x is the distance to the nearest wall in the x direction. x_{face} is an array of wall positions in the x direction, and n_{xp} is the x direction vector. With three distances calculated, $[d_x, d_y, d_z]$, the minimum of these is thus the distance to the nearest voxel wall.

The next step is to calculate the optical depth for this distance. The optical depth is calculated by rearranging Eq. (1.39) for τ , with L now the distance to the nearest wall. With the optical distance to the nearest wall calculated, the next step is to determine if there is “enough” optical distance left to travel the full distance to the nearest wall. Therefore the running total optical distance is compared to the randomly generated optical distance. If the running total + the new optical distance to the nearest wall, is less than the randomly generated optical depth, then the photon travels to the nearest wall. The photon is then placed in the next voxel by a distance δ , where δ is just larger than machine precision. If the running total + the new optical distance to the nearest wall is greater than the generated optical distance then an interaction event occurs in the current voxel. The distance to the interaction event is calculated and the photon moved to this location.

Figure 1.10 illustrates this whole process for a 2D example.

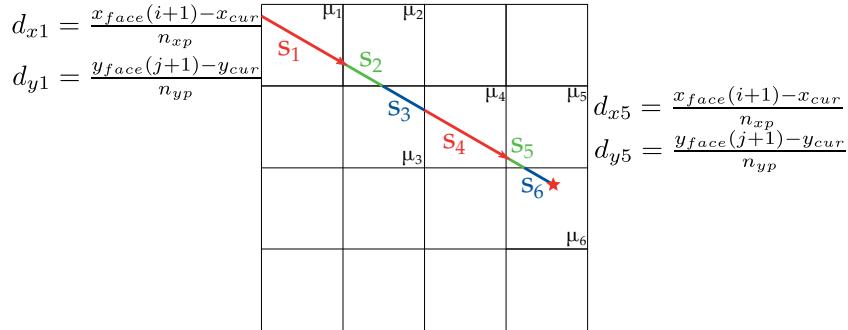


Figure 1.10: Illustration of photon propagation through a 2D grid. d_{x1} , and d_{y1} are the distances to the voxel walls in the x and y directions in the μ_1 voxel. In this case $S_1 = d_{x1}$ as d_{x1} is smaller than d_{y1} , thus the photon hits the voxel wall in the x direction. For the μ_5 voxel, d_y is smaller, thus the photon hits the voxel wall in the y^{th} direction.

This whole process is repeated until the photon undergoes an interaction event or leaves the voxel medium. The next step in the algorithm is the interaction event, which can consist of either: scattering, absorbing or another microphysics phenomena.

Photon Interaction Event

The next section of the algorithm is to decide how the photon interacts with the medium, either via scattering or absorption. There are other interaction events that can occur, however these are left for the chapters that detail these behaviours.

To decide whether a packet scatters or absorbs involves “throwing” a random number and comparing it against the albedo. The random number is compared to the albedo, and if the random number is less than the albedo then the packet scatters, otherwise the packet is absorbed.

Packet Absorption

If the interaction event is a photon packet absorption, then the algorithm terminates the photon packets and starts the next photon packet, see [Termination](#).

Packet Scattering

If the interaction event is a packet scattering, then the packet is scattered into a new direction and the above process are carried out until a termination clause is met, see [Termination](#).

Depending on the medium being simulated, it can either be isotropically scattering or anisotropic scattering. For the isotropic case, new θ and ϕ angles are sampled uniformly, and the direction vectors set as in section [Photon Launch and Initialisation](#). For the case where the scattering is anisotropic the calculation of the scattering angles, θ and ϕ , is more complicated. The random sampling of the scattering angles, θ and ϕ , are valid in the “centre of mass” frame containing the scatter, incident and scattered particle. The photons position is updated in the lab frame, thus the direction vectors also have to be updated in the lab frame. This means that the scattering angles need to be rotated into the lab frame. For the isotropic case we assume that the scattering is also isotropic in the lab frame, thus the new direction vector is easily calculated. However this is not the case for anisotropic scattering, as the centre of mass frame has to be rotated into the lab frame.

Figure 1.11 and Eq. (1.41) show how this process is achieved. Where $\mathbf{n} = (n_x, n_y, n_z)$, $\mathbf{n}_s = (n_x^{new}, n_y^{new}, n_z^{new})$, θ_s is chosen from the phase function Eq. (1.42), and $\varphi_s = 2\pi\xi$ with ξ being a random number in the range 0 to 1.

$$\begin{aligned} n_x^{new} &= \frac{\sin\theta_s}{\sin\theta} (n_x n_y \cos\varphi_s - n_y \sin\phi_s) + n_x \cos\theta_s \\ n_y^{new} &= \frac{\sin\theta_s}{\sin\theta} (n_y n_z \cos\varphi_s + n_x \sin\phi_s) + n_y \cos\theta_s \\ n_z^{new} &= -\sin\theta_s \cos\varphi_s + n_z \cos\theta_s \end{aligned} \quad (1.41)$$

$$\cos\theta = \frac{1 + g^2 - \left(\frac{1-g^2}{(1-g+2g\xi)^{3/2}}\right)^2}{2g} \quad (1.42)$$

Termination

The final section of the MCRT algorithm is to check if it should be terminated. This is a simple check to see if there are any more photons to run. If there are more photons to run then the algorithm goes back to the [Photon Launch and Initialisation](#) section and continues from there. If there are no more photons the algorithm terminates and any results are written out.

Scored Quantities

As MCRT is a computational method, we are able to record a wealth of information during the simulation. From the paths of individual photons, to average scattering angles and more.

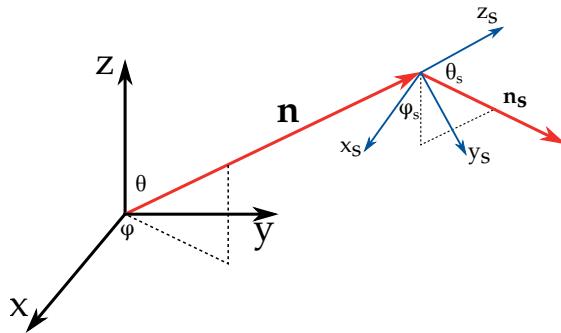


Figure 1.11: Illustration of rotating the the centre of mass frame to the lab frame. \mathbf{n} is the direction vector of the photon before scattering, and \mathbf{n}_s is the scattered direction vector. θ and φ are the scattering angles. z_s is in the same direction as \mathbf{n} .

However it is not practical to record all this information for every simulation, as this would lead to inefficient simulations, and expensive data storage solutions. Thus for a given problem we only store the pertinent information.

One important recorded variable is fluence. Fluence is the number of photons entering a sphere per unit cross section area [2]. In practise the average fluence per area is used, Eq. (1.43), as this is easier to calculate in an MCRT code. Lucy showed that the average fluence per area is proportional to the sum of the path length through a volume [45]:

$$J_i = \frac{L}{NV_\xi} \sum l \quad (1.43)$$

Where:

J_i is the fluence or mean intensity [$W\ m^{-2}$];

L is the luminosity or power of the light source [W];

N is the total number of photon packets [-];

V_ξ is the volume of the ξ^{th} voxel [m^3];

l is the path length of a photon packet through the ξ^{th} voxel [m].

The majority of the chapters in the thesis make use of Eq. (1.43) or modified versions of it as the main scored quantity.

Other common scored quantities are the exit location of a photon, the wavelength of an exiting photon or the distribution of photon packet absorption.

1.2.4 Code Details

The preceding section gave an overview of the algorithm and how it works. This section details the various code and implementation details such as how the code is parallelised.

Code

All code in this thesis is written in modern Fortran[§]. All subroutines and functions are contained in modules (with the exception of the main program—mc polar.f90). This is done in order to be able to “hide” data from subroutines and functions, and to arrange the code that relates to other parts of the code in the same file. Having the code in modules also allows the use of runtime

[§]modern Fortran is considered anything past Fortran 95 [46]

allocation of memory for arrays. This enables the user to specify the size of arrays depending on the need of the user for the problem at hand.

Modules are classified into three different modules types: data, routines and dependencies. Data modules are modules that contain no function or routines, but store variables that can be accessed anywhere in the program when required. Routine modules contain the subroutines and functions used in the code. Finally dependency modules are the modules that have not been written by me, and thus the code depends upon them in order to run.

Figure 1.12 show the relationship between the various modules, for a basic version of the MCRT as described in [MCRT Algorithm](#).

Using Fig. 1.12 as a reference each module contains:

`mcpolar.f90` is the entry point of the code. It calls all other subroutines and functions, as well as setting up various variables and printing progress.

`ch_opt` is the module where the optical properties are set or changed.

`gridset_mod` is where the optical properties grid and voxel walls are set.

`subs` contains general purpose routines that are used in various different parts of the code.

`writer_mod` contains routines that write out the results of the simulation.

`inttau2` is the module that contains the routines that propagate the photon through the voxel grid.

`sourcph_mod` contains the routines that initialise the photon position and direction.

`stokes_mod` contains the routine that calculates the scattering direction after a scattering event.

`iarray` is a data module that contains all the arrays in the code.

`constants` is a data module that contains all the constants and filepaths needed in the code.

`ieee_arithmetic` is an external dependency that gives various arithmetic checking routines such as `is_nan()`.

`vector_class` is a module that contains the vector type, and all its associated operation such as cross and dot products.

`photon_vars` is a data module that contains the data pertaining to each photon, such as wavelength or energy.

Finally, `opt_prop` contains the data about the current optical properties such as the albedo, and absorption coefficient.

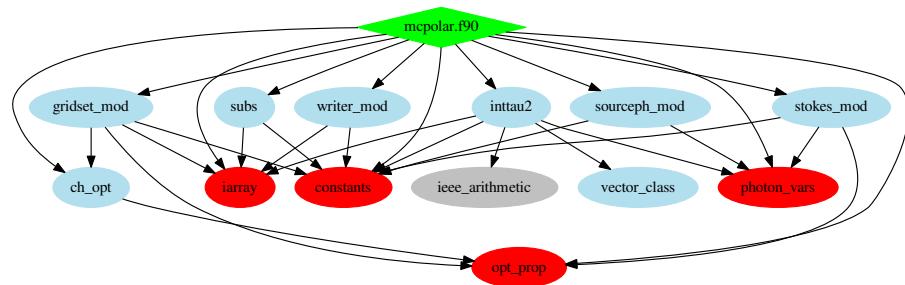


Figure 1.12: Source code hierarchy, showing the relationship between different modules. Green is the entry point for the simulation. Red are the data modules, light blue are the routine modules, and grey are the external dependencies.

Parallelisation of the MCRT Algorithm

As mentioned in the previous sections, MCRT can be computationally intensive, especially when dealing with highly scattering mediums. Fluorescence can also cause simulations times to drastically increase as photons are no longer ‘killed’ off, but rather re-emitted at a new wavelength. Other optical processes such as Raman scattering are highly unlikely events, which again can lead to a dramatic increase in simulation times, as many photons are required to be simulate in order to get ‘good’ statistics.

Fortunately MCRT is classed as an “embarrassingly parallel” problem[¶]. This means that it is trivial to parallelise in comparison to other algorithms. The reason that MCRT is classed as ‘embarrassingly parallel’, is that the algorithm can be split up onto separate processes, with little need for communication between them. In reality this means that n copies of the algorithm can run on n cores in a processor, with communication taking place at the start and end of each simulation run.

All the code in this thesis is parallelised using Message-passing interface (MPI) [49, 50], with the only communication taking place at the end, where the path length estimators are collated on to all process. The one exception to the is in ??, where the heat diffusion calculation needs communication between the process during the calculation.

The parallel efficiency of a code depends on the problem, and the number of photon packets run. To determine the speedup of a given problem Amdahl’s law is used [51]:

$$speedup = \frac{1}{(1 - P) + P/N} \quad (1.44)$$

Where P is the fraction of the code that is parallel, and N is the number of cores the code is run on. The consequence of Amdahl’s law is as N tends to infinity the speedup tends to a maximum:

$$speedup_{max} = \frac{1}{1 - P} \quad (1.45)$$

The value of P varies from problem to problem, and the number of photon packets run. Figure 1.13 shows the results of the profiling of the code, for various numbers of cores. This test consisted of running the same number of photons, in a highly scattering medium of size 2 cm^3 . This yielded a P of 0.999010 ± 0.000045 , and a maximum speedup of 1010.1

There are other ways the code could be parallelised, including task farms and domain decomposition [52]. However these methods are more involved to set up and validate, so the simplest approach was taken.

[¶]However this is not true for all MCRT applications. For example, using the Bjorkman & Wood [47] immediate temperature corrections method, turns MCRT into a different class of parallel problem [48].

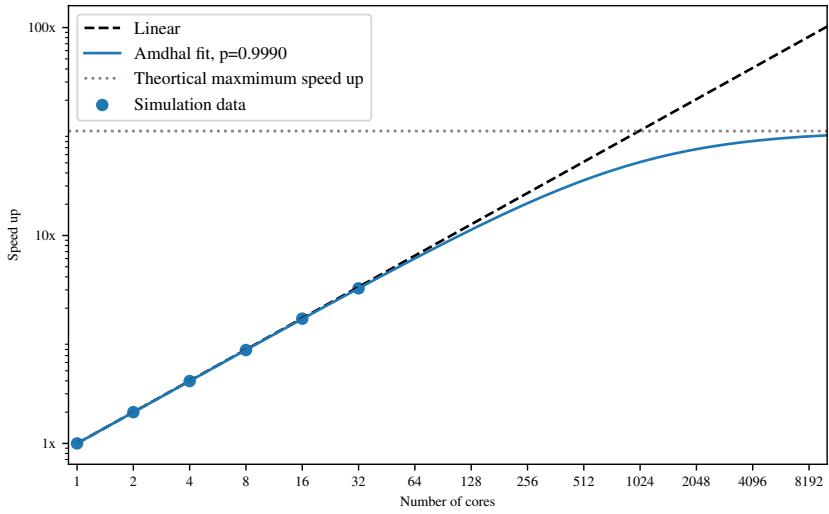


Figure 1.13: Performance of the parallelisation of the MCRT code using MPI.

1.3 Validation of MCRT Code

As the Monte Carlo method is an algorithm that depends upon random numbers, it is sometimes hard to ensure the correct result is obtained. Or to put it another way:

“Monte Carlo is easy to do wrong!” G.W. Collins III [53]

Thus the code has to be validated against various theoretical/experimental and other simulations, to determine whether the results are correct.

For the first test of a MCRT code, it is compared against a experimental expression, Eq. (1.46), for fluence as a function of depth [54]. This expression has also been matched by other MCRT simulations [55].

$$\Psi(z) = \Psi_0(C_1 e^{-k_1 z/\delta} - C_2 e^{-k_2 z/\delta}) \quad (1.46)$$

Where:

$\Psi(z)$ is the penetration of the excitation light, or equivalently the fluence rate [$W\text{ cm}^{-2}$];

Ψ_0 is a normalisation constant [$W\text{ cm}^{-2}$];

C_n and k_n are fitted coefficients [-];

and δ is the optical penetration depth, defined as $\delta = 1/\sqrt{3\mu_a(\mu_a + \mu_s(1-g))}$, [cm].

Jacques *et al.*, in their simulation used two different wavelengths, 420 nm and 630 nm. The medium in the simulation is a infinitely wide slab with a depth of 2 cm, with uniform optical properties. The medium has a refractive index of 1.38. The g value is in the range 0.7 – 0.9, and the optical properties are as in Table 1.1.

Using these values Jacques *et al.* calculated values for C_1 , C_2 , k_1 and k_2 using their MCRT code. The above optical properties and medium dimensions^{||} are recreated in the code and a value of 0.9 was chosen for g . 8 million photons were run for the simulation. This yielded the result as in Fig. 1.14.

^{||}The infinitely wide slab is implemented so that when a photon leaves the one of the sides of the voxel grid, it is moved to the other side of the grid, retaining its original direction vectors.

Wavelength/nm	Absorption μ_a/cm^{-1}	Scattering $\mu_s(1-g)/cm^{-1}$	Penetration			δ/cm	
	C1	k1	C2	k2			
420	1.8	82	5.76	1.00	1.31	10.2	0.047
630	0.23	21	6.27	1.00	1.18	14.4	0.261

Table 1.1: Table of optical properties and determined coefficients from Jacques *et al.* [55].

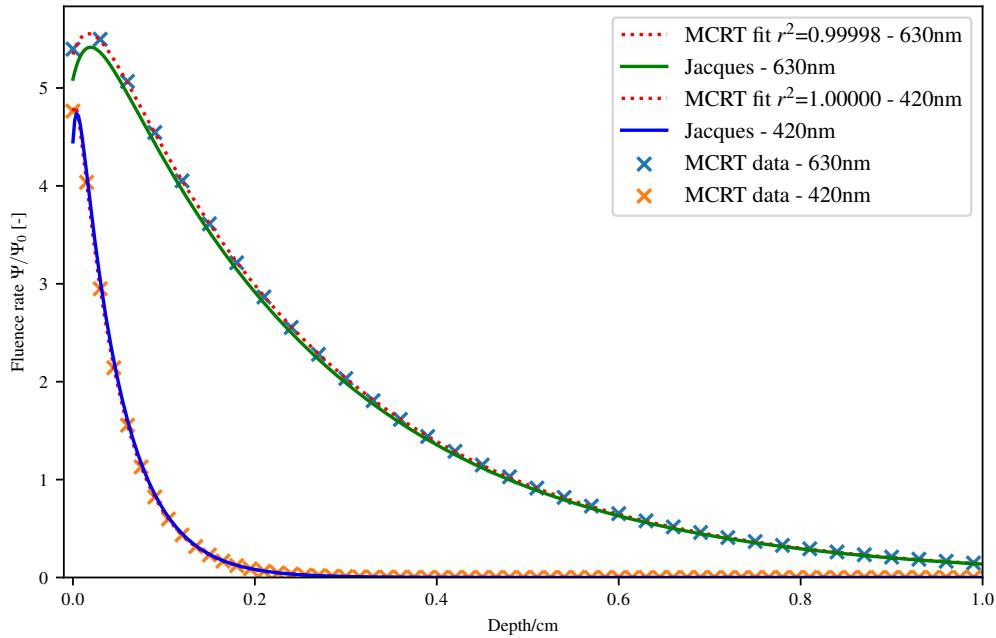


Figure 1.14: Figure shows the fluence as a function of depth. Figure also shows comparison to the Jacques MCRT simulation and the MCRT as described in this chapter.

Fitting Eq. (1.46) to the data calculated by our MCRT code for 630 nm, gave: $C_1 = 6.425$, $C_2 = 1.083$, $k_1 = 1.0$, and $k_2 = 12.966$. For 420 nm gave: $C_1 = 5.600$, $C_2 = 0.838$, $k_1 = 1.003$, and $k_2 = 9.846$. These are in good agreement with Jacques *et al.* results.

1.4 Conclusion

The Monte Carlo method relies on drawing random numbers in order to calculate unknown variables in problems. The Monte Carlo method can be used to calculate radiation transport in order to numerically solve the RTE. MCRT is the most flexible of the methods available to solve the RTE, allowing arbitrary geometries, light sources and turbid mediums. MCRT also allows the inclusion of various microphysics such as polarisation, Raman scattering and fluorescence. However the MCRT method does have the downside, that for some problems, many photon packets must be run in order to achieve a good signal to noise ratio. Though this is becoming less of an issue due to increased computational available capacity.

The algorithm covered in this chapter is the underpinning algorithm for this whole thesis. Each subsequent chapter builds upon the algorithm in order to solve the problem at hand.

Bibliography

- [1] Edmond D Cashwell and Cornelius Joseph Everett. A practical manual on the monte carlo method for random walk problems. 1959.
- [2] DWO Rogers and AF Bielajew. Monte carlo techniques of electron and photon transport for radiation dosimetry. *The dosimetry of ionizing radiation*, 3:427–539, 1990.
- [3] Lee Badger. Lazzarini’s lucky approximation of π . *Mathematics Magazine*, 67(2):83–91, 1994.
- [4] Petr Beckmann. *A history of Pi*. St. Martin’s Griffin, 2015.
- [5] Georges-Louis Leclerc Buffon. *Histoire naturelle générale et particulière*, volume 18. de l’Imprimerie de F. Dufart, 1785.
- [6] Peter Jäckel. *Monte Carlo methods in finance*. J. Wiley, 2002.
- [7] David B Hertz. Risk analysis in capital investment. *Harvard Business Review*, 42(1):95–106, 1964.
- [8] Jasper Vivian Wall and Charles R Jenkins. *Practical statistics for astronomers*. Cambridge University Press, 2012.
- [9] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, August 1986.
- [10] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *SIGGRAPH Comput. Graph.*, 18(3):137–145, January 1984.
- [11] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.
- [12] Nicholas Metropolis. The beginning of the Monte Carlo method. *Los Alamos Science*, 15:125–130, 1987.
- [13] Roger Eckhardt. Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science*, 15:131–136, 1987.
- [14] HL Anderson. Metropolis, Monte Carlo, and the MANIAC. *Los Alamos Science*, 14:96–108, 1986.
- [15] Stanislaw Ulam, RD Richtmyer, and J Von Neumann. Statistical methods in neutron diffusion. *LAMS-551, Los Alamos National Laboratory*, pages 1–22, 1947.

- [16] Kenneth Wood and RJ Reynolds. A model for the scattered light contribution and polarization of the diffuse $\text{H}\alpha$ galactic background. *The Astrophysical Journal*, 525(2):799, 1999.
- [17] DWO Rogers, BA Faddegon, GX Ding, C-M Ma, J We, and TR Mackie. Beam: a monte carlo code to simulate radiotherapy treatment units. *Medical physics*, 22(5):503–524, 1995.
- [18] BC Wilson and G Adam. A monte carlo model for the absorption and flux distributions of light in tissue. *Medical Physics*, 10(6):824–830, 1983.
- [19] Lihong V Wang and Hsin-i Wu. *Biomedical optics: principles and imaging*. John Wiley & Sons, 2012.
- [20] Subrahmanyam Chandrasekhar. *Radiative transfer*. Courier Corporation, 2013.
- [21] Vesna Džimbeg-Malčić Željka Barbarić-Mikočević and Katarina Itrić. Kubelka-munk theory in describing optical properties of paper (i). *Technical Gazette*, 18(1):117–124, 2011.
- [22] Marek Jasiński. Modelling of light and human skin interaction using kubelka-munk theory. *Scientific Research of the Institute of Mathematics and Computer Science*, 10(1):71–81, 2011.
- [23] Wai-Fung Cheong, Scott A Prahl, and Ashley J Welch. A review of the optical properties of biological tissues. *IEEE journal of quantum electronics*, 26(12):2166–2185, 1990.
- [24] Macaveiu Gabriela. Mathematical methods in biomedical optics. *ISRN Biomedical Engineering*, 2013, 2013.
- [25] Scott Alan Prahl. Light transport in tissue. 1990.
- [26] Reindert Graaff, Jan G Aarnoudse, Frits FM de Mul, and Henk W Jentink. Similarity relations for anisotropic scattering in absorbing media. *Optical engineering*, 32(2):244–253, 1993.
- [27] Gilwon Yoon, Scott A Prahl, and Ashley J Welch. Accuracies of the diffusion approximation and its similarity relations for laser irradiated biological media. *Applied Optics*, 28(12):2250–2255, 1989.
- [28] Thomas P Robitaille. On the modified random walk algorithm for monte-carlo radiation transfer. *Astronomy & Astrophysics*, 520:A70, 2010.
- [29] M Min, CP Dullemond, C Dominik, Alex de Koter, and JW Hovenier. Radiative transfer in very optically thick circumstellar disks. *Astronomy & Astrophysics*, 497(1):155–166, 2009.
- [30] Steven L Jacques. Origins of tissue optical properties in the uva, visible, and nir regions. *OSA TOPS on advances in optical imaging and photon migration*, 2:364–369, 1996.
- [31] Steven L Jacques. Optical properties of biological tissues: a review. *Physics in Medicine & Biology*, 58(11):R37, 2013.
- [32] Tom Lister, Philip A Wright, and Paul H Chappell. Optical properties of human skin. *Journal of biomedical optics*, 17(9):090901, 2012.
- [33] Louis G Henyey and Jesse L Greenstein. Diffuse radiation in the galaxy. *The Astrophysical Journal*, 93:70–83, 1941.

- [34] Steven L Jacques, CA Alter, and Scott A Prahl. Angular dependence of hene laser light scattering by human dermis. *Lasers Life Sci*, 1(4):309–333, 1987.
- [35] James M Dixon, Masahiko Taniguchi, and Jonathan S Lindsey. Photochemcad 2: A refined program with accompanying spectral databases for photochemical calculations. *Photochemistry and photobiology*, 81(1):212–213, 2005.
- [36] Scott Prahl. Photochemcad spectra. <https://omlc.org/spectra/PhotochemCAD/index.html>, 2017. [Online; Last accessed 4-February-2019].
- [37] D.J. Segelstein. *The complex refractive index of water*. PhD thesis, University of Missouri-Kansas City, 1981.
- [38] Robin M Pope and Edward S Fry. Absorption spectrum (380–700 nm) of pure water. ii. integrating cavity measurements. *Applied optics*, 36(33):8710–8723, 1997.
- [39] Robert LP van Veen, HJCM Sterenborg, A Pifferi, A Torricelli, and R Cubeddu. Determination of vis-nir absorption coefficients of mammalian fat, with time-and spatially resolved diffuse reflectance and transmission spectroscopy. In *Biomedical Topical Meeting*, page SF4. Optical Society of America, 2004.
- [40] Iyad Salam Saidi et al. *Transcutaneous optical measurement of hyperbilirubinemia in neonates*. PhD thesis, Rice University, 1992.
- [41] Jose A Iglesias-Guitian, Carlos Aliaga, Adrian Jarabo, and Diego Gutierrez. A biophysically-based model of the optical properties of skin aging. In *Computer Graphics Forum*, volume 34, pages 45–55. Wiley Online Library, 2015.
- [42] Alexey N Bashkatov, Elina A Genina, and Valery V Tuchin. Optical properties of skin, subcutaneous, and muscle tissues: a review. *Journal of Innovative Optical Health Sciences*, 4(01):9–38, 2011.
- [43] Tadeusz Sarna and Harold A Swartz. The physical properties of melanins. *The pigmentary system: physiology and pathophysiology*, pages 311–341, 2006.
- [44] Matthew D Keller, Robert H Wilson, Mary-Ann Mycek, and Anita Mahadevan-Jansen. Monte carlo model of spatially offset raman spectroscopy for breast tumor margin analysis. *Applied spectroscopy*, 64(6):607–614, 2010.
- [45] L.B. Lucy. Computing radiative equilibria with monte carlo techniques. *Astronomy and Astrophysics*, 344:282–288, 1999.
- [46] Michael Metcalf, John Reid, and Malcolm Cohen. *Modern Fortran Explained*. Oxford University Press, 2011.
- [47] Jon E Bjorkman and Kenneth Wood. Radiative equilibrium and temperature correction in monte carlo radiation transfer. *The Astrophysical Journal*, 554(1):615, 2001.
- [48] Thomas P Robitaille. Hyperion: an open-source parallelized three-dimensional dust continuum radiative transfer code. *Astronomy & Astrophysics*, 536:A79, 2011.
- [49] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. Scientific and Engineering Computation. MIT Press, 2014.

- [50] William Gropp, Torsten Hoefer, Rajeev Thakur, and Ewing Lusk. *Using advanced MPI: Modern features of the message-passing interface*. MIT Press, 2014.
- [51] G.M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, pages 483–485. ACM, 1967.
- [52] Blaise Barney et al. Introduction to parallel computing.
- [53] J. Bjorkman. Monte carlo radiation transfer. Presented as SAMCSS 2013, 2013.
- [54] Craig M Gardner, Steven L Jacques, and Ashley J Welch. Fluorescence and reflectance spectra specify intrinsic fluorescence spectrum corrected for tissue optics distortion. In *Advances in Fluorescence Sensing Technology*, volume 1885, pages 122–129. International Society for Optics and Photonics, 1993.
- [55] Steven L Jacques, Rachel Joseph, and Gary Gofstein. How photobleaching affects dosimetry and fluorescence monitoring of pdt in turbid media. In *Optical Methods for Tumor Treatment and Detection: Mechanisms and Techniques in Photodynamic Therapy II*, volume 1881, pages 168–180. International Society for Optics and Photonics, 1993.