DEVELOPMENT OF AN ACCELERATED MONTE CARLO RAY TRACING BASED RADIATION HEAT TRANSFER SOLVER

A Thesis

by

Faizan Pervez Siddiqui

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Electrical and Electronics Engineering

Özyeğin University
July 2017

DEVELOPMENT OF AN ACCELERATED MONTE CARLO RAY TRACING BASED RADIATION HEAT TRANSFER SOLVER

Approved by:

Assist. Prof. Altug Basol, Advisor
Department of Mechanical Engineering
Özyeğin University

Assoc. Prof. Fatih Ugurdag
Department of Electrical and Electronics Engineering
Özyeğin University

Prof. M. Pinar Menguc
Department of Mechanical Engineering
Özyeğin University

Asst. Prof. Melih Kandemir
Department of Computer Engineering
Özyeğin University

Date Approved: July 2017

Assoc. Prof. Hakan Erturk
Department of Mechanical Engineering
Bogazici University

To the two pillars of my life, Allah and my Parents.

ABSTRACT

The Monte Carlo ray tracing (MCRT) is a semi-analytical stochastic method for the solution of radiative transfer equation (RTE). However, its high computation cost prevents its extensive use. In this study, an accelerated MCRT based radiation heat transfer solver has been developed that is capable of handling scenarios with complex geometries in presence of absorbingemitting (but not scattering) gray medium and non-participating media. Additionally, the solver is augmented with the capability of simultaneous 3d visualization of the results. Hardware-wise the use of the graphic processors resulted in about 40x speed-up as compared to the modern high-end CPU's. Moreover, an additional 10x speed-up is also achieved with the use of efficient data structures.

In the developed solver, the RTE is coupled with the energy equation in a way that the solver treats predefined critical regions very precisely while the remaining regions are solved with mediocre accuracy. Using this adaptive technique in the coupling of the equations, the error in the calculations has dropped by 4x compared to the standard MCRT method. The proposed method requires 1.5x more memory compared to the standard MCRT and does not have any penalty in terms of computational time.

ÖZETÇE

.

Monte Carlo n izleme metodu(MCI), radyasyon transfer denkleminin (RTD) zm iin yar analitik bir stokastik yntemdir. Bununla birlikte, metodun yksek hesaplama maliyeti yaygn kullanmn engellemitir. Bu almada, hzlandrlm MC tabanl radyasyon s transferi zcs gelitirilmitir. Yazlm, karmak geometrileri barndran ve radyasyonun iinden getii ortam tarafndan sourulduu ve salnd gri ortamlar iin gelitirilmitir. Ayn zamanda yazlm, hesaplama ile e zamanl olarak 3B grselletirme yeteneine sahiptir. Hesaplamalarda grafik ilemcilerin kullanlmas modern st dzey CPU'lara kyasla 40 kat daha yksek hesaplama hzlar salamtr. Ayrca, kullanlan veri yaplar sayesinde ek olarak bir 10 kat hesaplama hz elde edilmitir.

Gelitirilmi olan yazlm, nceden tanmlanm baz kritik noktalarn daha hassas ve dier tm noktalarn ortalama bir hassasiyet derecesinde zlebilmesini salayacak bir ekilde dizayn edilmitir. Enerji ve RTD denklemlerini birbirine balamada kullanlan adaptif yntem sayesinde hesaplamalardaki hata standart MC yntemine kyasla 4 kat dmtr. nerilen yntem, standart MC'ye kyasla 1.5 kat fazla bellek gerektirir; ama hesaplama sresi asndan herhangi bir arta yol amamaktadr.

ACKNOWLEDGEMENTS

I will take this opportunity to thank my adviser Dr. Altug Basol for accepting me as his Masters student. I found working with him very pleasant because of his kind, considerate and positive personality. He always kept me motivated for the research through his guidance and encouragement during my research. He set realistic goals and deadlines and trusted me, which gave me the confidence to perform efficiently to my best efforts. I have learned a lot from him and I am grateful for his encouragement, support, and patience which enabled me to complete this thesis.

I would also like to thank Professor Pinar Menguc for his support and sharing his knowledge which helped me improve my research. My thanks also go to the other members of thesis defense committee, Dr. Hakan Erturk, Dr. Fatih Ugurdag and Dr. Melih kandemir, for contributing towards finalizing my work with their precious and positive feedback.

Special thanks to my family for their endless love and encouragement. They gave me the opportunities of education from the best institutions and provided unconditional support throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Thermal radiation is a significant mode of heat transfer in many modern engineering problems especially in high temperature applications. Some specific areas include the design and analysis of energy conversion systems such as furnaces, combustors, solar energy conversion devices, and the engines where high temperatures are present to ensure the thermodynamic efficiency of the processes. Thermal radiation can propagate in the absence of any medium and the change in its intensity is governed by the Radiative Transfer Equation (RTE), an integro-differential equation with six independent parameters. A detailed derivation of the radiative transfer equation has been discussed in a book by Howell et al. [1]. In differential form the RTE can be expressed as:

$$\frac{\partial I_\lambda(S,\Omega)}{\partial S} = \kappa_\lambda I_{\lambda b}(S,\Omega) - \kappa_\lambda I_\lambda(S,\Omega) - \sigma_{s,\lambda} I_\lambda(S,\Omega) + \frac{1}{4\pi}\int_{\Omega_i=4\pi} \sigma_{s,\lambda} I_\lambda(S,\Omega)\ \Phi_\lambda(\Omega_i,\Omega) d\Omega_i \tag{1}$$

where the term on the left hand side of the equation expresses the time rate of change in the spectral intensity per unit projected area and the terms on the right hand side dictate the effects of emission, absorption and scattering onto the intensity of the light.

1.1 Thermal Radiation and Its Numerical Modeling

Except some simplified scenarios, RTE can only be solved numerically. Over the years, different type of numerical methods have been developed and used depending on the type of the problem. Radiation heat transfer problems can be divided into

two main categories depending on the way the medium that the thermal radiation is passing through is interacting with it. In conditions where the medium is participating with the thermal radiation transfer, the most frequently used numerical methods are discrete ordinates [2], the spherical harmonics (P-N) [3], the zonal method [4], the finite volume method [5] and Monte Carlo method [6] [7] [8]. On the other hand, for the problems where the medium like air is not participating in the transfer of thermal radiation, it can be modeled as the interactions between the surfaces only. Under these conditions, the calculation is restricted on finding the so called view factors between the surfaces. Calculating view factors can be conducted analytically but only for problems involving regular geometries. Detailed solutions for different view factor scenarios have been provided by Howell and Menguc [9]. For most of the practical scenarios involving complex geometries however, the view factors can only be calculated numerically. Optical projection method by Farrell [10], Cross-string method by Hottel [11], unit sphere method by Nusselt [12], Hemi-cube method by Cohen [13] and Monte Carlo ray tracing method [14] are the most popular ones.

In general, one expects a numerical method to be both accurate and computationally fast. Even though these two requirements seem to be contradicting, depending on the type of the problem even a computationally much faster method can give a more accurate solution compared to the computationally very expensive methods. For example, in problems involving absorbing and highly scattering media even the most simplistic version of spherical harmonics method P_1 can give a very accurate result at a fraction of the computation cost of the Monte Carlo method. Similarly, using the discrete ordinate method for a scenario involving non-participating medium will be both waste of computational resources and as well as accuracy is also deteriorated due to the systematic errors present in the method as discussed by Chai, Lee, Patankar [15].

Among all the methods mentioned, Monte Carlo method is known to be the most

accurate one in the complicated scenarios as discussed in a study by Howell [14]. Its solution is considered to be semi-analytical for the thermal radiation problems. Regarding the view factor studies, Emery et al. [16] compared Nusselts unit sphere method, Farrells optical projection method, Hottels Cross-string method and Monte-Carlo ray tracing method in diffuse view factors calculations. The result showed the superiority of the Monte Carlo method in terms of accuracyof the calculations. The study by Mirhosseini and Saboonchi [17] for the calculation of view factor between strip elements and the cylinder highlighted the accuracy of the MCRT method. In a similar study, Hajji et al. [18] pointed out the accuracy of the MCRT method over the cross string method in a scenario consisting of fins and a semi-cylinder. Problems involving participating medium have also been studied with different numerical techniques. Demirkaya [19] evaluated the Monte Carlo ray tracing method and discrete ordinate method in terms of accuracy in a problem involving three-dimensional, absorbing, emitting and scattering media. The study showed that the Monte Carlo method provides more accurate solution for the given problem. In a similar study conducted by Henson et al. [20], also showed in a similar way the advantage of the Monte Carlo method in accuracy ove the discrete ordinate method.

Due to its stochastic nature, a Monte Carlo method may have random errors resulting uncertainty in the numerical results. This is a major advantage over the conventional mesh based methods such as discrete ordinate or finite volume based numerical techniques. Moreover, in mesh based methods the numerical solution heavily depends on the type and size of the mesh. Even though grid refinement studies tries to reduce the sensitivity of the solution on the mesh, still the solutions include mesh dependent features. This problem becomes even more severe when the geometry becomes more complicated. On the other hand, in Monte Carlo type methods the accuracy depends only to the number of trials. The higher the number of trials, the higher the accuracy gets irrespective of the grid size of the mesh. In this regard, one

can say that the Monte Carlo method is insensitive to the type and size of the mesh.

In spite of the advantages of the Monte Carlo method regarding the accuracy, the prohibitively high computation cost of the Monte Carlo method prevents its widespread use. However, the required computational cost can be substantially decreased with the use of the modern multi-core processors and many-core graphic accelerators. Graphic processor units (GPU's) have mainly been developed to assist the computationally expensive rendering tasks which is explained in a study by loop et al. [21]. In recent years, with the dramatic rise in their computation power and emergence of code libraries GPU's started also to be heavily used as accelerators in scientific computing including radiative heat transfer problems. Halverson [22] presents the solar radiation modeling of an urban area using GPU and Nvidia's Optix engine, He et al. [23] solved radiation in combustion and propulsion applications utilizing graphic processors. Siddiqui et al. [24] used GPU for solving surface to surface radiative heat transfer problems involving complex three-dimensional geometries. Takizawa et al. [25] have implemented the radiosity method on graphic processors. All these studies exploits the computing power of graphics processing units (GPUs) to accelerate the calculation. Moreover, Efremenko et al. [26] compared the performance of multi-core CPUs and GPUs by solving one dimensional RTE using discrete ordinate method. The speed up of 50x has been achieved with the efficient use of graphical processors.

1.1.1 Monte Carlo Ray Tracing (MCRT) Method

The Monte Carlo method is a stochastic method which uses random sampling for the mathematical modeling of a problem. The accuracy of the solution obtained depends on the sample size. Generally speaking, the higher the sample size gets, the more accurate the result becomes. The Monte Carlo technique started to become popular with the emergence of modern computers which provided the necessary computation

power for the method. Today, Monte Carlo method is used in different fields of science and engineering.

In the field of heat transfer, Monte Carlo ray tracing method (MCRT) is used in solving different type of complex problems involving thermal radiation and even conduction. The application of MCRT method for the solution of the RTE was first discussed by John Howell and Perlmutter [27]. They solved the RTE in participating medium between two parallel plates. The validation of the developed method was made by the analytical solution of Usiskin and Sparrow [28]. They concluded that the Monte Carlo method could easily be applied to the solution of complex radiation transfer problems. In a study by Howell [6], it was concluded that the Monte Carlo method has a huge advantage over the other numerical techniques in terms of accuracy as it allows the accurate treatment of inhomogeneous media, spectral properties and complex geometries . The problems could be treated with simplicity and with greater flexibility with the use of the Monte Carlo method. He also highlighted the importance of speed up achieved by implementing Monte Carlo method on massively parallel machines. Several modifications have also been made to the standard Monte Carlo method. In recent years, Haji-Sheikh [29] applied this method to solve for radiation, conduction and convection problems.

Monte Carlo ray tracing technique is particularly suited for the solution of RTE since the method reflects the physics of the light propagation very accurately. According to the general principle of ray tracing, each ray travels in an independent straight path until it interacts with other objects. The method is based on tracing several numbers of rays, which are acting as photon bundles, carrying the finite amount of radiative energy. The path of the bundle is determined by random numbers according to the radiative properties of the medium. Physical events such as absorption, emission, reflection, and scattering can happen in the life of the photon bundle. These events are very well explained by Modest [7].

1.1.2 Ray tracing method

Ray tracing is a technique developed to render images in computer graphics. Today, it is heavily used to generate the so called photo-realistic images from CAD drawings. It is capable of calculating the light effects on the objects very realistically. In this technique, in order to generate the image of a three-dimensional object, a virtual camera is placed at the viewing point. The rays are emitted from the pixels of the view plane and traced throughout the domain until they hit the objects in the scene. This ray-object intersection event determines the color to be represented on the screen. The basic principle of the ray tracing method is shown in Figure 1.

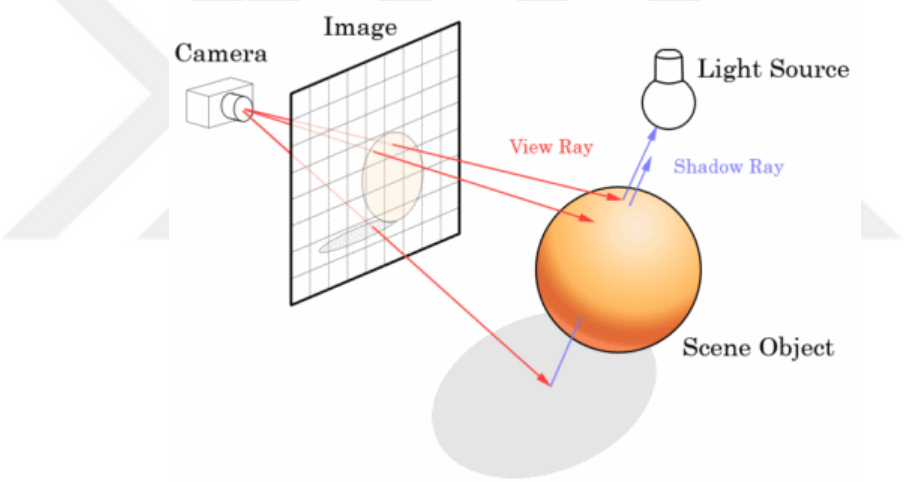

Figure 1: Ray tracing Algorithm

The concept of ray tracing was first introduced to the field of computer graphics by Appell [30] back in 1980 when he rendered a three dimensional image including the effects of reflection and refraction. Kajiya [31] later on combined ray tracing with the Monte Carlo method where he introduced randomness in the path the rays are following to study the effect of the indirect lighting onto the scene.

Even though the ray tracing technique can generate very realistic images, it also is computationally more expensive as compared to other rendering techniques. However, with the dramatic rise in the computation power of the modern graphic processors,

the ray tracing technique started to be used even in real-time applications. And in the near feature, it might be the dominant rendering technique in the computer graphics field.

1.1.3 Data Structure Techniques used for Acceleration

The main part of the ray tracing technique is the calculation of the ray-object intersection points. To calculate the complicated light effects, one has to consider the rays transferred between all the triangular surface elements in the scene. This is a computationally intensive task and the complexity increases even quadratically with the number of triangles present in the domain. To reduce the complexity of the problem, sophisticated data structure techniques have been developed and a dramatic decrease in the computation cost of the problem is achieved especially for scenarios containing large number of surface elements. The two popular data structure techniques, called also acceleration structures, are KD-tree [32] and bounding volume hierarchy [33]. A detailed study has been conducted by Vinkler et al. [34] on the performance comparison of KD-tree and Bounding volume hierarchy on many core architectures (GPU). As discussed, the Bounding Volume Hierarchy showed a superior performance over the KD-tree.

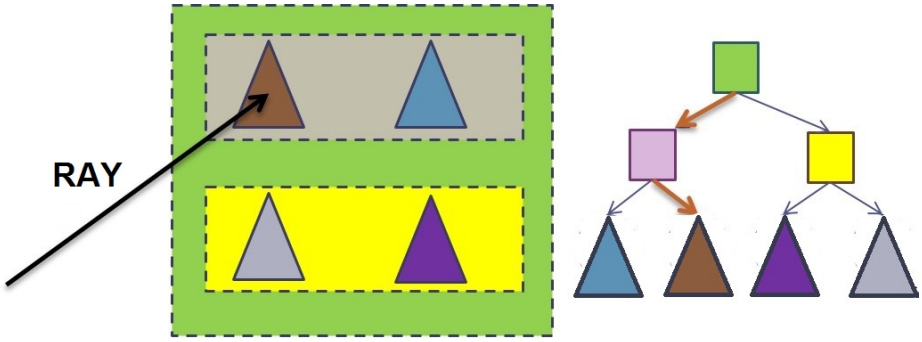

Figure 2: Bounding Volume Hierarchy acceleration technique

Figure 2 explains the basic principle of Bounding Volume Hierarchy (BVH) acceleration technique. In BVH, the basic triangular elements of the geometry are enclosed

inside a virtual bounding volume. The basic elements are shown by the triangular shape while its corresponding bounding volume is represented by a rectangle. These bounding volumes enclose each other to form a tree structure in which each node contains a bounding box of all the objects below it. When the ray hits any of the triangular element, as shown by black arrow, the ray object intersection has to be checked just on its corresponding bounding volumes eliminating the others which results in the massive reduction of computational expense. The speed up achieved by using this acceleration technique is explained in the results section.

1.1.4 Graphical Processing unit (GPU)

Graphical processing unit (GPU) is a specialized processor entirely developed for the rendering applications in the computer. [35]. The term GPU was popularized by Nvidia in 1999, who marketed the GeForce 256 as "the world's first GPU" [36]. Due to the completely data parallel nature of the rendering task, GPU's are designed by hardware to deal with data parallel computations. The main difference between the CPU and GPU processing is that the former is optimized for latency whereas the latter is optimized for throughput. Today, CPU's consist of tens of processing cores whereas GPU's have thousands of them on a single chip. The requirements in the computer graphics field made the GPU develop much differently compared to CPU.

While GPU was actually developed for graphics applications,the parallel processing capability can also be used to speed-up other computationally intensive tasks which are not related with the computer graphics. Its sophisticated parallel architecture and extreme computational efficiency made it applicable in the field of scientific computing, called general purpose GPU-accelerated computing. GPU-accelerated computing is the efficient use of GPUs together with the CPU to accelerate the application. GPU-accelerated computing offloads the computationally expensive portion of the code to the GPU, while the remaining portion of the code still processed on

the CPU. The first scientific problem to be successfully processed faster than CPU was the implementation of LU factorization [37]. Today, GPUs are widely used in different fields of applications ranging from computer vision [38], video processing [39], physical simulations involving fluid dynamics [40] [41] to molecular modeling [42] astrophysics [43] and bioinformatics [44]

1.1.5 Introduction to CUDA and OptiX

Due to the inherent hardware architecture of GPU, their performance heavily depends on the way it is programmed. Nowadays, these GPU's can be found in every personal computer and mobile phones and the efficiency and computation power of GPUs is increasing day by day. Many applications, software, and libraries, like CUDA, OpenCL, Optix, are developed to allow users to use a graphic processor for general purpose computing. The application of these developed frameworks helps to optimize and utilize the computation power of GPU for other massively parallel applications rather than rendering.

CUDA is a framework, developed by NVIDIA, for the programming of the graphics processing unit for general purpose computing. CUDA framework allows the user to develop any general purpose applications which can be processed on GPU [45]. This breakthrough development helps the researchers in accelerating their applications using the parallel processing capabilities of GPU.

OptiX is a ray tracing engine developed over CUDA to provide additional optimization technique specifically for ray tracing problems [46]. It provides several functions to the user to populate the scene with complex objects and different lighting methods. It also provides different acceleration method for the ray-object intersection to further improve the performance. The main purpose of this framework is the optimized utilization of graphical processing unit. Although it was developed to accelerate the graphical operations in gaming industry, it can be used to develop any

general purpose ray tracing application.

1.1.6 Visualization of the results

Solving and post-processing are generally considered to be isolated stages of a numerical study. Integration of post-processing and visualization stages into the solver is a recent concept and offers some certain advantages. Visualization of the results alongside the computation gives the user the ability to control and check the results while the simulations are running. Following this recent approach, the present software is also designed to conduct both the calculations and the visualization of the results simultaneously. This way the user can monitor the status of the simulation in a rich 3D environment.

In this regards, the same ray tracing library used in the solution of the RTE is also used for the visualization of the simulation data. Figure 3 and 4 shows the color coded and visualized view factor values on the surfaces.

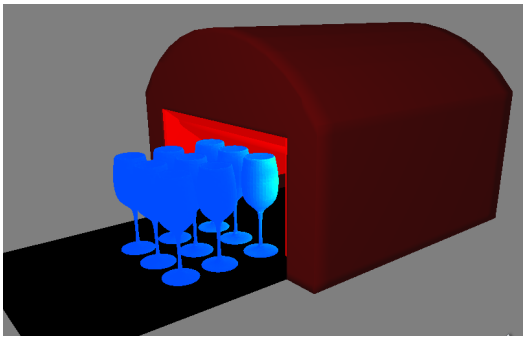

Figure 3: Heat transfer in annealing furnace

Figure 4: heat transfer involving complex geometries

1.2 Thesis Contribution

The Monte Carlo ray tracing method is known for its superior accuracy and suitability for complex geometries but its extreme computational cost prevents its use in most realistic scenarios. The present work focuses on the techniques to accelerate the Monte Carlo ray tracing method and thereby increase its applicability in the next

generation radiation heat transfer solvers. The acceleration techniques cover the use of graphic processors on the hardware level and data structures on the programming level. Additionally, MCRT method has been modified algorithm-wise in the way the RTE is coupled with the energy equation. The new method enables to solve some user-defined critical regions very accurately and everywhere else at some mediocre accuracy and thereby optimizes the memory usage of the hardware, computational speed of the calculation and the accuracy of the solution.

1.3 Thesis Outline

This thesis is organized as follows. Chapter I is the introduction of the study with the relevant literature review. Chapter II talks about the numerical methods and their implementation in the solver with validation cases. Chapter III is about the results and test cases with proposed numerical methods. Finally, Chapter IV discusses the conclusion.

CHAPTER II

NUMERICAL METHODS

2.1 Surface to surface radiative heat transfer

Thermal radiation can not penetrate into opaque surfaces. It is absorbed and emitted entirely on the surface. Under these conditions the transfer of heat by thermal radiation heavily depends on how the objects are positioned with respect to each other. The so-called view factor defines the fraction of radiant energy leaving any given surface that is incident upon another surface. It entirely depends on the geometry of the objects and their relative positions.

The view factor between two differential areas, as shown in Fig. 5,was studied by Sparrow [47]. This view factor can be calculated by using the equation 2

$$dF_{dA1-dA2} = \frac{cos(\theta_1)cos(\theta_2)}{\pi S^2} dA_2 \tag{2}$$

And specifically, the view factor F between a differential area dA_1 and the surface A_2 is expressed as:

$$F_{dA1-A2} = \int_{A2} \frac{cos(\theta_1)cos(\theta_2)}{\pi S^2} dA_2 \tag{3}$$

This equation illustrates how the view factors depend on the shape and orientation of the surfaces as well as the distance between them. If area A_1 is not differential then another integral over that area has to be considered, then equation 3 becomes:

$$F_{A1-A2} = \int_{A1} \int_{A2} \frac{cos(\theta_1)cos(\theta_2)}{\pi S^2} dA_2 dA_1 \tag{4}$$

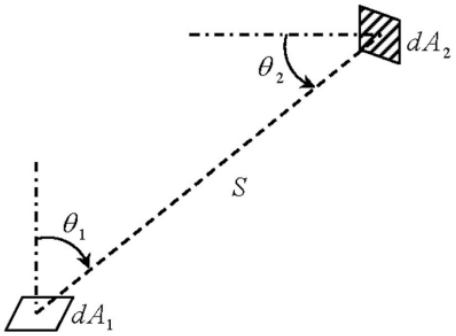

Figure 5: Geometric configuration for radiative heat exchange between two finite surfaces

2.1.1 Calculation of Radiative Heat Flux to the Surface

The net radiative energy exchange between the differential surfaces d_{A1} and d_{A2} is given by the Equation 5

$$\dot{Q}_{A1-A2} = A_1 F_{A1-A2} \epsilon \sigma (T_1^4 - T_2^4) \tag{5}$$

Where $\dot{Q}$ is the net heat flux, A_1 is the area of the first surface, F_{A1-A2} is the view factor from 1 to 2 , σ is the Stefan-Boltzmann constant , ϵ is the emissivity of the surface and T_1 and T_2 are the temperatures of the surfaces 1 and 2.

2.1.2 Numerical calculation of view factor

The developed tool is able to solve surface to surface radiative heat exchange between the surfaces for complex three-dimensional geometries using Monte Carlo ray tracing (MCRT) method. The flow chart for the calculation of view factor is given in Figure 6.

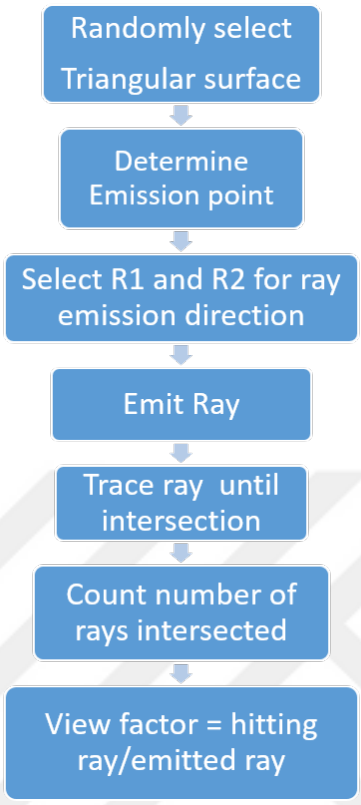

Figure 6: Flow chart for the calculation of view factor

As a first step, the solver needs a CAD geometry as an input and after the calculations it provides the user with the visual map of view factor. The step by step procedure for the calculation of the view factor between the surfaces is explained in the Figure 6

2.1.2.1 Random Number Generation

The most critical issue in the implementation of MCRT method is the generation of random numbers. The accuracy of the methods heavily depends on them. In the present work, uniform random numbers are generated. In uniform distributions all the numbers are equally probable to be selected. The probability density function (U) of the continuous uniform distribution between interval a and b is:

$$U(a,b) = \frac{1}{b-a} \qquad for\ a \leqslant x \leqslant b \tag{6}$$

$$U(a,b) = 0 \qquad for\ x < a$$

$$U(a,b) = 0 \qquad for\ x > b$$

Different algorithms are present in the literature for the generation of uniformly distributed random numbers. In this study, Tiny encryption algorithm (TEA) is used for that purpose. TEA is actually a deterministic algorithm that generates pseudo-random numbers by introducing a seed through the cryptographic function. Zafar [48] has used the Tiny Encryption Algorithm on graphic processors successfully.

2.1.2.2 Determination of the point of emission:

The determination of the positions of the emission points plays an important role in the accuracy of the Monte Carlo method. In this regard, the geometry is divided into several triangular surface elements and several methods are tested to pick the emission points from every region of the surface at equal probability. In this work, the position of the emission point is determined with the use of two different random numbers. First, any point on the edge AB of the triangle shown in Figure 7 is selected randomly. Then using a second random number any position on the line $CR1$ is chosen. In this way, the coordinates of the emission points on the triangular surface elements are determined.

2.1.2.3 Initial Launch Angles:

After calculating the emission point, the direction of the ray is determined by defining the azimuthal ϕ and zenith angle θ.[1] The zenith angle θ can be calculated using Equation 7, where R_1 is selected randomly between 0 and 1:

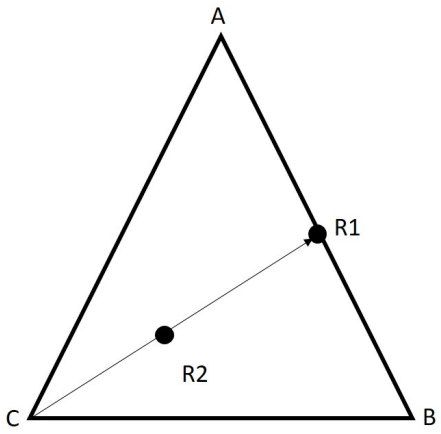

Figure 7: Selecting Emission point from the triangular primitive

$$\theta = sin^{-1}\sqrt{R_1} \tag{7}$$

The azimuthal angle ϕ can be calculated using Equation 8, where R_2 is selected randomly between 0 and 1

$$\phi = 2\pi R_2 \tag{8}$$

Finally, the direction vector p can be determined using zenith and azimuth angles as given in the Equation 9.

$$p = \left\{ \begin{array}{c} \sin\theta cos\phi \\ \sin\theta sin\phi \\ \cos\theta \end{array} \right\} \tag{9}$$

2.1.2.4 Determination of the ray-object Intersection:

After the ray is launched from its emission point in the determined direction, it travels throughout the domain until it hits another surface whose coordinates has to be calculated. Calculation of ray-object intersections is computationally a very expensive task. This requires a lot of vector computations between the ray and the objects as explained by Glassner[49]. The Optix library [46] library is specifically

optimized to deal with this computationally expensive task. Several acceleration techniques such as KD-tree, Bounding volume hierarchy etc are used to accelerate this expensive task.

2.1.2.5 Computing View Factors:

In order to calculate the view factor between the surfaces, a counter program is implemented to count the number of rays emitted from the source which are absorbed by the target body. Finally the view factor can be determined by using equation 10.

$$F_{1-2} = I/J \tag{10}$$

Where I is the number of rays intercepted by the object and J is the total number of rays emitted from the source.

2.1.2.6 Accuracy analysis

As mentioned before, the accuracy of the Monte Carlo method heavily depends upon the distribution of random numbers. The accuracy of this method can be calculated statistically. Let the result obtained from the Monte Carlo method, after tracing N energy bundles, be X. Several independent trials is conducted, each with the unique set of random numbers. The calculated value $\bar{x}$ is the arithmetic mean of n number of trials . The standard error of the calculation, can be calculated by using the Equation 11

$$S_n(\bar{x}) = \frac{s_n(x)}{\sqrt{n}} = \frac{1}{\sqrt{n(n-1)}}\sqrt{\sum_i (x_i - \bar{x}_i)} \tag{11}$$

where, $s_n(x)$ is the standard deviation. According to the central limit theorem, the standard deviation of the solution decreases by the factor of $1/\sqrt{N}$. The Central limit theorem states that the error of each independent simulation follows a Gaussian distribution. This implies that 68.3% confidence can be claimed that actual answer

lies within the limits of $\bar{x} \pm s(n)$. 95.5% confidence within$\bar{x} \pm 2s(n)$, or with 99% confidence within $\bar{x} \pm 2.58s(n)$.

2.1.3 Validation case

To validate the developed method the numerically calculated view factor results are compared with the analytical solutions. In this regard, the view factor calculation scenario between a sphere and a plate is selected from the catalog by Howell [50]. The analytical solution for the problem was derived by Feingold and Gupta [51] and expressed by the equation 12

$$F_{1\to2} = \frac{1}{2\pi} sin^{-1}[\frac{2B_1^2 - (1-B_1^2)(B_1^2+B_2^2)}{(1+B_1^2)(B_1^2+B_2^2)}] + sin^{-1}[\frac{2B_2^2 - (1-B_2^2)(B_1^2+B_2^2)}{(1+B_1^2)(B_1^2+B_2^2)}] \tag{12}$$

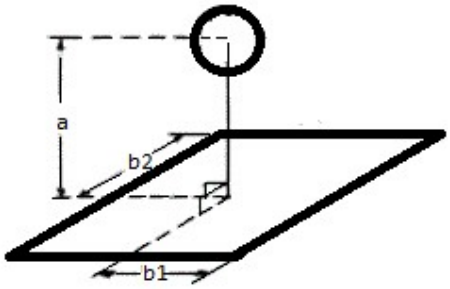

Figure 8: Validation case for calculating view factor

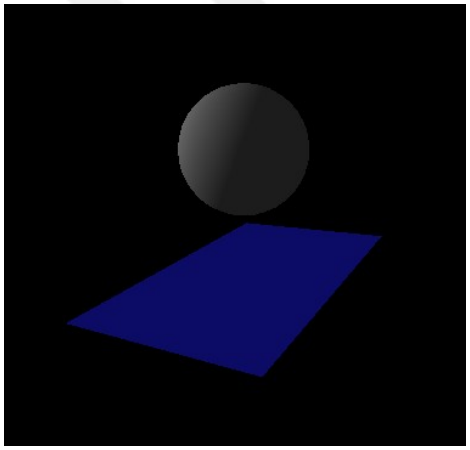

Figure 9: Validation case for calculating view factor as seen in the solver

The numerical calculation of the view factor has been conducted with varying distances between the sphere and the plate.The test case has been shown in Figure 8 and 9. For each case, 0.275 million rays were used for the calculation. Figure 10 shows the calculated view factors with respect to the spacing between the objects. As shown, the numerical results agree well with the analytical solution.

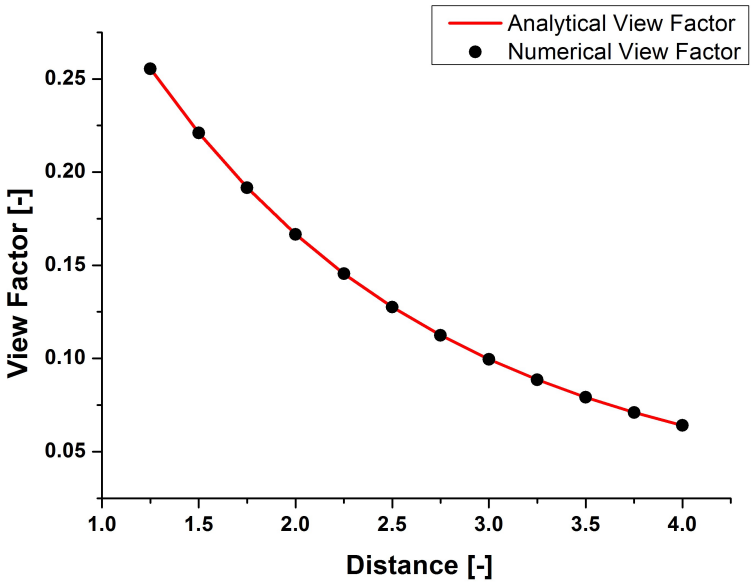

Figure 10: View Factor vs Distance between plate and sphere

Since Monte Carlo method is a stochastic technique, where the outcome of every view factor calculation even with the same number of rays would be different. Repeating the view factor calculation thousand time for the scenario, shown in Figure 8, using exactly 0.275 million rays, one can obtain the histogram of the relative errors in the calculated view factor values with respect to the analytical one. As shown in the Figure 11, the distribution of the relative errors agree with the Gaussian distribution around its mean. This observation also validates the stochastic nature of the implemented method.

Figure 12 shows the calculated view factors for a fixed distance between the sphere and the plate vs. the number of rays used in the calculation. The calculated view factors on the plot are the mean of hundred different calculations carried outwith the same number of rays. The analytical solution is represented by a continuous line on the plot. As shown, the higher the number of rays traced ,the closer is the calculated result gets to the analytical solution. The vertical bars on the plot represent the standard deviations in the view factor values. The height of the bar is equal to the

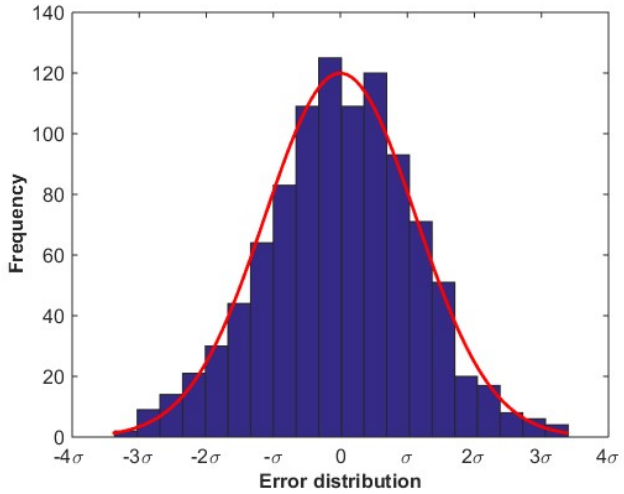

Figure 11: Distribution of Error

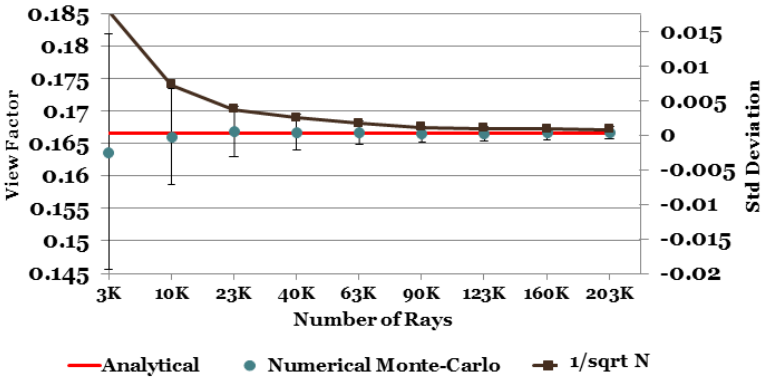

Figure 12: View factor with number of ray and standard deviation

twice of the standard deviation, which means 95% of the result lies within that range. As shown, the height of the vertical bars also reduce with the increase in the number of rays. More precisely, the relation is matching with the inverse square root function added on top of the plot. Central limit theorem dictates the relationship between the standard deviation and number of trials or samples in Monte Carlo methods. As shown in this validation case, the drop in the standard deviation is completely in line with the Central limit theorem. This also validates the implementation of the developed tool.

2.2 Radiation heat transfer in absorbing medium

Thermal radiation can penetrate into the medium if the surfaces are not opaque. Similarly, thermal radiation emitted from the interior of the objects can also leak out from the object. The medium that the thermal radiation is interacting with can absorb the incident radiation or emit as well. In presence of non-scattering medium the RTE is simplified into the form given in Equation 13:

$$\frac{\partial I_\lambda(S,\Omega)}{\partial S} = \kappa_\lambda I_{\lambda b}(S,\Omega)dS - \kappa_\lambda I_\lambda(S,\Omega)dS \tag{13}$$

This section discusses the implementation of the MCRT method for the solution of RTE in presence of absorbing and emitting medium.

2.2.1 Mathematical relation for the divergence of heat flux

In problems involving absorbing and emitting medium one is interested in the net amount thermal radiation emitted per unit time from any particular region expressed by the term $\boldsymbol{\nabla} \cdot q_R$ and calculated using Equation 14.

$$\int_{V_l} \boldsymbol{\nabla} \cdot q_R \; dV = 4\kappa_{Pl}\sigma T_l^4 V_l - \sum_{j=1}^{J} \varepsilon_j \sigma T_j^4 F_{j\to l} - \sum_{k=1}^{K} 4\kappa_{Pk}\sigma T_k^4 V_k F_{k\to l} \qquad l = 1,2,3...K \tag{14}$$

Where

$\boldsymbol{\nabla\cdot}\ \ q_R = Divergence\ of\ the\ radiative\ heat\ flux$

$\kappa = Plank's\ mean\ absorption\ coefficient$

$\sigma = Stefan\ -\ Boltzmann\ constant$

$F_{j\to l} = Exchange\ factor\ matrix\ from\ surface\ j\ to\ volume\ l$

$F_{k\to l} = Exchange\ factor\ matrix\ from\ volume\ k\ to\ volume\ l$

The first term on the right hand side of the Equation 14 expresses the rate of outgoing heat transfer by emission and the other terms stand for the rate of incoming heat transfer by absorption of the radiation emitted from surface and volume elements inside the domain.

2.2.2 Numerical solution for emitting and absorbing medium

MCRT method is adapted for the numerical solution of the particular problem. Steps of the algorithm are explained in the flowchart given in Figure 13.

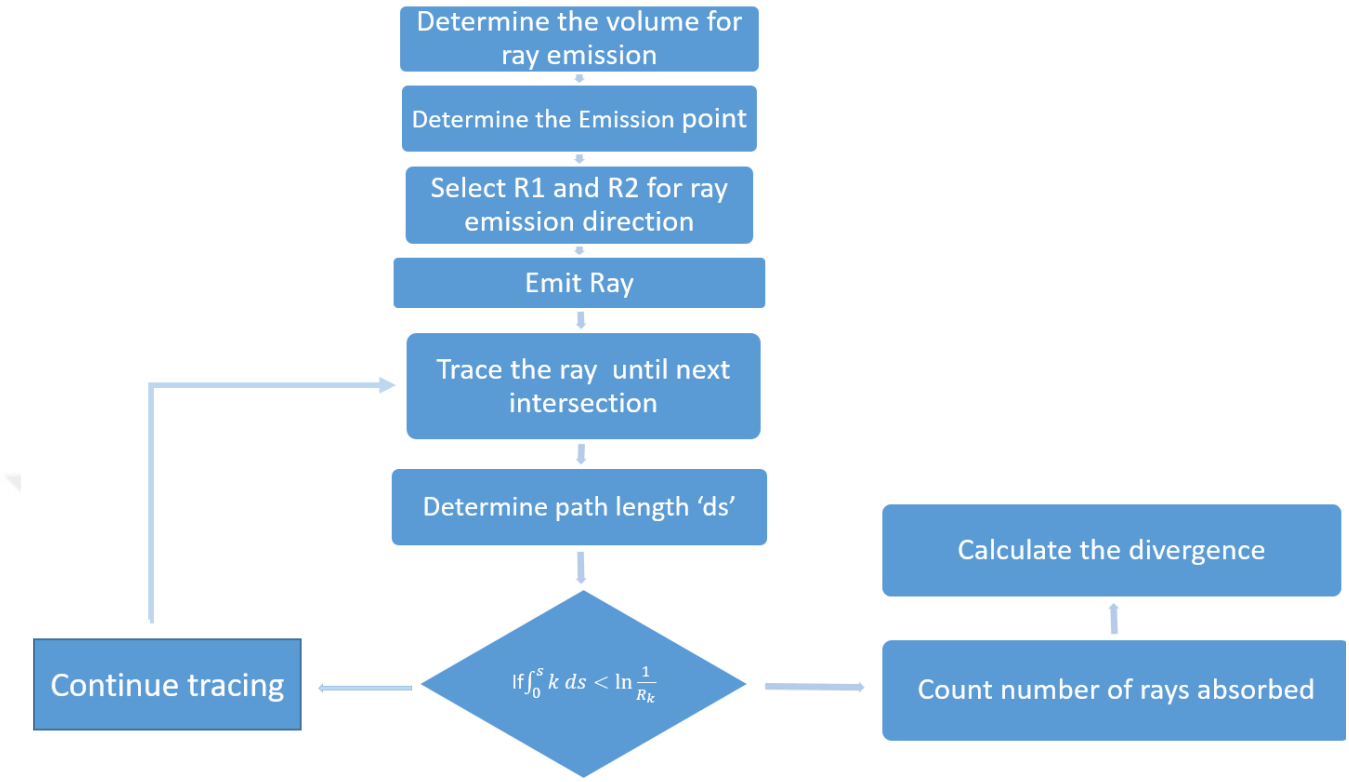

Figure 13: Flow chart for the calculation divergence of radiative heat flux

2.2.2.1 Calculation of Rate of Heat Transfer by Emission

Rate of heat transferred by the volume element K via emission is calculated with

$$E_k = \int_{V_k} 4\kappa_P \sigma T^4 dV \tag{15}$$

Since $dV = dxdydz$, Equation 15 becomes:

$$E_k = \int_0^X (\int_0^y \int_0^z 4\kappa_P \sigma T^4 dzdy)dx = \int_0^X E_k^{'}(x)dx \tag{16}$$

This emitted thermal radiation is transferred to the surrounding by the rays released from the same volume element. The coordinates of the starting positions of the rays are calculated using three random numbers $R_x R_y and R_x$:

$$R_x = \frac{1}{E_k}\int_0^x E_k^{'}\,dx = \int_0^x \int_0^y \int_0^z \kappa_P \sigma T^4 dz\; dydx / \int_0^X \int_0^y \int_0^z \kappa_P \sigma T^4 dz\; dy\; dx \tag{17}$$

$$R_y = \int_0^y \int_0^z \kappa_P \sigma T^4 dz dy / \int_0^Y \int_0^Z \kappa_P \sigma T^4 dz dy \tag{18}$$

$$R_z = \int_0^z \kappa_P \sigma T^4 dz / \int_0^Z \kappa_P \sigma T^4 dz \tag{19}$$

or

$$x = x(R_x), \qquad y = x(R_y, x), \qquad z = z(R_z, x, y) \tag{20}$$

2.2.2.2 Directions for the emission within the medium

After calculating the coordinates of the points the rays are released, the direction of the rays are next determined. The solid angle of $4\pi = \int_0^{2\pi} \int_0^{\pi} sin\theta d\theta \psi sin\theta d\theta d\psi$, contains all the possible direction of the emission within the medium. Since integrand is separable,

$$R_\psi = \frac{\psi}{2\pi}, \qquad or \qquad \psi = 2\pi R_\psi \tag{21}$$

$$R_\theta = \frac{1}{2} \int_0^\theta sin\theta d\theta = \frac{1}{2}(1 - cos\theta), \tag{22}$$

$$\theta = cos^{-1}(1 - 2R_\theta) \tag{23}$$

Finally, the direction vector p can be determined using zenith and azimuth angles as given below.

$$p = \left\{ \begin{array}{c} \sin\theta cos\phi \\ \sin\theta sin\phi \\ \cos\theta \end{array} \right\} \tag{24}$$

2.2.2.3 Absorption within medium

When the ray travels through the absorbing medium, the energy carried with each ray is reduced. This reduction of energy carried by a ray, traveling through a thickness l_k, is given by the Equation 25:

$$R_\kappa = exp(-\int_0^{l_k} \kappa ds) \tag{25}$$

From Equation 25 , it can be implied that the fraction of energy R_k will travel over a distance l_κ. Thus, the distance any one bundle can travel before the absorption can be determined by inverting the Equation 25

If the absorption coefficient κ is constant throughout the medium, the bundle travels a distance of l_k , given in Equation 26, before it gets absorbed.

$$l_\kappa = \frac{1}{\kappa} ln \frac{1}{R_\kappa} \tag{26}$$

If κ is not constant , the allowed distance to be travelled can be determined by breaking the volume in K sub volumes, such that κ is constant within the cell. The integral in Eq 25 can be converted into summation.

$$\int_0^s \kappa ds \simeq \sum_k \kappa s_\kappa, \tag{27}$$

The summation in Equation 27 is over all the cells through which the bundle has passed and s_k is the total distance the bundle has travelled. The bundle is not absorbed as long as the Equation 28 satisfies:

$$\int_0^s \kappa ds < \int_0^{l_k} \kappa ds = ln \frac{1}{R_k} \tag{28}$$

Equation 28 is used to determine the distance that the ray has to travel until it gets absorbed.A counter is used to count the number of rays absorbed in a volume,

which later on used for the evaluation of the exchange factor. Finally, the divergence of radiative heat flux is calculated by using Equation 14.

2.2.3 Validation case

The test case developed by Hsu and Farmer [52] is used to validate the developed tool. It has previously been studied in the literature. Henson [20] has used the same test case to compare the discrete transfer and Monte Carlo methods. Also, Guo and Maruyama [53] discussed a new numerical method for the solution of RTE in participating medium using the same test scenario.

The test problem consists of a unit cube enclosing an isothermal, absorbing, emitting medium. All the surfaces of the enclosure are cold and black such that there is no emission or reflection from the boundaries. The extinction coefficient inside the cube is defined by the relation :

$$\beta(x, y, z) = 0.9(1 - 2|x|)(1 - 2|y|)(1 - 2|z|) + 0.1 \tag{29}$$

The coordinate origin lies at the center of the cube extending by 0.5 in length in every directions. The participating medium has unity black body emissive power. First, the domain is sub-divided into 9 uniform and cartesian cells in every direction. The centers of the cells are located at (x, y, z) = (0,1/9, 2/9, 3/9, 4/9). For every cell the extinction coefficient is evaluated from equation 29 and assumed to be constant within the cell. Following the explained collision based Monte Carlo method, the divergence of radiative fluxes for every cell in the cube is calculated.

Table 1 shows the divergence of the radiative flux of the cells at the centerline of the cube calculated with the present method along with the other studies from the literature.

As shown in Table 1 the results of the present tool is in agreement with the results from the literature. In total 5 million rays are used in the calculation.

Table 1: Divergence of the radiative heat flux at (x,0,0)

x	MCRT [present]	Uncertainty	MCRT[52]	MCRT [20]	DT	YIX
± 4/9	0.723284	0.000257	0.72910	0.72336	0.72860	0.72219
± 3/9	1.376448	0.000515	1.38739	1.37701	1.38099	1.37208
± 2/9	1.969357	0.000851	1.98360	1.96893	1.96458	1.95658
± 1/9	2.519262	0.001158	2.53635	2.51700	2.52182	2.49628
0	3.07424	0.001449	3.09813	3.07462	3.08144	3.03664

The uncertainty in the results are also listed in Tables 1. They are obtained by hundred independent Monte Carlo simulations performed. The variation in the standard deviation with respect to the number of rays is shown in Figure 14.

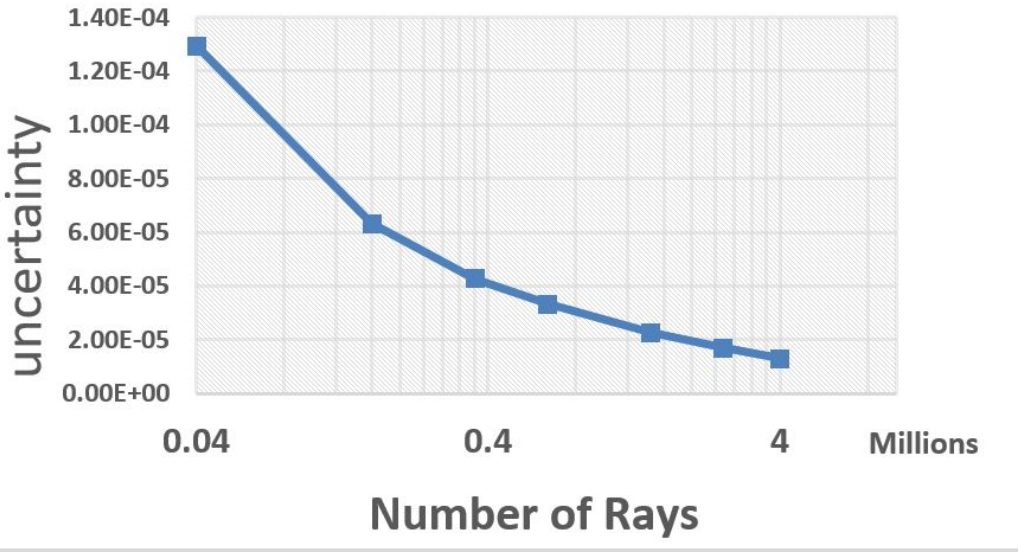

Figure 14: Uncertainty with the number of rays

2.3 Coupling radiative transport and heat transfer equation

Solution of RTE only is rarely of any practical use if it is not coupled with the energy equation to solve for the change in the temperature. Accordingly, the divergence of the radiative flux obtained from the RTE is fed into the energy equation as a source term. Solving for the energy equation, the change in the temperature due to the thermal radiation can be calculated. However, this requires the divergence of the radiative flux term to be re-calculated with the updated temperature level in the

cells. This iterative process continues until the required time level is reached.

The ordinary differential equation for the solution of the change in temperature only due to radiation, neglecting the effect of conduction and convection, is given in the Equation 30.

$$\rho c \frac{dT}{dt} = -\boldsymbol{\nabla} \cdot q_R \tag{30}$$

For the numerical time integration of the problem the 1st order Forward Euler method is used given in Equation 31

$$T_{n+1} = T_n - (\frac{\Delta t}{\rho c})\boldsymbol{\nabla} \cdot q_R \tag{31}$$

Equation 31 is used for the calculation of the temperature at each time step. This updated temperature is then used for the re-calculation of the divergence. The whole iterative process for the solution of the temperature is shown in Figure 15.

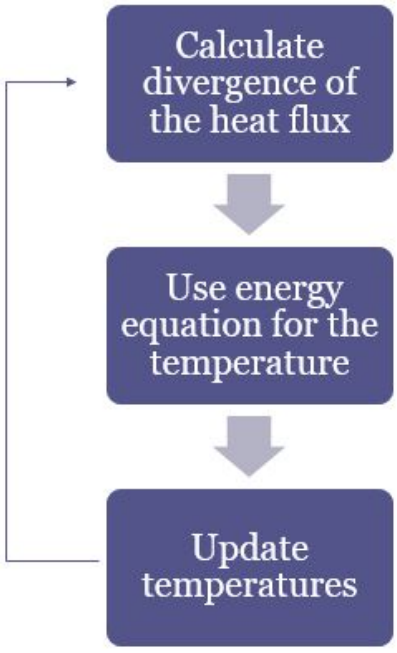

Figure 15: Process flow for the calculation of the temperature

CHAPTER III

RESULTS AND DISCUSSION

This section first discusses the effectiveness of acceleration techniques both hardware- and programming-wise on reducing the computational cost of the MCRT method. The speed-up tests are conducted on a standard view factor calculation scenario. The selected test case has already been studied in the literature, so the computation time for the number of rays traced can be compared.

Another purpose of this study is the coupling of RTE with the energy equation for solving the heat transferred by radiation. In this regard, a new method is proposed that balances the memory usage of the GPU, computational time of the problem and as well as the accuracy of the solution.

3.1 Acceleration of Ray Tracing

As discussed in the numerical method section, Monte Carlo ray tracing method requires tracing of millions of rays to get a solution with acceptable accuracy. The computation cost associated with that can be dealt with the computational power of modern graphic processors and as well as with the use of efficient data structures. In this study the effectiveness of each of these acceleration techniques will be separately discussed. The speed-up due to the running the ray tracing calculation on GPU is evaluated by running the scenario studied by Mirhosseini and Saboonchi [17] and Cosson et al. [54] with the present tool and comparing their run times with the present one. The test case consist of a cylinder and rectangular plate as shown in the Figure 16.

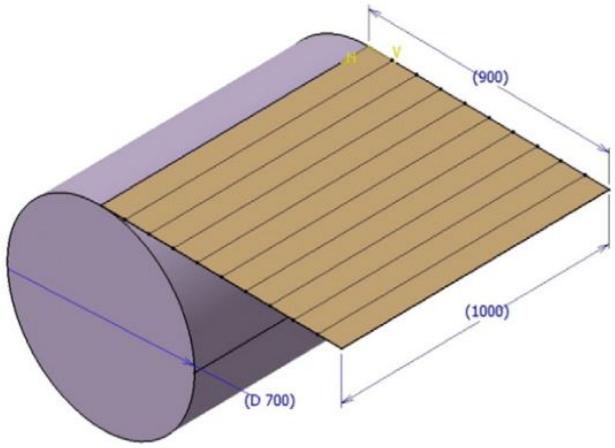

Figure 16: Test case for the comparison of computation time

3.1.1 Acceleration due to the GPU Hardware only

The fact that the ray tracing calculations are completely independent from each other makes the MCRT method ideal for the GPU hardware. Table 2 shows the number of rays traced per second on the indicated hardware and as well as the year of publication. The past studies mentioned in the table are conducted on CPU whereas the present work is carried out on two different GPUs; one being a low-end laptop GPU while other one being a high-end GPU. As shown in Table 2 even with a low-end GPU the number of rays being traced could be increased by 15x. Moreover, the high end GPU is able to trace 30 million rays in a second which means about 1500x increase in the number of rays being traced compared with the recent studies shown in the table.

However, it is important to indicate that the modern CPU hardware is much more powerful than the ones used in the past studies shown in the Table 2. Using a modern high end CPU, it is estimated that the studies mentioned in the table would result in a 15-20x increase in the number of rays traced. However, one should also consider that the computations on the CPU probably were conducted using double precision accuracy whereas in the current study on the GPUs single precision accuracy was

Table 2: Comparison of the number of rays traced per second

Compared cases	Hardware used	Number of rays traced per second (million)
Cosson et al.[54]	Intel Centrino CPU (2.6 GHz)	0.02
Mirhosseini and Saboonchi [17], 2011	Not specified / CPU	0.07
Present work	Nvidia Geforce 610M GPU	0.28
Present work	Nvidia TitanX	30

used. Conducting the same computations on CPU using single precision accuracy would mean a further 2x increase in the number of rays traced. So, taking all into account it is estimated that the implementation of Cosson et. al. [21] would run about 40-50x on todays high end multi-core CPUs using single precision accuracy. This would mean that the current implementation still be better by factor of 50x in terms of the number of rays being traced. It is noteworthy to indicate that this achieved speed up is due to the computational power of GPU only. The performance of the ray tracing method can be further improved by using the so called acceleration data structures.

3.1.2 Acceleration due to the efficient data structures

The computational cost of the ray tracing calculations depends both on the number of rays being traced and on the number of the triangles present in the domain. In the naive implementation the ray-triangle intersection calculations are carried out for each ray and for each triangle present in the domain regardless whether the ray will intersect or not. However, organizing the geometry data composed of triangles in a tree-like data structure can accelerate the intersection calculation massively. Different types of acceleration data structures exist but in this study the Bounding Volume Hierarchy (BVH), one of the most efficient acceleration data structures as discussed by Vinkler et al. [15], is considered. In order to investigate the effect of

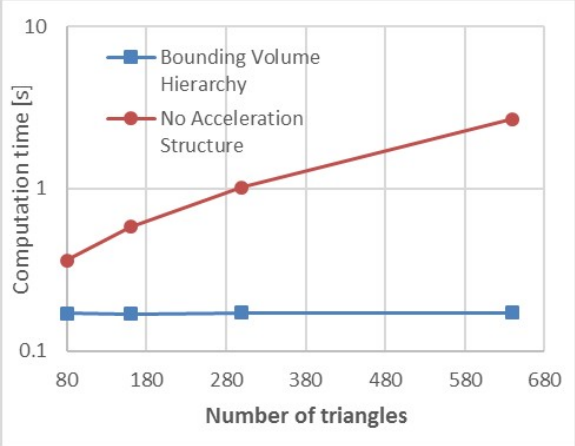

Figure 17: Number of triangles vs computation time to trace 100 million rays with and without acceleration structure

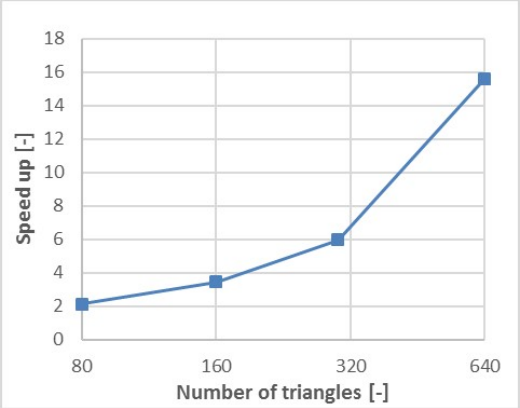

Figure 18: Speed up due to the acceleration structure vs number of triangles

the BVH data structure on the performance, a view factor calculation study has been conducted. For this study the view factor test case shown in Figure 16 is considered where the number of triangles the plate is made of has been progressively increased without altering its dimensions. The computation time to trace 100 million ray has been recorded for each case.

As shown in Figure 17 there is an almost 8x increase in the computation time as the number of triangles was raised by 8x. On the other hand, the calculation carried out with the BVH acceleration data structures seems to be almost completely unaffected by the number of triangles present in the domain. Figure 18 shows the gained speed up entirely due to the usage of the BVH. As shown, the speed-up increases dramatically with the number of the triangles in the domain which highlights the importance of the acceleration data structures especially for large view factor calculations.

3.1.3 Combined speed up

Combining the computational power of GPU and the acceleration data structure mentioned above very impressive speed-up values are achieved even with very low-end graphic cards. Table 2 shows the number of rays traced per second on the indicated

hardware compared. As compared to the studies on CPU, the current one is able to trace around 300 and 700 times more rays within the same time frame compared to the studies in [54] and in [17] respectively with low end GPU. Moreover, the results become even more dramatic with the high end GPU showing a speed-up in excess of 10,000x.

Table 3: Comparison of the number of rays traced per second rays with acceleration structure

Compared cases	Hardware used	Number of rays traced per second (million)
Cosson et al.[54]	Intel Centrino CPU (2.6 GHz)	0.02
Mirhosseini and Saboonchi [17], 2011	Not specified / CPU	0.007
Present work	Nvidia Geforce 610M GPU	5.2
Present work	Nvidia TitanX	250

3.2 Solution of radiative heat transfer problems involving complex geometries

3.2.1 Surface to surface radiative heat exchange

The numerical method is next tested for a scenario involving a complex three-dimensional geometry. In this regard, the view factor calculations is conducted between two glasses shown in Figure 19. The dimensions of the glasses are 6.5 x 13.7 cms. The surfaces are represented by 7056 surface triangles. Rays are emitted continuously over the randomly chosen surface elements and the obtained results are plotted after indicated ray numbers given in Figure 19.

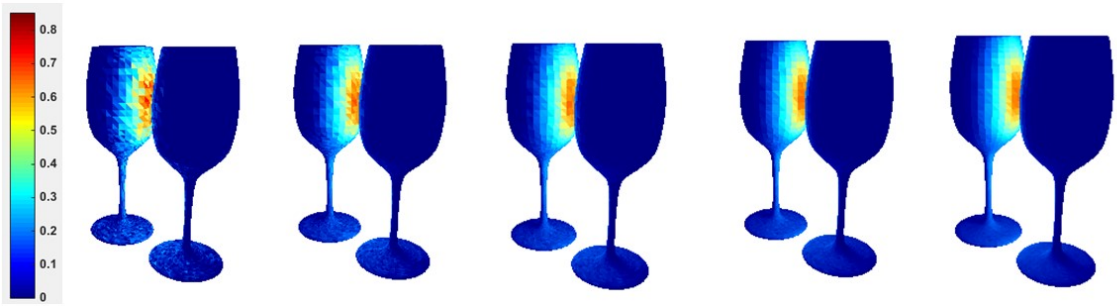

Figure 19: Map of view factor over the glass surface for different number of rays used (in millions left to right: 0.5 - 2.5 - 6.4 - 9.6 16). The hot points are becoming more clear and sharper with the increasing number of rays.

3.2.2 Radiative heat exchange in absorbing medium

After the evaluation of view factor, the numerical method is then tested for the more complicated case of participating medium consisting of a glass and a heater. A glass geometry is divided into 13343 tetrahedral elements. Figure 20 shows the cross sectional view of the mesh. The glass is made up of homogeneous pure absorbing medium. The temperature of the heater is fixed at 600 K and the initial temperature of glass is 400 K. Divergence for each sub-volume of a glass is first calculated using MCRT method. This divergence is then used to calculate the temperature of each cell using Equation 31 with Δt of 1 second. The map of the temperature of the glass after one second is shown in Figure 21 .

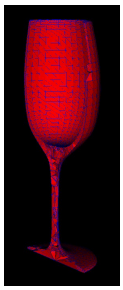

Figure 20: Volume mesh of the geometry.

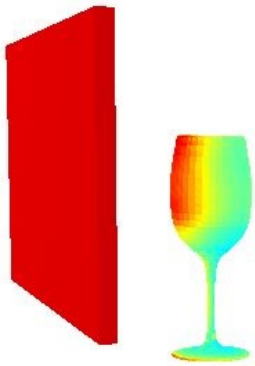

Figure 21: The temperature profile on the glass.

3.2.3 Radiative heating in presence of moving geometries

Continuous furnaces are widely used in a wide variety of manufacturing processes where the objects are moved through the furnace via a conveyor belt. They are especially suited for large scale production. Continuous furnaces are also used in glass annealing where the glass objects are fed into the furnace after they are taken out from the mold. After the glass object moves into the furnace, it is first heated to around 800 K and then kept at this temperature for a while and finally started to be cooled down initially very slowly and then at a faster rate. This is a critical stage in the glass manufacturing process. Therefore, modeling the process and calculating the glass temperatures are of vital importance for finding the right operating conditions of the furnace. The main challenging part in modeling of continuous furnaces is the

change in the geometry due to the relative movement between the glasses and the furnace walls.

The developed tool is tested using a simplified annealing furnace scenario as shown in Figure 22. The case consist of nine glasses moving. Initially, the temperatures of the glasses are set to 400K. The temperature of the heaters, placed on either side of the furnace, are set to 800K. Conveyor speed is adjusted to 0.03 m/s. Here, the glasses are treated as semi-transparent medium where the thermal radiation is penetrating into the object. Both the heaters and as well as the glass objects themselves are considered as sources. The time integration of the energy equation is done using first order Forward Euler method. A distinctive feature of the tool is the integration of the calculation and visualization stages. While the simulation is running the glasses are also simultaneously color coded with respect to their instantaneous temperature levels as shown in Figure 22. As shown, depending on the orientation of the glasses, the heating rate considerably varies.

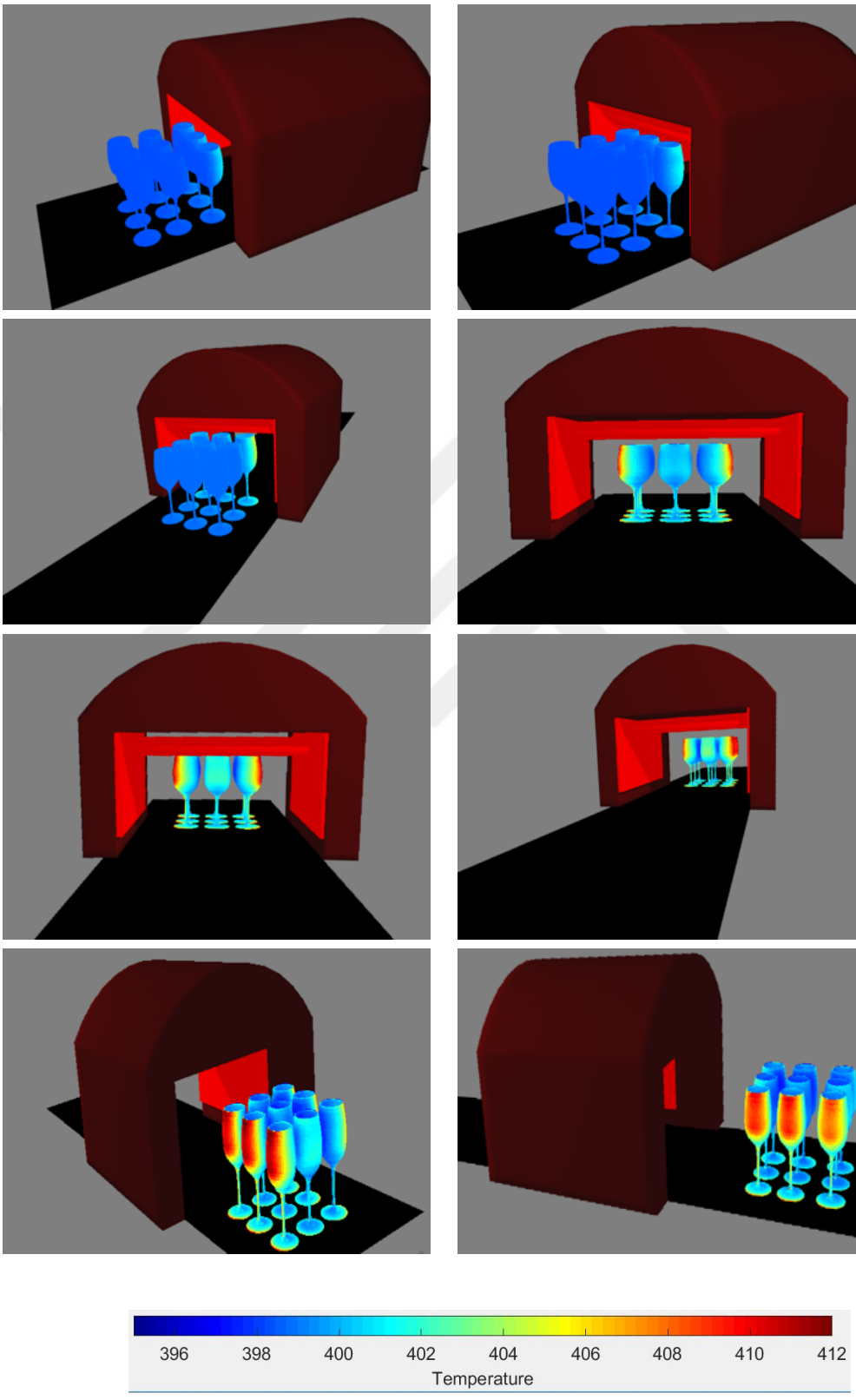

Figure 22: Simulation of the heat transfer of annealing process in glass manufacturing industry

3.3 Adaptive Monte Carlo ray tracing method

The iterative nature of the heat transfer problem might require solving the computationally costly RTE equation over and over again. Even though acceleration techniques are of great use, running MCRT multiple times can still be prohibitive. The main computational cost in solving the RTE lies in the calculation of the exchange factor matrix by the MCRT methods. Once the matrix is calculated, there is no more need to re-calculate it unless there is a change in the geometry or change in the radiative properties of the regions with temperature. So, once the matrix is calculated it can be stored in the memory and used in evaluating the divergence of radiative fluxes over and over again. At the end, one has to pay the computational price of the MCRT only once. However, for most realistic problems the exchange factor matrix ends up being very large in size. Problems involving only 1000 surface cells, leads to a matrix with 1000x1000 elements. Even though graphic cards have very high computational power, their memory space is rather limited. So, one does not have the memory resources to fit the exchange factor matrix into the memory of the graphic card even for small problems. The limitation in the memory requires the matrix to be re-calculated multiple times even though it might not be needed.

This section discusses a new MCRT based algorithm that balances the computational time and the memory usage of the hardware and the accuracy of the solution. By specifying the most critical regions of the domain before running the simulation, one can dedicate the computational resources on solving these particular regions very accurately and everywhere else at some mediocre accuracy. This is proposed as an engineered solution for the computational cost and accuracy problem.

The steps of the algorithm are :

1) Specify the critical regions within the domain that needs to be resolved at high accuracy

2) Calculate and store the exchange factors involving these critical regions only

3) Calculate and store divergence of the radiative fluxes for every cell within the domain

4) Update the temperature by solving the energy equation

5) Re-calculate the exchange factors involving the critical regions only and use them to correct the previous exchange factor values.

To test the effectiveness of the method, the proposed method is applied on a test scenario which consists of a stationary wine glass inside a hot furnace. The glass is considered as a pure absorbing and emitting medium with constant coefficients. Initially, the temperature of the glass is set to 573 K uniformly. The heaters at the walls of the furnace are taken as 800 K. The scenario is shown in the Figure 23. The heating of the glass in time is calculated using Monte Carlo based three different numerical methods:

1) Standard Monte Carlo ray tracing method

2) Adaptive Monte Carlo method (the proposed method in this section)

3) Standard Monte Carlo method with hundred times more number of rays, which serves as a reference solution.

Methods 1 and 2 use the same number of rays and method 3 uses an excessive number of rays and its solution will be taken as the reference solution to compare the accuracy of the order methods with. Figure 24 shows the time variation of the temperature of a single point of the glass object calculated with the methods mentioned above.

The results of the standard Monte Carlo ray tracing method (Method 1) fluctuates around the reference solution due to the relatively low number of rays used in the calculation. Method 3 does not exhibit the nonphysical fluctuations in temperature due to the large number of rays used in the calculation. But it also has a much higher computational cost. The Method 2 agrees quite well with the reference solution and

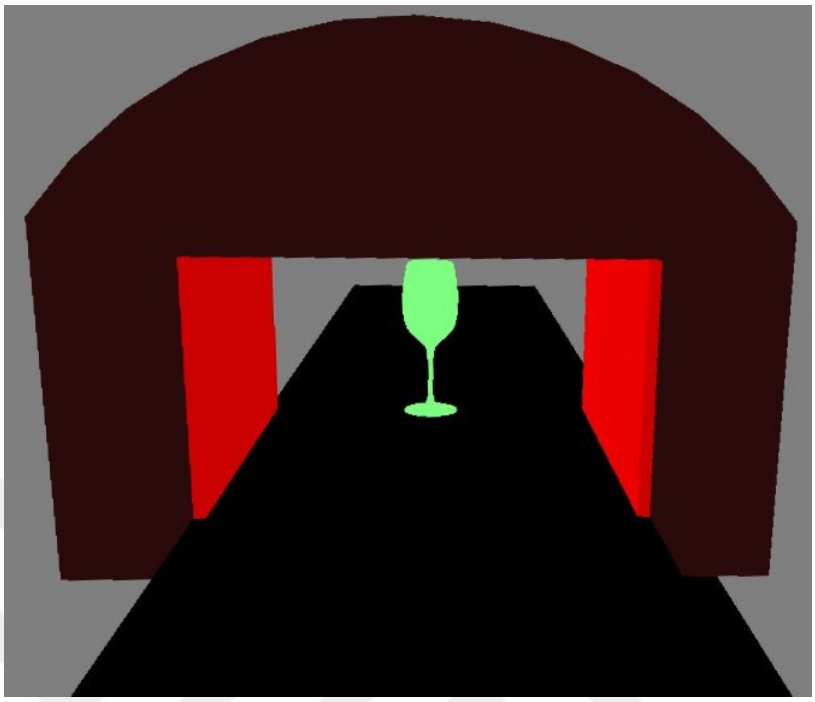

Figure 23: Test case for the Adaptive method

does not exhibit the oscillations of the Method 1 either. The rise in the accuracy of the Method 2 over the accuracy level of the Method 1 is achieved by targeting the computational resources on solving particular regions very accurately and solving everywhere else at the same level of accuracy of the Method 1.

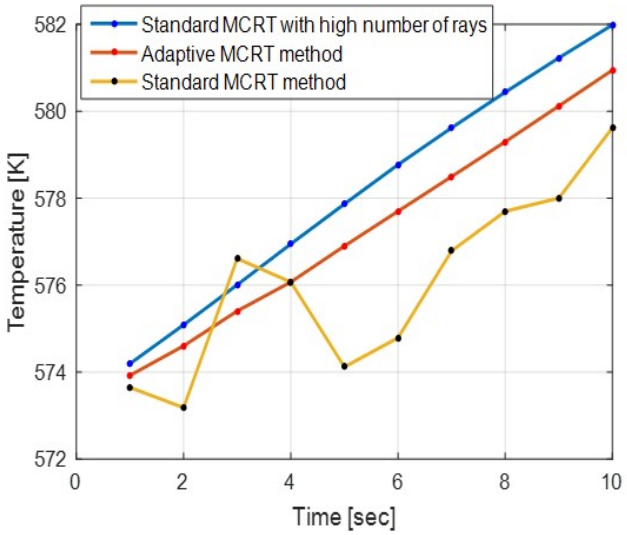

Figure 24: Increase in temperature of the glass with time

The additional accuracy that the Method 2 provides is due to the extra memory usage. The exchange factor values for a particular point are corrected over time as the simulation proceeds. Figure 26 shows the comparison of the memory usage for Method 1 and 2. As shown, Method 2 has a minor rise in memory usage as compared to the Method 1. This is due to saving of the exchange factor values for one particular point in the memory. Figure 27 shows the comparison of the computation time of each iteration for all the methods. As opposed to the difference in the memory usage, in terms of computation time, method 1 and method 2 behave the same due to the same number of rays utilized in both calculations, while method 3 takes more computation time due to the usage of high number of rays. The accuracy of the developed method is determined by using Equation 32. Figure 28 shows the comparison of the mean errors of adaptive MCRT and standard MCRT method.

$$MeanError = \frac{T_{calc} - T_{ref}}{time\ steps} \tag{32}$$

In summary, this method manages the computational resources to solve some critical regions of the problem more accurately. In a way one can compare this approach with the mesh refinement done in the critical regions of the domain in the discrete ordinate and finite volume based methods. As shown in Figure 25, designing a solver the interrelations between the computational resources and the accuracy should always be kept in mind. In this particular numerical method, high accuracy and less computation time can be achieved at the expense of huge memory usage. On the other hand, high accuracy and less memory consumption can be achieved with an expense of huge computation time. Considering these limiting conditions, the solvers has to be designed in a way all the interrelated aspects i.e computation power and memory usage and accuracy of the solution are balanced according to the needs and availability of the resources.

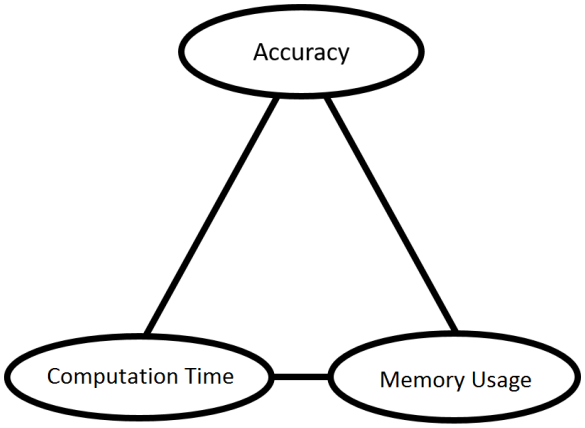

Figure 25: computation power triangle

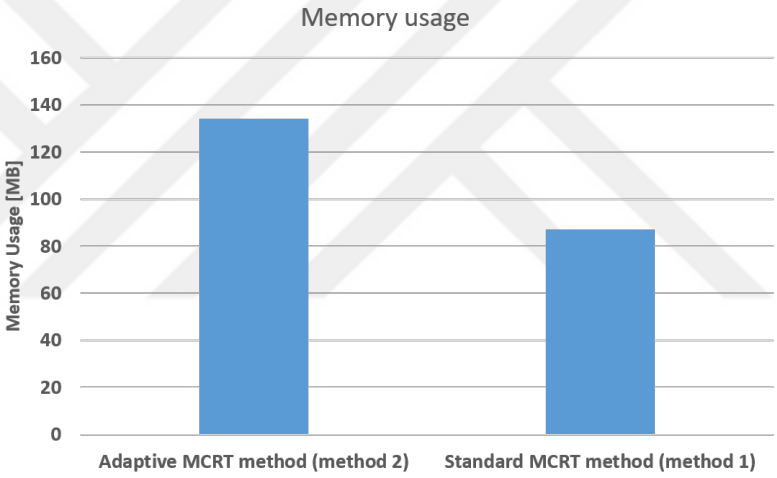

Figure 26: Comparison of the memory usage

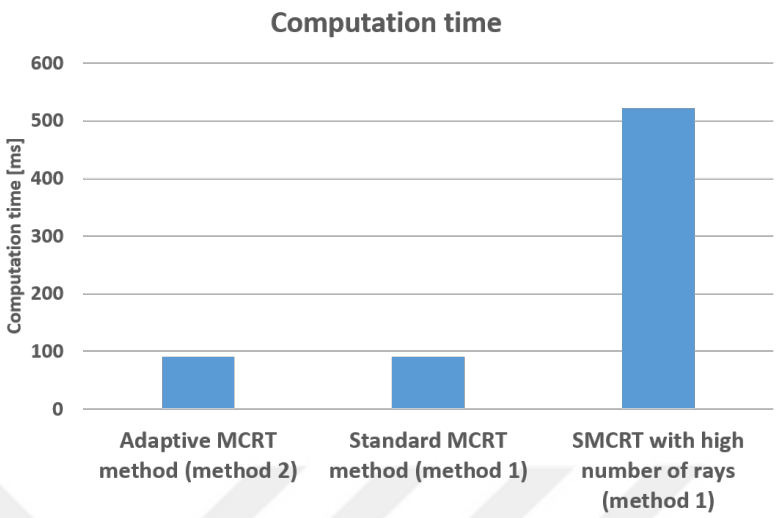

Figure 27: Comparison of the computation time

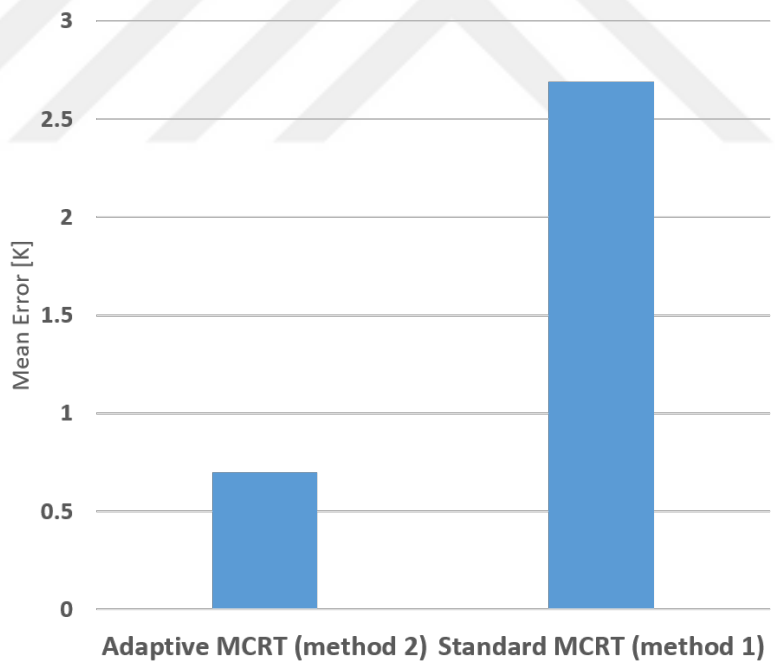

Figure 28: Comparison of the mean error

CHAPTER IV

CONCLUSION

The main conclusion of the study are listed below:

- A Monte Carlo ray tracing based radiation heat transfer solver has been developed. The tool has been validated and its capability on solving heat transfer problems with absorbingemitting (but not scattering) gray and non-participating media involving complex geometries has been demonstrated.

- With the use of graphic processors (GPU) the computational cost of the method is substantially reduced. It is calculated that a high-end GPU offers a performance benefit of approximately 40x in comparison to todays high-end multi-core CPU's. The developed tool takes advantage of the graphical processing capability of the GPU and is able to display the results simultaneously with the calculation.

- An additional 10x speed-up is achieved with the implementation of the acceleration data structure in ray tracing calculations.

- A new method is proposed that solves user defined critical regions with high accuracy and the remaining regions with mediocre accuracy. The method does not lead to any further increase in the computation time but it only requires 1.5x more memory for the storage of the exchange factors. The solver is developed in a way that the computational resources and the accuracy of the solution are balanced.

- Results of the present study reveal the potential of GPU computing in expanding

the applicability of the ray tracing based numerical methods for the solution of the radiative transfer equation.

Bibliography

[1] J. R. Howell, M. P. Menguc, and R. Siegel, *Thermal Radiation Heat Transfer*. CRC press, 2016.

[2] W. Fiveland, "Three-dimensional radiative heat-transfer solutions by the discrete- ordinates method," *Journal of Thermophysics and Heat Transfer*, vol. 2, no. 4, pp. 309–316, 1988.

[3] M. Mengüç and R. Viskanta, "Radiative transfer in three-dimensional rectangular enclosures containing inhomogeneous, anisotropically scattering media," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 33, no. 6, pp. 533–549, 1985.

[4] H. Hottel and E. Cohen, "Radiant heat exchange in a gas-filled enclosure: Allowance for nonuniformity of gas temperature," *AIChE Journal*, vol. 4, no. 1, pp. 3–14, 1958.

[5] G. Raithby and E. Chui, "A finite-volume method for predicting a radiant heat transfer in enclosures with participating media," *ASME, Transactions, Journal of Heat Transfer*, vol. 112, pp. 415–423, 1990.

[6] J. R. Howell, "Application of Monte Carlo to heat transfer problems," *Advances in heat transfer*, vol. 5, pp. 1–54, 1969.

[7] M. F. Modest, *Radiative heat transfer*. Academic press, 2013.

[8] J. T. Farmer, "Improved algorithms for monte carlo analysis of radiative heat transfer in complex participating media," 1995.

[9] J. R. Howell and M. P. Mengüç, "Radiative transfer configuration factor catalog: A listing of relations for common geometries," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 112, no. 5, pp. 910–912, 2011.

[10] R. Farrell, "Determination of configuration factors of irregular shape," *Journal of Heat Transfer*, vol. 98, no. 2, pp. 311–313, 1976.

[11] H. C. Hottel, "Radiant heat transmission," *Heat transmission*, vol. 3, 1954.

[12] W. Nusselt, "Grapische bestimmung des winkelverhaltnisses bei der warmestrahlung," *Zeitschrift des Vereines Deutscher Ingenieure*, vol. 72, no. 20, p. 673, 1928.

[13] M. F. Cohen and D. P. Greenberg, "The hemi-cube: A radiosity solution for complex environments," in *ACM SIGGRAPH Computer Graphics*, vol. 19, pp. 31–40, ACM, 1985.

[14] J. R. Howell, "The Monte Carlo method in radiative heat transfer," *Transactions-American Society of Mechanical Engineers Journal of Heat Transfer*, vol. 120, pp. 547–560, 1998.

[15] J. Chai, H. S. Lee, and S. V. Patankar, "Improved treatment of scattering using the discrete ordinates method," *Journal of Heat Transfer*, vol. 116, no. 1, pp. 260–263, 1994.

[16] A. Emery, O. Johansson, M. Lobo, and A. Abrous, "A comparative study of methods for computing the diffuse radiation viewfactors for complex structures," *Journal of heat transfer*, vol. 113, no. 2, pp. 413–422, 1991.

[17] M. Mirhosseini and A. Saboonchi, "View factor calculation using the monte carlo method for a 3d strip element to circular cylinder," *International Communications in Heat and Mass Transfer*, vol. 38, no. 6, pp. 821–826, 2011.

[18] A. Hajji, M. Mirhosseini, A. Saboonchi, and A. Moosavi, "Different methods for calculating a view factor in radiative applications: Strip to in-plane parallel semi-cylinder," *Journal of Engineering Thermophysics*, vol. 24, no. 2, pp. 169–180, 2015.

[19] G. Demirkaya, *Monte Carlo solution of a radiative heat transfet problem in a 3-D rectangular enclosure containing absorbing, emitting and anisotropically scattering medium.* PhD thesis, METU, 2003.

[20] J. C. Henson and W. Malalasekera, "Comparison of the discrete transfer and monte carlo methods for radiative heat transfer in three-dimensional nonhomogeneous scattering media," *Numerical Heat Transfer, Part A Applications*, vol. 32, no. 1, pp. 19–36, 1997.

[21] C. Loop and J. Blinn, "Real-time gpu rendering of piecewise algebraic surfaces," in *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 664–670, ACM, 2006.

[22] S. Halverson, "Energy transfer ray tracing with optix," 2012.

[23] X. He, E. Lee, L. Wilcox, R. Munipalli, and L. Pilon, "A high-order-accurate GPU-based radiative transfer equation solver for combustion and propulsion applications," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 63, no. 6, pp. 457–484, 2013.

[24] F. P. Siddiqui, A. M. Basol, and M. P. Menguc, "Ray tracing on graphic processors: Towards high fidelity radiative transfer solvers," in *ICHMT DIGITAL LIBRARY ONLINE*, Begel House Inc., 2016.

[25] H. Takizawa, N. Yamada, S. Sakai, and H. Kobayashi, "Radiative heat transfer simulation using programmable graphics hardware," in *Computer and Information Science, proceedings of the 5th IEEE/ACIS International Conference on Component-Based Software Engineering*, pp. 29–37, IEEE, 2006.

[26] D. S. Efremenko, D. G. Loyola, A. Doicu, and R. J. Spurr, "Multi-core-cpu and gpu-accelerated radiative transfer models based on the discrete ordinate method," *Computer Physics Communications*, vol. 185, no. 12, pp. 3079–3089, 2014.

[27] J. R. Howell and M. Perlmutter, "Monte Carlo solution of thermal transfer through radiant media between gray walls," *Journal of Heat Transfer*, vol. 86, no. 1, pp. 116–122, 1964.

[28] C. Usiskin and E. Sparrow, "Thermal radiation between parallel plates separated by an absorbingemitting nonisothermal gas," *International Journal of Heat and Mass Transfer*, vol. 1, no. 1, pp. 28–36, 1960.

[29] A. Usmani, "Handbook of numerical heat transfer," 1989.

[30] A. Appel, "Some techniques for shading machine renderings of solids," in *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, pp. 37–45, ACM, 1968.

[31] J. T. Kajiya, "The rendering equation," in *ACM Siggraph Computer Graphics*, vol. 20, pp. 143–150, ACM, 1986.

[32] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[33] C. Ericson, *Real-time collision detection*. CRC Press, 2004.

[34] M. Vinkler, V. Havran, and J. Bittner, "Bounding volume hierarchies versus kd-trees on contemporary many-core architectures," in *Proceedings of the 30th Spring Conference on Computer Graphics*, pp. 29–36, ACM, 2014.

[35] J. Fung, F. Tang, and S. Mann, "Mediated reality using computer graphics hardware for computer vision," in *Wearable Computers, 2002.(ISWC 2002). Proceedings. Sixth International Symposium on*, pp. 83–89, IEEE, 2002.

[36] "World's first gpu by nvidia." `http://www.nvidia.com/object/IO_20020111_5424.html`.

[37] P. Du, R. Weber, P. Luszczek, S. Tomov, G. Peterson, and J. Dongarra, "From CUDA to OpenCL: Towards a performance-portable solution for multi-platform GPU programming," *Parallel Computing*, vol. 38, no. 8, pp. 391–407, 2012.

[38] V. Garcia, E. Debreuve, and M. Barlaud, "Fast k nearest neighbor search using gpu," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pp. 1–6, IEEE, 2008.

[39] R. Wilson, "Dsp brings you a high-definition moon walk," *EDN. Retrieved*, vol. 3, 2009.

[40] K. Crane, I. Llamas, and S. Tariq, "Real-time simulation and rendering of 3d fluids," *GPU gems*, vol. 3, no. 1, 2007.

[41] M. J. Harris, "Fast fluid dynamics simulation on the gpu.," in *SIGGRAPH Courses*, p. 220, 2005.

[42] K. S. Hasan, A. Chatterjee, S. Radhakrishnan, and J. K. Antonio, "Performance prediction model and analysis for compute-intensive tasks on gpus.," in *NPC*, pp. 612–617, 2014.

[43] T. Hamada and K. Nitadori, "190 tflops astrophysical n-body simulation on a cluster of gpus," in *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–9, IEEE Computer Society, 2010.

[44] S. A. Manavski and G. Valle, "CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment," *BMC bioinformatics*, vol. 9, no. 2, p. S10, 2008.

[45] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda," *Queue*, vol. 6, no. 2, pp. 40–53, 2008.

[46] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, *et al.*, "Optix: a general purpose ray tracing engine," in *ACM Transactions on Graphics (TOG)*, vol. 29, p. 66, ACM, 2010.

[47] E. Sparrow, "A new and simpler formulation for radiative angle factors," *Journal of Heat Transfer*, vol. 85, no. 2, pp. 81–87, 1963.

[48] F. Zafar, M. Olano, and A. Curtis, "GPU random numbers via the tiny encryption algorithm," in *Proceedings of the Conference on High Performance Graphics*, pp. 133–141, Eurographics Association, 2010.

[49] A. S. Glassner, "An introduction to ray tracing academic," *New York*, 1989.

[50] "A catalog of radiation configuration factors." `http://www.thermalradiation.net/indexCat.html`.

[51] A. Feingold and K. Gupta, "New analytical approach to the evaluation of configuration factors in radiation from spheres in infinitely long cylinders," ASME, 1969.

[52] P.-f. Hsu and J. Farmer, "Benchmark solutions of radiative heat transfer within nonhomogeneous participating media using the monte carlo and yix method," *Journal of heat transfer*, vol. 119, no. 1, pp. 185–188, 1997.

[53] Z. Guo and S. Maruyama, "Radiative heat transfer in inhomogeneous, nongray, and anisotropically scattering media," *International Journal of Heat and Mass Transfer*, vol. 43, no. 13, pp. 2325–2336, 2000.

[54] B. Cosson, F. Schmidt, Y. Le Maoult, and M. Bordival, "Infrared heating stage simulation of semi-transparent media (pet) using ray tracing method," *International Journal of Material Forming*, vol. 4, no. 1, pp. 1–10, 2011.

[55] F. Lockwood and N. Shah, "A new radiation solution method for incorporation in general combustion prediction procedures," in *International Symposium on Combustion*, vol. 18, pp. 1405–1414, Elsevier, 1981.

[56] D. Klein, "Coupled radiative and conductive heat transfer in a two-dimensional rectangular enclosure with gray participating media using finite elements," *Journal of heat transfer*, vol. 106, p. 613, 1984.

[57] Z. Tan and J. R. Howell, "New numerical method for radiation heat transfer in nonhomogeneous participating media," *J. Thermophys. Heat Transfer*, vol. 4, no. 4, pp. 419–424, 1990.

[58] T. Whitted, "An improved illumination model for shaded display," in *ACM Siggraph 2005 Courses*, p. 4, ACM, 2005.

[59] S. Mittal and J. S. Vetter, "A survey of cpu-gpu heterogeneous computing techniques," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, p. 69, 2015.

[60] J. Fung and S. Mann, "Computer vision signal processing on graphics processing units," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 5, pp. V–93, IEEE, 2004.

VITA

Mr. Faizan Siddiqui has completed his BE degree in Electronics Engineering from NED University, Karachi, Pakistan in 2013. He joined Ozyegin University in 2015 and started his MS degree in Electrical and Electronics Engineering under the supervision of Assistant Professor Altug M. Basol and Professor M. Pinar Menguc. His research interests include radiation heat transfer and GPU-accelerated computing.