ABI Studentship Report

Deroshan Padotan

Department of Engineering Science
(Biomedical Engineering)

Auckland Bioengineering Institute

UniServices House, Level 6, 70 Symonds Street, Grafton, Auckland City

Work Period: 19/11/2012 - 27/02/2013

Date: 13/03/2013

Summary

Over the university summer break between November 2012 and February 2013 I completed a studentship at the Auckland Bioengineering Institute (ABI). The ABI is a research centre which collaborates with various faculties within the University of Auckland to apply principals of engineering and physics to research and development in the medical science fields.

During my time at the ABI the main focus of my project was to build a computer algorithm which simulated a light ray passing through a material composed of interfaces between two types of media. Using the algorithm I then had to reconstruct the path travelled by a light ray before it exited the material.

Throughout the modelling process I considered a number of different approaches to components of the algorithm such as the interface orientation, the distance between successive interfaces, interface intercept point coordinate frames and reconstruction of the light ray path. By comparing the different approaches at each stage in the modelling process I was able to gradually build up a functioning algorithm which produced realistic output. The tests runs of the final programme produced a range of between 700 and 12,000 interfaces before the light ray exited the material.

In order to extend the work done during this project experimental validation of the algorithm predictions is required. The programme could also be expanded in order to improve its representation of the physical situation.

Completing a studentship at the ABI provided me with a unique opportunity to work and gain experience in a research area with which I was unfamiliar. Over the duration of my studentship I have been able to broaden my skill set and acquire new knowledge which will be of benefit to both my final year of study and future as a professional engineer.

Acknowledgements

I would like to express my gratitude to both my supervisors Dr Turuwhenua and Dr Vorobyev for providing me with the opportunity to complete this studentship at the ABI. Their assistance and feedback throughout the project was vital to my understanding of many areas within the project which were conceptually challenging.

I would also like to thank all of the students on level 4 with whom I worked, for making my summer studentship an enjoyable and memorable experience.

Table of Contents

1.0 Introduction 1

2.0 Early Sections 2

2.1 Organisation 2

2.2 Building 2

3.0 Central Sections 3

3.1 Project Outline 3

3.2 Background 3

3.3 Assumptions 3

3.4 Algorithm Outline 4

3.5 Modelling 4

3.5.1 Interface Randomisation 5

3.5.2 Refraction and Reflection 6

3.5.3 Reflection Coefficient 6

3.5.4 Distance Between Interfaces 7

3.5.5 Vector Rotation 8

.... 9

3.6 Output 10

3.7 Future Work 11

3.8 Skills Gained 11

4.0 Conclusions 12

5.0 References **13**

1.0 Introduction

The Auckland Bioengineering Institute (ABI) is a research centre which focuses primarily on applying engineering and mathematical principals to biological systems. By employing the various experimental and modelling techniques developed in the fields of engineering and physics, the ABI strives to further our understanding of biology and human physiology as well as improve the methods of diagnosis and treatment of many diseases.

Established by the University of Auckland, the ABI has strong ties with the faculties of Engineering, Biology and Medical and Health Sciences. As well as this the ABI also has a significant amount of international collaborators.

The institute's research spans a wide range of areas including organ modelling, systems biology, computational fluid mechanics and bioinstrumentation development. Many of the research projects involve cross-faculty collaboration to form multi-disciplinary teams. This facilitates a broad range of approaches to every challenge as well as a pooling of knowledge.

2.0 Early Sections

2.1 Organisation

The ABI consists of about 180 members with over 70 of those being graduate students. A large portion of the remaining members is made up of the principal investigators and research fellows of the ABI. Many of these researchers also hold teaching positions at the University of Auckland in the faculties of Engineering, Science and Medical and Health Sciences.

The director at the ABI is highly regarded Professor Peter Hunter. Along with Professor Hunter, the management at the ABI consists of a deputy director, principal investigator group and budget monitoring group.

The principal investigators group at the ABI consists primarily of academic staff who are involved in managing the institutes research projects, supervision of graduate students as well as raising money in the form of grants for research.

2.2 Building

The ABI is situated in the UniServices building in Grafton, Auckland which is also home to the University of Auckland Engineering Science department. The building houses University of Auckland and ABI staff offices, computer labs for students as well as research and development labs utilised by the ABI. During my studentship I worked on level 4 of the UniServices house in the Biomedical Engineering computer labs.

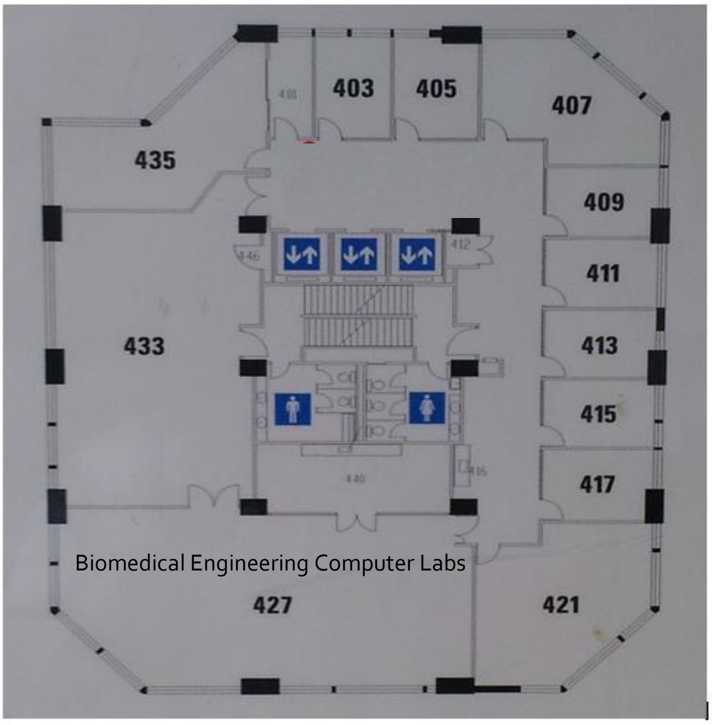

Figure 1: UniServices House level 4 floor plan

3.0 Central Sections

3.1 Project Outline

Between November 2012 and February 2013 I worked at the ABI on a studentship offered to Engineering students at the University of Auckland. My project revolved around building a computer algorithm which modelled the various refractions and reflections which constitute the path a light ray travels inside a material. Using MATLAB I gradually built up a programme which simulated a single light ray passing through a material which contained interfaces between two types of media. The programme tracked the position of the light ray until it had exited the material at which point the overall path of the light ray was reproduced.

3.2 Background

At the boundary between two different types of media light rays are both refracted and reflected. When light rays pass through a given material they experience a series of refractions and reflections as a result of the various interfaces between different media within the material. This could include the interfaces between air and material particles or between material particles of different refractive index. The refractive index of a medium is a dimensionless quantity that characterises the way in which light behaves within the medium. Generally the direction of the refracted and reflected rays is dependent on the angle of the incident rays relative to the slope of the interface as well as the refractive indices of the media. If a light ray intercepts an interface at an angle above what is known as the critical angle, then total internal reflection occurs whereby the ray is not transmitted at all through the interface but completely reflected.

3.3 Assumptions

In order to simplify the modelling of a light ray's path a few key assumptions were made:

- The orientation of each interface within the material was random over 180 degrees about the incident ray.
- The probability distribution of distances between successive interfaces followed a decaying exponential. This means that the further into the material the ray travels the more likely the distance to the next interface will be small.
- The light rays were not polarised as a result of the randomness of the interfaces
- The material through which the light ray was travelling was only composed of two different types of media.

3.4 Algorithm Outline

The overall outline of the final algorithm is summarised in the flow chart below

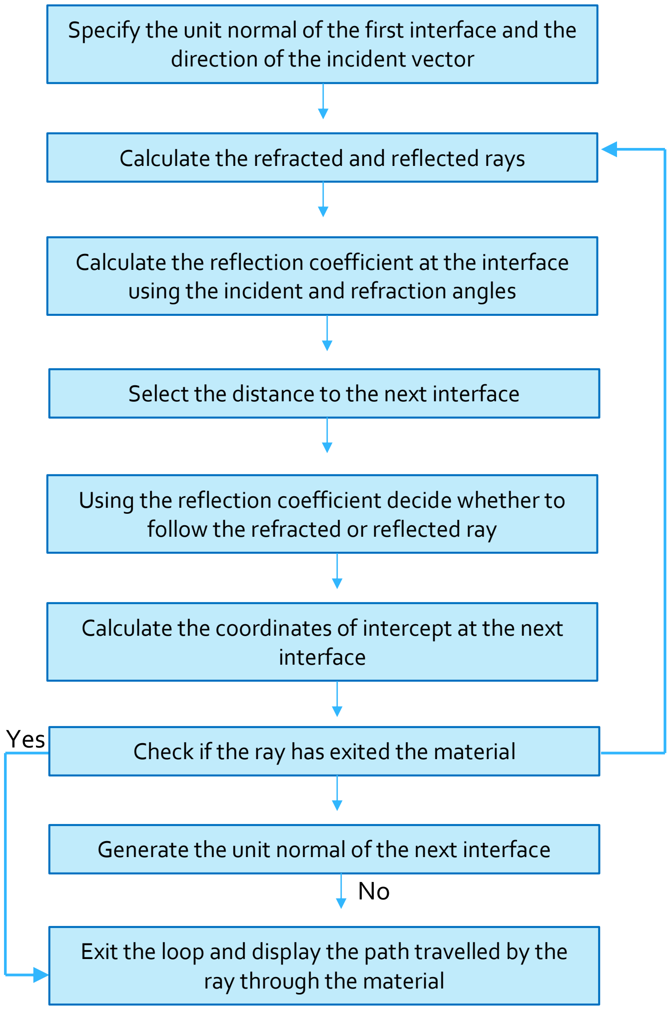

Figure 2: Algorithm summary diagram

3.5 Modelling

Within the model each light ray was represented using two vectors. The first vector described the light ray's position in space whereas the second vector determined the direction in which the light ray was travelling. This was adapted from a representation used by (Poon and Kim, 2006) It should be noted that the developed algorithm uses an altered coordinate system where the horizontal axis which is also the optical axis is in the z-direction.

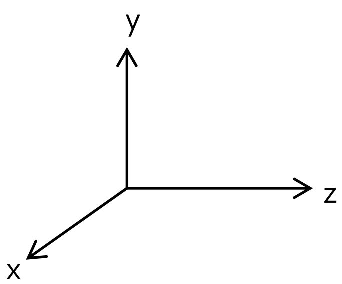

Figure 3: Coordinate System implemented in the programme

3.5.1 Interface Randomisation

One of the first things which needed to be established during the modelling process was the way in which the interfaces were going to be randomised. To randomise the slope of interfaces the normal vector of each interface was randomised over 180 degrees about the incident vector to the plane. In order for the spread of the normal vectors to be equal over the 180 degrees, they would need to be randomly selected from a uniform distribution over a hemisphere. During the development of the ray tracing code we considered going about this in several different ways. In order to choose the best method, code was written to compare the various approaches and visualize their effectiveness at choosing points on a sphere.

The methods we compared were:

1. Randomising spherical coordinate angles over defined ranges and then using the angles to calculate the coordinates of a point on the sphere
2. Randomising tangent angles over a defined ranges
3. A sphere point picking method found on Wolfram Mathworld (Weisstein, 2002).
4. A sphere point picking method developed by George Marsaglia of McGill University (Marsaglia, 1972).
5. A sphere point picking method developed by Mervin E. Muller of Princeton University which was tested with random numbers chosen from both a normal and uniform distribution (Muller, 1959).

The results of this comparison are illustrated in figure 4 below

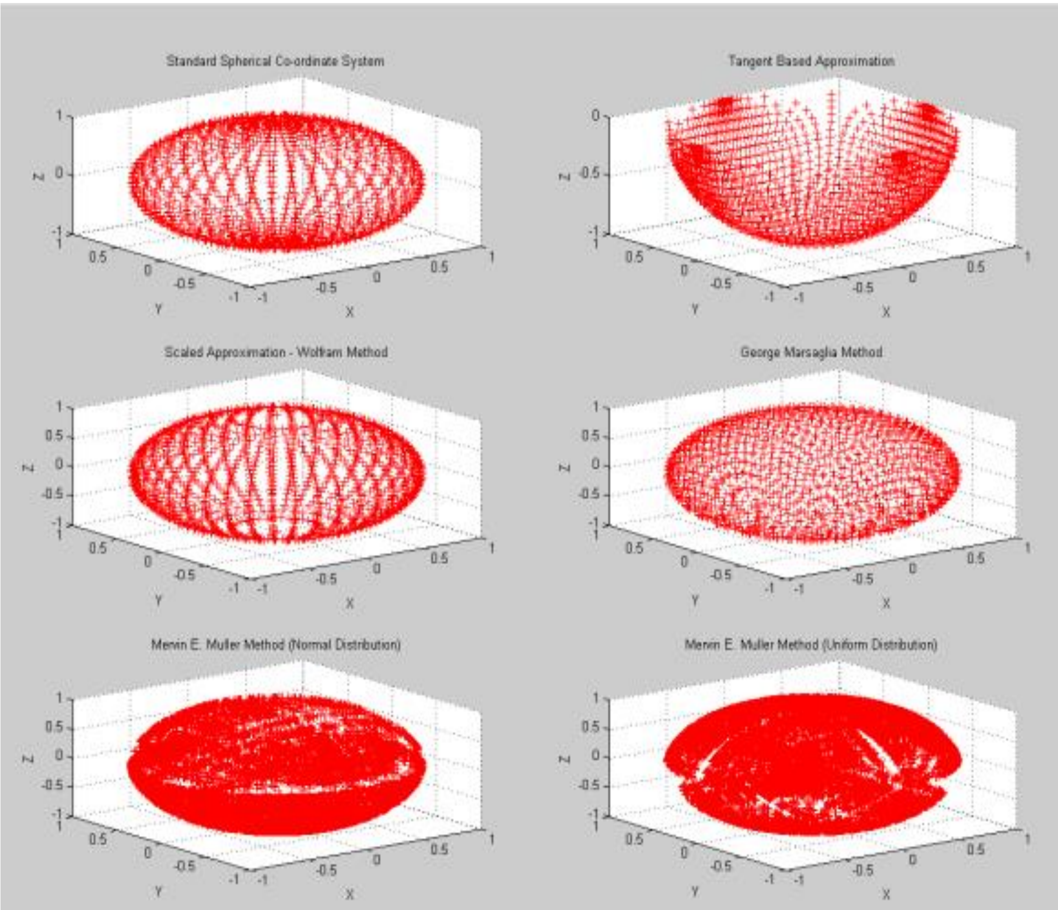

Figure 4: Comparison of sphere point picking methods

From figure 4 it can be seen that the method developed by George Marsaglia provides the closest estimate of a uniform distribution of points on a sphere. For this reason it was the chosen scheme used to randomly generate the components of the normal vector.

3.5.2 Refraction and Reflection

At each interface the direction of the refracted and reflected vectors were calculated using functions supplied by Dr Turuwhenua at the start of the project.

The refract function takes as inputs the direction of the incident ray, the direction of the normal vector at the interface and the refractive indices of the media on either side of the boundary and returns the direction of the refracted ray. When this particular refract function returns complex output it indicates that total internal reflection has occurred.

The reflect function takes as inputs only the normal vector to the interface and the direction of the incident ray and returns the direction of the reflected ray.

At each interface the incident and refracted angles were also calculated. These angles were used in two separate checks in the code. The first of these was to confirm when total internal reflection occurred. Whilst complex output from the refract function indicated total internal reflection, a check that the incident angle was greater than the critical angle was also performed in order to verify this. The second check in which the angles were used in was a validation using Snell's Law that helped to prove the refract function was returning the correct refracted vector. Snell's Law is given by the expression:

$$n_1 sin\theta_i = n_2 sin\theta_t$$

Where:

- n_1 and n_2 = Refractive indices of media 1 and 2
- θ_i = Incident angle
- θ_t = Transmitted/refracted angle

Using the incident and refracted angles, within the programme the right and left hand sides of Snell's Law were individually calculated in order to prove that they were equal to each other.

3.5.3 Reflection Coefficient

Since the algorithm developed for this project only models a single light ray, at each interface a decision needed to be made whether to follow the refracted or the reflected ray to the next interface. In order to do this in a statistically accurate way the reflection coefficient was used. The reflection coefficient provides a measure of the proportion of rays which will be reflected at a given interface or for a single ray, the probability that the ray will be reflected. For our model as previously mentioned it was assumed that the light rays were unpolarised. As a result the equation for the reflection coefficient (R) is given by the following expression derived from Fresnel's equations (Poon and Kim, 2006).

$$R = \frac{\left[\frac{n_1 cos\theta_i - n_2 cos\theta_t}{n_1 cos\theta_i + n_2 cos\theta_t}\right]^2 + \left[\frac{n_1 cos\theta_t - n_2 cos\theta_i}{n_1 cos\theta_t + n_2 cos\theta_i}\right]^2}{2}$$

The returned value of R is a value between zero and one indicating the probability a given ray will be reflected.

At each interface a random number was generated between zero and one and then compared to the reflection coefficient. If the random number generated was equal to or less then the reflection coefficient, the reflected ray would be followed or else the refracted ray would be followed. It should be noted that in the case of total internal reflection since there was no refracted ray, the reflected ray was always followed.

3.5.4 Distance Between Interfaces

In this model of light ray tracing it was assumed that the probability distribution of the distances between interfaces followed a decaying exponential. This is as a result of the fact that the longer the ray spends in the material the more likely it is to hit an interface.

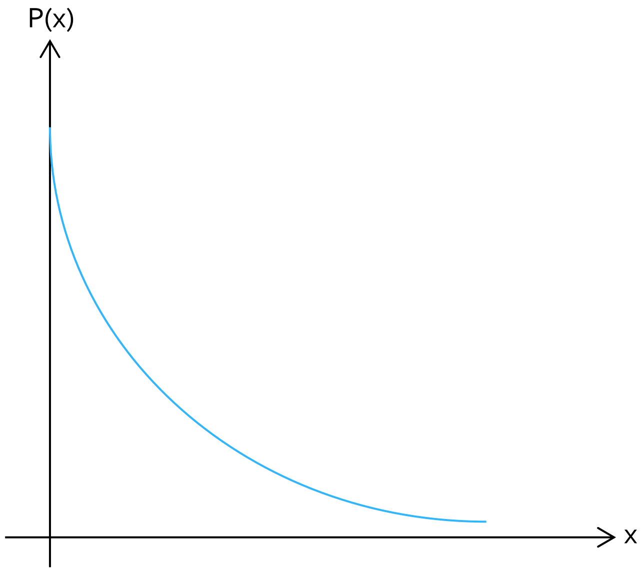

Figure 5: Decaying exponential probability distribution of distance

The equation for this graph is given by:

$$P(x) = e^{-\lambda x}$$

Where:

P(x) = Probability that the distance to the next interface is x units
x = Distance to the next interface
λ = A parameter which controls the rate of decay

Re-arranging this for distance gives:

$$x = \frac{-\ln[P(x)]}{\lambda}$$

By randomizing the variable P(x) between 0 and 1 the resulting distance, x, will have a value between 0 and infinity with a decaying exponential distribution. At each interface the distance to the next interface was randomly assigned using this method. The parameter λ, is a material property which will be determined by the size of the media particles and the type of media. This parameter was made modifiable in the programme.

3.5.5 Vector Rotation

During the calculation of the refracted and reflected rays at each iteration in the code, the incident vector used as an input to both functions was a unit vector along the z-axis ([0, 0, 1]). This meant that both the returned reflected and refracted rays had a direction specified within the coordinate frame of the incident ray i.e. in a local coordinate system where the incident ray lies along the z-axis. In order to reconstruct the path of the light ray in global space the positions at which each ray intercepted an interface would need to be expressed in global coordinates.

Two methods for doing this were investigated. The first was to rotate the refracted and reflected rays into the global coordinate system with each iteration before calculating the intercept point on the next interface. In this way the intercept coordinates calculated would already be in global space. In order to rotate the direction of the rays into global space at each interface a rotation matrix was calculated between the incident ray and the global z-axis. The inverse of this matrix was then applied to the refracted and reflected rays which transformed them from the local coordinate frame of the incident ray to the global coordinate frame. This is illustrated in the example diagram below.

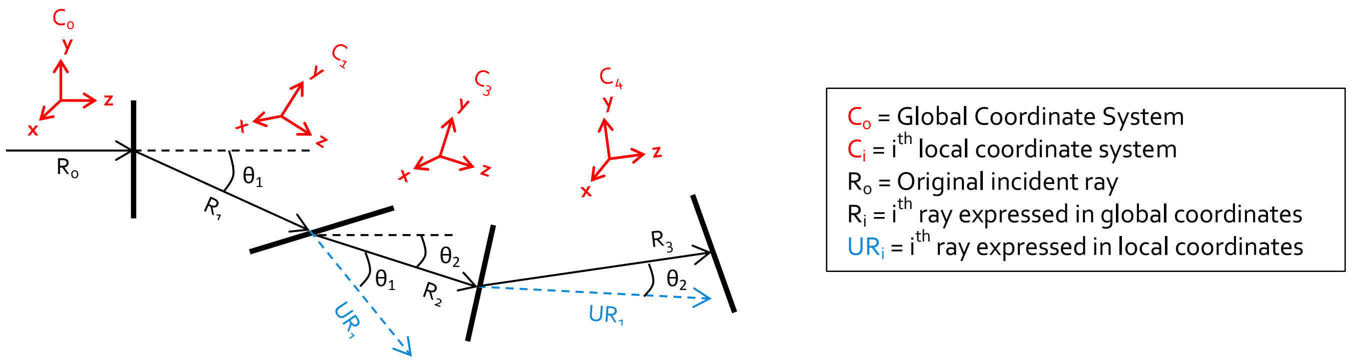

Figure 6: Light ray direction vector rotation example

In the example shown in figure 6 it can be seen that each ray (R_i) is written in terms of the C_{i-1} coordinate system. In order to express each of these rays in terms of the global coordinate system (C_o), a rotation is applied given by (θ_{i-1}) which is calculated between the incident ray (R_{i-1}) and the global z-axis.

In the second method the intercept coordinates at each interface were calculated using the refracted and reflected rays expressed in terms of the local coordinate system of their incident ray. At each step the rotation matrix required to rotate the ray from its current coordinate system into the previous local coordinate system was stored. The stored matrices were then applied in an inverse order to the intercept coordinates to express each point in global space.

It should be noted that calculation of the rotation matrices was done using a function developed by (Möller and Hughes, 1999). The function calculated the 3x3 matrix required to rotate one unit vector into another specified as inputs to the function.

In order to verify that both methods were working correctly a test case in which the ray encountered 22 interfaces was used. The results of the test are shown below.

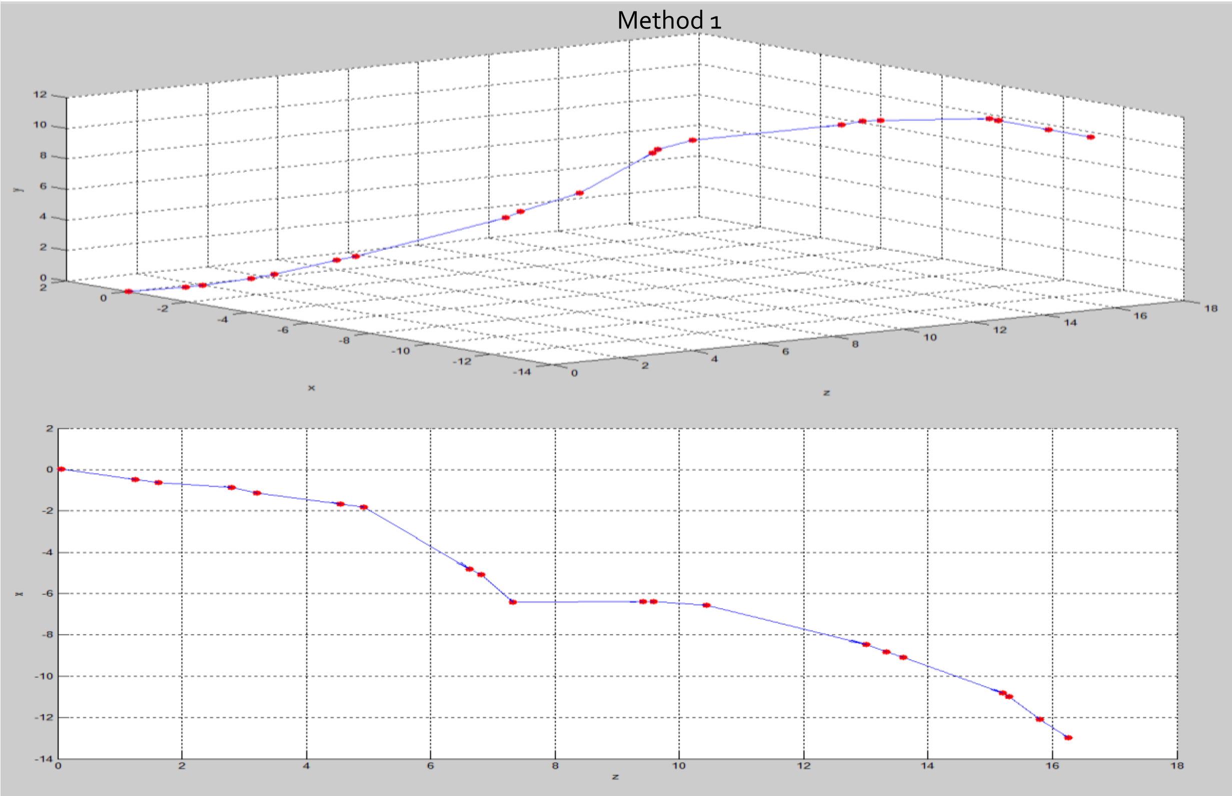

Figure 7: Vector rotation test case method 1. 3-D and x-z plane views shown.

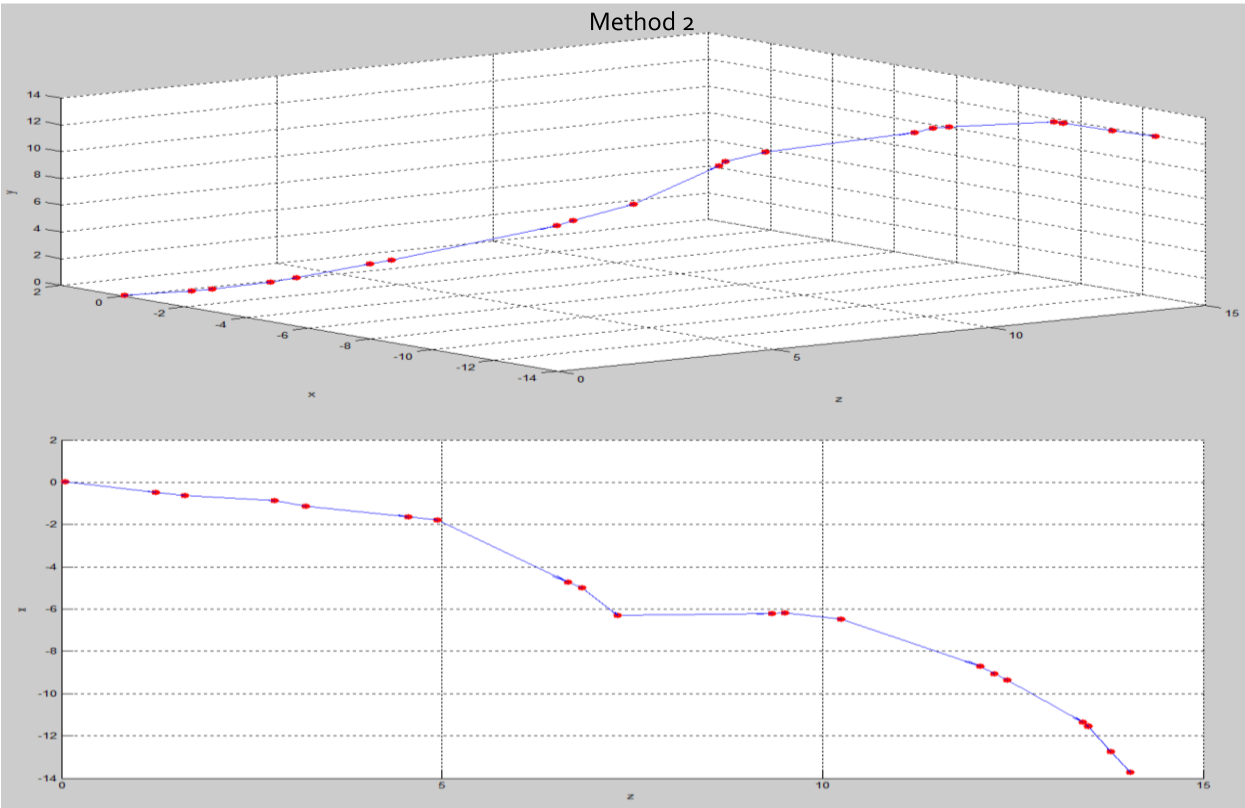

Figure 8: Vector rotation test case method 2. 3-D and x-z plane views shown.

- Direction of light ray

- Interface intercept point

From figures 7 and 8 it can be seen that the overall shape of the light ray path constructed from both methods appear to match. However upon analysing the data points it was noticed that at the later interfaces there was a very small difference in coordinate values between the two methods. Using a few more sets of test data it was discovered that this error was compounded as the number of interfaces increased. Unfortunately due to time constraints I was unable to find the source of this error and so could not fix it.

3.6 Output

In the final version of the programme the rotation code implemented used the method of rotating the refracted or reflected ray into the global coordinate system with each iteration (method 1). The reason for using this system over the second method discussed is due to computational efficiency considerations. In the first method the rotation matrix is overwritten with each step whereas in the second method every rotation matrix has to be stored. By having to store a large number of rotation matrices and then apply them to all the intercepts points in order to reconstruct the ray's path, it was assumed that the second method would be more computationally expensive.

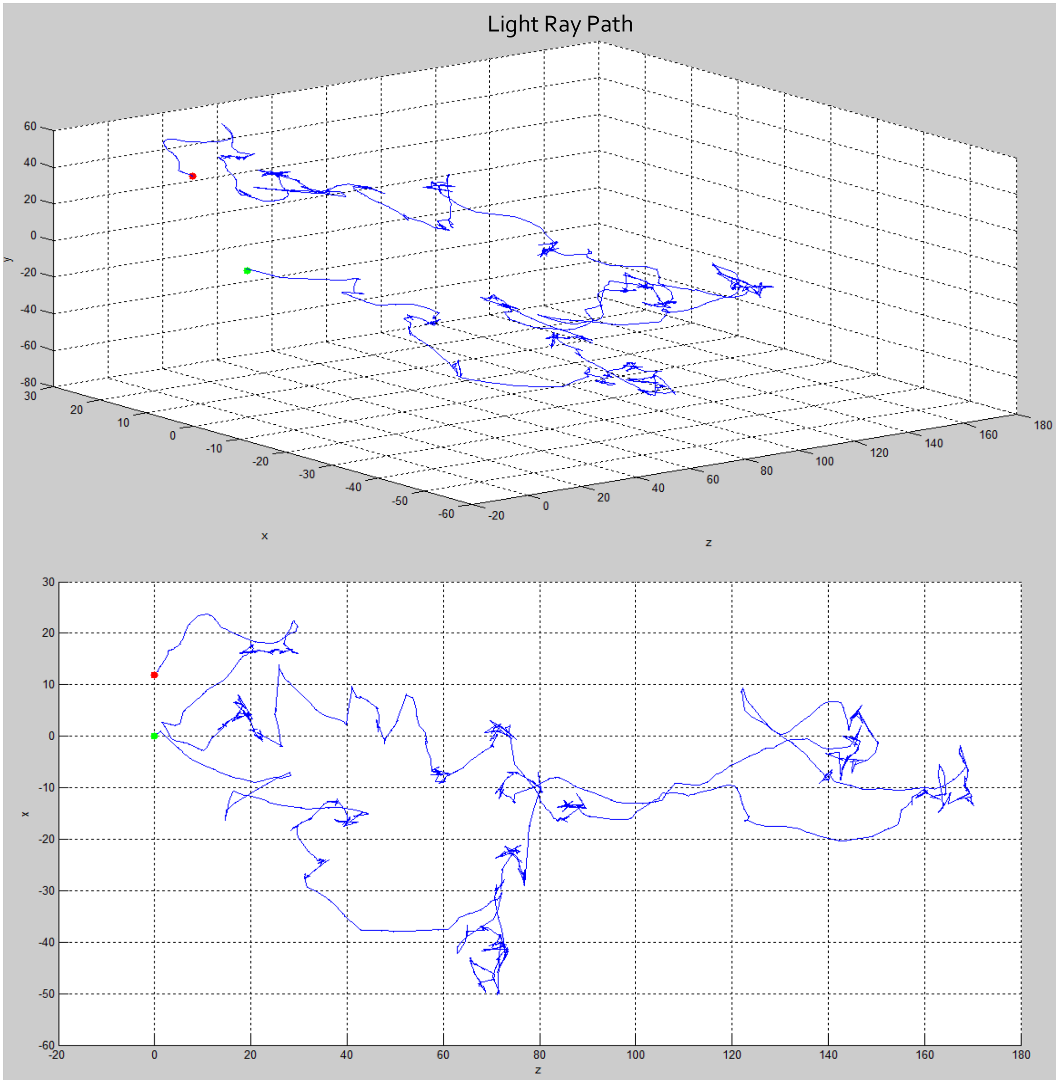

Figure 9: Example Light Ray Path produced by final programme. 3-D and x-z plane views shown.

- The point at which the ray enters the material
- The point at which the ray exits the material

In the sample output shown in figure 9, the ray encountered 2114 interfaces before exiting the material at the point shown. From the limited amount of test runs I performed using the final programme I obtained a range of between about 700 and 12,000 interfaces encountered before the ray exited the material.

The visualisation of the vectors shown in figures 7, 8 and 9 were all done using a function developed by (Xiong, 2005).

3.7 Future Work

One extension to the project which could be carried out is to run the algorithm a large number of times in order to generate statistical data on the average time rays spend inside a given material as well as the average locations where the rays exit the material, both as predicted by the model. The algorithm predictions would then need to be experimentally validated by collecting data on how rays travel inside well characterised materials.

One experimental procedure suggested by Dr Turuwhenua and Dr Vorobyev was to use a defined volume of coloured crushed glass as the material in which lights rays are passed through. In this case the material would be composed of a series of interfaces between glass and air. By collecting light rays which exit the material the experimental results could be used as a comparison to the model predictions.

The model could also be made more realistic by taking into account the amount of energy a ray has and the amount that is lost with each refraction and reflection. As well as this, increasing the computational efficiency of the code is another area for possible improvement.

3.8 Skills Gained

Working at the ABI was an enjoyable experience which allowed me to gain invaluable skills and knowledge. Through my studentship I have learnt a large amount about the field of optics, I have become more proficient at MATLAB and have been able to develop the skill of representing physical phenomena computationally through modelling. As well as this I have learned a great deal about the iterative nature of the research and modelling process.

4.0 Conclusions

The algorithm developed during this project was able to model a light ray interacting with the various boundaries between different media in a material and reconstruct the path of the ray within the material. In order to confirm that the output produced by this programme is realistic experimental data is needed for comparison.

Overall I felt that working at the ABI has allowed me to not only gain new skills and knowledge but also to experience the working environment of an academic institution. The skills which I have developed over the course of the studentship will be of benefit to both my future studies and professional engineering career.

5.0 References

MARSAGLIA, G. 1972. Choosing a Point from the Surface of a Sphere. *The Annals of Mathematical Statistics,* 43, 645-646.

MÖLLER, T. & HUGHES, J. F. 1999. Efficiently Building a Matrix to Rotate One Vector to Another. *J. Graph. Tools,* 4, 1-4.

MULLER, M. E. 1959. A Note on a Method for Generating Points Uniformly on *N*-Dimensional Spheres. *Commun. ACM,* 2, 19-20.

POON, T. C. & KIM, T. 2006. *Engineering optics with Matlab*, World Scientific.

WEISSTEIN, E. W. 2002. Sphere Point Picking. Available: http://mathworld.wolfram.com/SpherePointPicking.html [Accessed 11th December 2012].

XIONG, R. 2005. vectarrow.