KERNEL DENSITY ESTIMATOR METHODS FOR MONTE CARLO RADIATION TRANSPORT

by

Kaushik Banerjee

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in The University of Michigan
2010

Doctoral Committee:

Professor William R. Martin, Chair
Professor James Paul Holloway
Professor Edward W. Larsen
Professor Robert M. Ziff

To my ma and baba (dad), Krishna and Rabindra Nath Banerjee
And
To my advisor, William R. Martin

Acknowledgements

First of all, I thank my advisor, Professor William R. Martin, for his continuous support, guidance, encouragements and patience. His deep insight, vast knowledge and perseverance will always be the best source of inspiration for me. I would also like to thank him for providing me freedom to pursue my research in my own way. I feel truly privileged to have him as my advisor. I also wish to thank Professor Edward Larsen, James P. Holloway and Robert M. Ziff for serving on my doctoral committee. I thank Professor Holloway for his valuable initial feedback on my work. I consider myself fortunate for getting the rare opportunity of learning numerical transport theory from Professor Larsen. His transport theory lectures and home works are something hard to describe but have to experience. I would also like to extend my thanks to Professor John C. Lee for many helpful and intriguing research discussions, while working under him on a different project.

This study was supported by the DOE NERI grant DE-FC07-06ID14745.

I would also like to thank Shikha Prasad, Emily Wolters, Eva Sunny, Gokhan Yesilyurt, Kyeong Oh, Jishnu and Ananya Bhattacharya, Bibhas Chakraborty, Bodhisattva Sen, Debadyuti Roy, Saikat Roy, Jeremy Conlin, Troy Becker, Greg Davidson, Peggy Gramer, David Griesheimer, Jinan Yang, Wei Ji and many other unnamed friends, mentors at Michigan for their friendship and support.

I will always be indebted to my cousin and sister-in-law, Somnath and Shampa Mukhopadhyay, for their love and support. They always make me feel at home in Michigan and make my life less stressful in several ways.

I am most grateful to my family, my parents and wife, Sriparna, for all their support. Without their encouragements and patience this thesis would not have existed. Words

alone can not express my feelings for them. Everything that I have accomplished has been possible for them.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

The expected behavior of particles moving freely in space is governed by the linear Boltzmann transport equation [1]. In nuclear engineering many important problems can be categorized as particle transport problems and therefore can be formulated by the Boltzmann equation. These particle transport problems include: (a) shielding design, where the design parameters of a shield, used for stopping radiation to move from one side to the other, are desired, (b) source detector problem, which is used to design radiation detector for detecting and measuring radiation emitted by radiation sources, (c) criticality, in which the eigenvalue of the system is estimated and is used for reactor design.

In the radiation transport process particles are born from radiation sources and travel stochastically through the problem medium. The transport process consists of birth of a particle from a radiation source, collision events, in which either the particle changes its direction and energy (scattering) or the particle is absorbed (absorption), streaming in space between scattering events and leaking out of the system through the external boundaries. The Boltzmann transport equation contains terms which represent radiation source, collisions, scatterings and streaming in the system and out of the system. The expected particle distributions within the system of interest can be obtained by numerically solving the Boltzmann transport equation subject to suitable initial and boundary conditions.

Two general types of methods, namely deterministic methods and Monte Carlo method, are used to solve the Boltzmann transport equation. Deterministic methods solve the Boltzmann equation by discretizing the phase space and forming a system of algebraic

equations that can be evaluated iteratively [2]. However, these discretization schemes introduce truncation error, and many small geometric details that cannot be represented by the discretization schemes are not accounted for. In addition, for the deterministic calculation of a full core configuration, a series of assembly level transport calculations are performed to generate group-collapsed homogenized cross sections, which are then used as input into a global code, typically a nodal diffusion code. Generating homogenized cross sections is a complicated task.

On the other hand, Monte Carlo simulations are nothing but stochastic numerical experiments. Instead of numerically solving the Boltzmann transport equation for the expected particle distribution at all locations in the phase space, Monte Carlo calculations simulate the actual physical system by analyzing the transport of a finite number of particles that move through it. This procedure yields a number of stochastic particle trajectories that could occur in the system. Statistical inference about the system parameters and phase space integrated quantities can be drawn from a statistically significant number of particle tracks generated inside the system. Hence, a solution obtained by a Monte Carlo simulation is actually a statistical estimate based on a finite number of realizations of the expected behavior of particles in the system. Thus the solutions from a Monte Carlo simulation are stochastic and correct only within statistical uncertainties.

In Monte Carlo simulations each particle history can be considered as an independent Markov chain. In a Markov process, the future behavior of a particle history depends only on its present state [3]. This particular property helps to minimize computational work required to track a particle during its random walk. There are several advantages of Monte Carlo method over deterministic methods. In Monte Carlo, complicated geometries and complex physics can be modeled exactly. Great research efforts are involved to incorporate CAD (computer aided design) with the Monte Carlo geometry input routines, such that Monte Carlo simulations can be executed directly from the design drawing without any approximation. In addition, Monte Carlo can solve the neutron transport problem in its continuous energy form, and there is no need to calculate the group-collapsed cross sections. Although Monte Carlo requires more computer time

compared to deterministic calculations, Monte Carlo simulations can be easily implemented on massively parallel processors and has been described as “embarrassingly parallel”.

In this thesis, we address three different Monte Carlo radiation transport problems and develop a new approach to addressing these problems. These Monte Carlo problems are: (a) global tally distributions inside the system, which are conventionally carried out with histogram estimators, (b) Monte Carlo tallies with unbounded variance, where the conventional approach is to use approximate estimators near the singularity, hence introducing a bias, and (c) criticality source convergence problems with high dominance ratios, where a large number of iterations may be needed to converge the fission source. These problems will be described in more detail later in this chapter. A nonparametric statistical probability density estimator, known as the kernel density estimator (KDE), is applied to develop novel solutions to these problems. This introductory chapter presents Monte Carlo particle transport method in general and an overview of this thesis work.

1.2 The Monte Carlo Method

The particle transport process can be considered as a sequence of stochastic events. A particle’s life start by its birth from a given radiation source with the initial or starting location, direction of travel and energy stochastically selected from that given source. The new born particle travels a random distance inside the system until it suffers from a collision event or leaves the system. Collisions can be of two types, scattering and absorption. Scattering collisions provide the particle a new set of direction and energy. The absorption collision on the other hand may produce progenies or simply terminate the particle’s life. This process continues until the particle is absorbed or lost from the system. This entire life of a particle is known as particle history.

The physical process described above is governed by the steady state Boltzmann transport equation [1] and is given by

$$\begin{aligned}
&\underline{\Omega}.\underline{\nabla}\psi\left(\underline{x},\underline{\Omega},E\right)+\Sigma_t\left(\underline{x},E\right)\psi\left(\underline{x},\underline{\Omega},E\right)\\
&\qquad=\int_0^\infty\int_{4\pi}\Sigma_s\left(\underline{x},\underline{\Omega}'.\underline{\Omega},E'\to E\right)\psi\left(\underline{x},\underline{\Omega}',E'\right)d\Omega'dE'+\frac{Q\left(\underline{x},E\right)}{4\pi},\\
&\qquad\underline{x}\in V,\ 0<E<\infty,\ \underline{\Omega}\in 4\pi,\\
&\psi\left(\underline{x},\underline{\Omega},E\right)=\psi^b\left(\underline{x},\underline{\Omega},E\right),\ \underline{x}\in\partial V,\ 0<E<\infty,\ \underline{\Omega}.\underline{n}<0,
\end{aligned}\tag{1.1}$$

where $\underline{x}$ is the coordinates of the location of the particles, $\underline{\Omega}$ is the unit direction vector, and E is the particle energy. The system volume is given by V and $\underline{n}$ is the outward unit surface normal vector. ψ denotes the angular flux, Σ_t is the total macroscopic cross section, Σ_s is the differential scattering macroscopic cross section and Q is the isotropic external source. ψ^b represents the boundary condition and can be a vacuum boundary, reflecting boundary or a prescribed incident flux. For a criticality problem, $Q\left(\underline{x},E\right)$ is replaced by the fission source:

$$Q\left(\underline{x},E\right)=\frac{\chi\left(\underline{x},E\right)}{k}\int_0^\infty\int_{4\pi}\upsilon\Sigma_f\left(\underline{x},E'\right)\psi\left(\underline{x},\underline{\Omega}',E'\right)d\Omega'd\ E'\tag{1.2}$$

Σ_f is the macroscopic fission cross section, υ is the average number of neutrons produced per fission and χ is the fission energy spectrum. The quantity k is the eigenvalue, known as the criticality of the system, and the corresponding ψ is the eigenfunction. k is used to maintain the balance between neutron loss and gain in the steady state Boltzmann transport equation.

The unknown angular flux ψ is of primary interest as it can be used to compute reaction rates and leakage rates. For example, we often calculate an integral quantity of ψ such as a detector response,

$$R=\int_V\int_{4\pi}\int_0^\infty\Sigma\left(\underline{x},E\right)\psi\left(\underline{x},\underline{\Omega},E\right)d\Omega dEdV,\tag{1.3}$$

where V is the detector volume and Σ is the response function. Instead of solving the Boltzmann equation directly, Monte Carlo simulates various realizations of the actual problem. These collective realizations are then used to estimate average values of some

system parameters and the variance of those averages. Comprehensive overviews of the mathematical aspects of Monte Carlo and its applications to general problems are documented by Hammersley and Handscomb [4], Kalos and Whitlock [5], and Spanier and Gelbard [6]. The monographs by Carter and Cashwell [7] and Lux and Koblinger [8] cover the particle transport application of the general Monte Carlo method.

To simulate a single realization of the problem, Monte Carlo starts by determining a particle's position in space, initial direction of travel, and the initial energy by random sampling from the given source distribution (criticality problems start with a source guess). Next, the distance to the next collision is sampled. The probability that a particle will collide in ds about s is given by [7]

$$f(s)ds = e^{-\Sigma_t s}\Sigma_t ds. \tag{1.4}$$

The total cross section is assumed constant along the trajectory for convenience. Applying the inverse cumulative distribution method [8], the distance s to the next collision can be sampled by

$$s = -\frac{1}{\Sigma_t}\ln(1-\xi), \tag{1.5}$$

where ξ is a random number uniformly distributed between 0 and 1. Hence $1-\xi$ is also uniformly distributed between 0 and 1, and we can rewrite Eq. (1.5) as

$$s = -\frac{1}{\Sigma_t}\ln(\xi). \tag{1.6}$$

If the distance to the collision is greater than the distance to the nearest boundary in the direction of flight then the particle is moved to the boundary and the distance to the next collision is resampled by using Eq. (1.6) and the new Σ_t of the region on the other side of the boundary.

If a collision occurs, then the type of collision is sampled from a discrete probability density function (pdf) using the macroscopic cross sections for each interaction [8]. If the collision type is absorption, then the particle history is terminated. For scattering

collisions a new direction of flight and new energy are sampled from their corresponding pdfs and the particle moves to the next collision or boundary.

In addition to stochastically following the particles inside the system, we need to extract desired information from the Monte Carlo simulation. This job is performed by Monte Carlo tallies. Conceptually Monte Carlo tallies perform counting experiments. During each realization or history, particular events, such as collisions or boundary crossings are used to accumulate the score for the corresponding tally. At the end of the simulation the probability of a desired event can be obtained by dividing the accumulated score for that event by the total number of particles run. Tallies can be used to estimate different quantities of interest in particle transport, such as volume averaged flux in a region, average flux over a surface, or current integrated over a surface.

Monte Carlo tallies may be viewed as statistical estimators of particle behaviors obtained from collective realizations or histories of the actual physical problem. Since tallies represent statistical estimators, it is also possible to estimate the tally variance. The tally variance is inversely proportional to the number of particles run in the simulation. In the limit of an infinite number of particles the Monte Carlo estimate will converge to the true solution of the Boltzmann equation. However, running a large number of particles is computationally prohibitive. Another problem of Monte Carlo is undersampling of a particular region that could result in a biased estimate. Several variance reduction techniques [8] have been developed, which could potentially reduce the solution variance while reducing the number of particles required to be run.

Monte Carlo is becoming the method of choice among nuclear professionals, due to its accuracy and flexibility. However, there are many areas of Monte Carlo radiation transport that could be improved. This thesis is an effort to formulate new and effective solutions of three such areas. These three Monte Carlo challenges will be described next.

1.3.1 Conventional Monte Carlo Tallies

Carter and Cashwell [7] categorized Monte Carlo tallies into four basic type of estimators, namely collision estimators, last-event estimators, track length estimators, and

next-event (point detector) estimators. Collision, last-event, and next-event estimators are discrete estimators and score only when there is a certain type of event. On the other hand, the track length estimator scores along the entire neutron trajectory. All these traditional Monte Carlo tallies perform satisfactorily for estimating integral quantities. However, conventional Monte Carlo tallies can not easily extract the shape information of the underlying and unknown distribution function that represents a given tally from the random walk. The conventional approach to obtaining shape information employs histograms.

To construct a histogram, the phase space variables are divided into a specified number of bins. Monte Carlo simulation estimates integrated quantities, such as flux or reaction rate, inside each bin. The histogram approximation has several limitations. First, the histogram is a discontinuous step approximation to the actual distribution. Secondly, the degree of smoothness depends on the number of bins. However, a large number of bins reduces the number of particles that score in an individual bin, resulting in an increase in the variance of the estimate in each bin. Finally, histogram tallies depend on the choice of origin. Changing the origin may reveal or obscure certain features of the unknown density function.

In the first part of this thesis, we develop continuous, mesh-free global flux tallies by using the kernel density estimator (KDE) [9], a nonparametric statistical density estimator. These global tallies are based on both collision and track length estimators and are capable of estimating flux at any point inside the system.

1.3.2 Monte Carlo Tallies with Unbounded Variance

There are two popular Monte Carlo estimators with singularities. These estimators are the point detector or next-event estimator (e.g., F5 estimator in MCNP5) and the surface crossing flux estimator (e.g, F2). The point detector tally is used to estimate the scalar flux at a point. The point detector estimator is widely used, especially in shielding calculations, but because of the singularity in the estimator, it can yield incorrect or misleading results if not used carefully. The singularity in the point detector tally has the form $1/r^2$, where r is the distance between a collision or source location and the

location of the point detector. In Monte Carlo simulations this singularity manifests itself when a source or collision occurs arbitrarily close to the detector point. It can be shown that, as a result of this singularity, the point detector estimator exists but its variance is infinite [10]. The conventional way of removing this singularity is to employ an approximate estimator near the singularity [11], which generally introduces a bias in the estimator.

The surface crossing flux estimator is used to estimate the average flux over a surface. This is another very important estimator and has extensive application in many nuclear engineering applications. This estimator scores the inverse of the absolute value of cosine of the angle between the direction of flight and the surface normal ($|\mu|$), each time a particle crosses the surface of interest. Similar to the point detector estimator, the surface crossing flux estimator exists (i.e., its mean is unbiased and finite), but its variance is infinite. In Monte Carlo simulations, this estimator becomes unreliable when particles cross the surface of interest in grazing angles and contribute large scores to the estimator. The conventional approach to address this deficiency is to score the expected value of the quantity $1/|\mu|$ for values of $|\mu|$ near $\mu = 0$. However, this introduces a bias in the estimator.

An alternative approach is developed in this thesis for both the point detector and surface crossing flux estimators by using KDE. Based on the observation that KDE estimates the underlying PDFs for the particle interactions (i.e., collisions or surface crossings) that are scored for these tallies, variance reduction methods have been developed for both tallies that yields 1/N convergence for their variances while maintaining unbiased (at least asymptotically) estimators.

1.3.3 Monte Carlo Fission Source Iteration

Monte Carlo criticality calculations are very important in the nuclear engineering community and are widely used by reactor design and waste management groups. Criticality calculations are initiated with an arbitrary guessed source distribution. This initial fission source distribution could be a point source or a uniform source, for

example. In criticality calculations, the total number of neutron histories is divided into N cycles or batches or generations with M neutrons per cycle. M neutrons are sampled from the initial source distribution with their initial location, direction of flight, and energy, and they are then followed through their random walk within the system. If a fission event occurs, that location is stored in a place (array) known as fission bank. All these stored locations are used as the fission source for the next cycle. This iterative process is continued until all N batches are completed.

Since the initial neutron source is a guess, the first few initial batches are used to converge the actual fission source distribution, within statistical fluctuations. These first few batches are discarded and are not used for the batchwise tallying of any important parameters like the k-eigenvalue of the system. The discarded batches are known as inactive batches, and batches thereafter are called active batches. The final k-eigenvalue is calculated as the average over all the active batches.

Depending on the system, sometimes we require many cycles to converge the source reasonably. This makes the Monte Carlo criticality calculation computationally prohibitive. This slow convergence of fission source is mainly a characteristic of loosely coupled systems with high dominance ratio. The dominance ratio is defined by the ratio of the second and first eigenvalues of the system. In this thesis, we propose a novel technique of sampling the next generation fission sites from the previous generation by using KDE-based sampling. This new sampling of the fission bank method is able to achieve a significant speed up in fission source convergence. In the conventional fission bank method, since the histories in a previous generation are used to determine the fission source in the next generation, the Monte Carlo samples are correlated. In the KDE method, as the next generation fission sites are actually sampled from the estimated fission source distribution, this method may also reduce the cycle to cycle correlation in the fission source iteration.

1.4 Thesis Outline

The purpose of this thesis is to apply the kernel density estimator to different Monte Carlo radiation transport related problems. We now provide an outline of the rest of the thesis.

Chapter 2: Kernel Density Estimator and its Properties

In this chapter, we give a comprehensive overview of the probability density estimation technique by using kernel density estimator with numerical examples. We describe both the univariate and multivariate kernel density estimator. The statistical properties of the kernel density estimator are also reviewed in detail.

Chapter 3: Kernel Density Estimated Monte Carlo Tallies

This chapter introduces two new global KDE tallies, based on the well-known collision and track length tallies. These mesh-free tallies are capable of estimating the shape of a tally inside the problem domain without any bin structure. We demonstrate the effectiveness of these tallies over conventional histogram based Monte Carlo tallies with a set of one dimensional and multi dimensional criticality problems. The convergence properties of KDE tallies are also compared, both analytically and numerically, with two existing Monte Carlo tallies, namely the conventional histogram tally and the functional expansion tally (FET) [12].

Chapter 4: Kernel Density Estimation Method for Monte Carlo Tallies with Unbounded Variance

In this chapter, a kernel density estimator based method is proposed to remove the singularities from the point detector and surface crossing flux tallies. Due to the singularities, both these tallies have unbounded variance and the variance can not be used to measure the convergence of these tallies. The performance of this KDE based method over the conventional approximate methods is impressive, yielding $1/N$ variance convergence for both tallies, where N is the number of histories. Numerical examples are used to demonstrate the effectiveness of this method.

Chapter 5: Kernel Density Estimation Method for Monte Carlo Eigenvalue Calculations

The KDE sampling method is used to sample fission locations from the fission bank. The conventional fission bank method converges slowly to the actual fission source distribution for loosely coupled systems. The KDE based method significantly improves the fission source convergence. This KDE based fission source sampling method is demonstrated by one dimensional and three dimensional criticality problems.

Chapter 6: Summary and Future Work

This final chapter includes a summary of the major results of this thesis and a brief overview of the future work needed in this area.

1.5 References

[1] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*, John Wiley & Sons (1976).

[2] E. E. Lewis and W. F. Miller, Jr., Computational methods of Neutron Transport, American Nuclear Society, La Grange Park, Illinois (1993).

[3] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, Fourth ed, Tata McGraw-Hill, New Delhi (2002).

[4] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, John Wiley & Sons, New York (1964).

[5] M. H. Kalos and P. A. Whitlock, Monte Carlo Methods, Volume I: Basics, John Wiley & Sons, New York (1986).

[6] J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*, Dover Publications, Mineola, New York (2007).

[7] L. L. Carter and E. D. Cashwell, *Particle Transport Simulation with the Monte Carlo Method*, Technical Information Center, Energy Research and Development Administration, Oak Ridge (1975).

[8] I. Lux and L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, CRC Press, Boca Raton, Florida (1991).

[9] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London (1986).

[10] M. H. Kalos, "On the Estimation of Flux at a Point by Monte Carlo," *Nuclear Science and Engineering*, **16**, 111-117, (1963).

[11] S. A. Dupree and S. K. Fraley, *A Monte Carlo Primer – A Practical Approach to Radiation Transport*, Plenum, New York (2002).

[12] D. P. Griesheimer, *Functional Expansion Tallies for Monte Carlo Simulations*, Ph.D. Thesis, Nuclear Engineering and Radiological Sciences, University of Michigan (2005).

CHAPTER 2

Kernel Density Estimator and Its Properties

2.1 Introduction

The random variable and its corresponding probability density function (pdf) are the fundamental concepts in statistics. A random variable is a function that associates a real number to every unique outcome from a random experiment. Each time the random experiment is run, the random variable is evaluated and returns a number that describes the outcome. Each value produced in this way is referred to as a realization of the random variable. In other words for every outcome ξ of an experiment specified by the space S, a number is assigned by $\mathbf{x}(\xi)$. This function $\mathbf{x}$ with domain the set S and range a set of real numbers is called a random variable. The probability that a single realization of a random variable will take on a specific value or fall between a range of values in the continuous case is governed by the parent distribution of the random variable, denoted by $f(x)$ for the random variable $\mathbf{x}$. The function $f(x)$ is known as the probability density function. For any random variable $\mathbf{x}$, the probability of events $(a < \mathbf{x} \le b)$ is given by

$$P(a < \mathbf{x} \le b) = \int_a^b f(x)dx, \quad a < b. \tag{2.1}$$

For further details on the general theory and applications of random processes, the text by Papoulis [1] is a well known reference in this field. A pdf can be obtained in two different ways. If the physical laws governing the random process are known, a pdf can be defined theoretically. However if we do not know the exact theory a priori, then we can estimate the probability density function from the realizations of the underlying pdf, which may be an implicit function of a physical process, such as particle transport. There are two approaches to density estimation, namely parametric and nonparametric. If the shape of the pdf to be estimated is known beforehand, parametric estimation can be used. In

parametric estimation we assume the realizations or the observed data are from a known parametric family of distributions like the normal distribution with mean μ and variance σ^2. The true density $f(x)$ of the observed data could then be estimated by finding estimates of the parameters μ and σ^2. However if the shape of the underlying density function is unknown, nonparametric density estimation techniques have to be employed. In other words for nonparametric density estimation, data are allowed to speak for themselves. This thesis is based on a nonparametric density estimation technique, the kernel density estimator (KDE). The KDE is used to formulate alternative approaches for three distinct and existing Monte Carlo radiation transport problems, which are discussed in Chapter 1 and will be discussed in detail in Chapters 3, 4, and 5.

There is a vast literature on density estimation. The earliest known reference to KDE is a 1951 report by Fix and Hodges [2]. The first published paper to deal explicitly with KDE-like estimator was by Rosenblatt [3] in 1956. The next known reference of the KDE was by Parzen [4] in 1962. Since then, thousands of articles have been written about kernel density estimators. Some of them will be referred in the following sections of this chapter and also in the subsequent chapters. Kernel density estimators are also treated in several books. The text book by Silverman [5] is widely considered as the definitive reference in this field. Silverman has described the sample properties of KDE in great detail in his book with several illustrations. Other readable books on KDE are by Devroye and Györfi [6], Härdle [7], Wand and Jones [8], and Bowman and Azzalini [9]. The theory about the statistical density estimation techniques discussed in the following sections are mainly based on the textbooks by Silverman, Wand and Jones, Härdle, and Bowman and Azzalini.

Previous application of KDE to nuclear engineering have been to resample electron trajectories as a part of a response matrix approach by Du and Martin [10], and to resample photon trajectories in the phase space resulting from the patient-independent portion of a radiation transport calculation as a part of cancer treatment plan by Tyagi et al. [11]. Though there is a wide range of nuclear engineering applications of histogram estimation, which is actually a primitive nonparametric density estimator, sophisticated

density estimators have not been explored to a great extent in nuclear engineering. One notable exception is the functional expansion tally (FET), developed by Griesheimer [20], which uses an orthogonal series estimator. This thesis will discuss application of another nonparametric density estimator, viz. KDE, to nuclear engineering applications.

Before we discuss the application of the kernel density estimator to problems in Monte Carlo simulations, we will discuss in Section 2 the evolution of the kernel density estimator. In Section 3, the univariate KDE and its properties are discussed in detail. Bandwidth calculation and the boundary problem associated with the KDE are investigated in this section. A numerical example is given to demonstrate the effectiveness of KDE compared to the histogram estimator. Section 4 presents the multivariate KDE and illustrates its properties with a numerical example.

2.2 The Histogram Density Estimator

The oldest and most widely known density estimator is the histogram or binning. Given an origin at x_0 and a constant (for convenience) bin width b, we can define the bins of the histogram to be the intervals $\left[x_0 + mb,\ x_0 + (m+1)b\right]$ for any integer m. Then the histogram density estimator is defined by

$$\hat{f}(x) = \frac{1}{Nb}\left(\text{Total No. of } X_i \text{ in same bin as } x\right), \tag{2.2}$$

where N is the total number of samples (X_i).

To resolve the shape of the unknown density function f, a histogram requires a large number of bins and a large data set is essential to reduce the uncertainty inside each individual bin. In Chapter 3 we will discuss about the optimal bin width calculation for histogram. Another drawback of the histogram is, even in one dimension, the choice of origin can have significant effects. Just by changing the origin it is possible to obscure some structure of the unknown density function. The aim of the nonparametric estimation, "*let the data speak for themselves*", does not seem to be fulfilled. Histograms also present several difficulties for multivariate density estimation. Also it follows from the definition that $\hat{f}$ is not a continuous function, rather it is a step function.

2.2.1 The Naive Estimator

The naïve estimator estimates [5] the following probability density function

$$f(x) = \lim_{h \to 0} \frac{1}{2h} P(x - h < X < x + h), \tag{2.3}$$

by

$$\hat{f}(x) = \frac{1}{2Nh} \{ \text{No. of } X_i \text{ in } (x - h, x + h) \}, \tag{2.4}$$

for any given h. To express the estimator in a concrete mathematical form, let us define the weight function w by

$$w(x) = \begin{cases} \frac{1}{2}, & \text{for } |x| < 1 \\ 0, & \text{otherwise} \end{cases}. \tag{2.5}$$

Then the naïve estimator can be written as

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h} w\left(\frac{x - X_i}{h} \right). \tag{2.6}$$

Monte Carlo sampling of the naïve estimator is efficient, because it is an equal weighted summation of N weight functions. However, we still need to define h, and the estimate $\hat{f}$ jumps at the points $X_i \pm h$. The importance of the naïve density estimator is that it leads to the definition of the kernel density estimator, which will be discussed in the next section, and which is the primary subject of this thesis. It follows from Eq. (2.5) and Eq. (2.6) that the estimate is constructed by placing a box of width $2h$ and height $(2Nh)^{-1}$ on each observation, and these are summed to obtain the overall estimate. If we replace the box $w(x)$ by a smooth function $k(x)$ which satisfies the condition

$$\int_{-\infty}^{\infty} k(x) dx = 1, \tag{2.7}$$

then the estimate will be a smooth function of x, known as the kernel density estimator.

2.3 Univariate Kernel Density Estimation Method

Consider N real observations $X_1, ..., X_N$ (independent and identically distributed) whose underlying density ($f(x)$) is to be estimated. The kernel density estimator with kernel k for univariate data is defined by

$$\hat{f}(x) = \frac{1}{Nh}\sum_{i=1}^{N} k\left(\frac{x - X_i}{h}\right), \tag{2.8}$$

where h ($h \to 0 \, as \, N \to \infty$) is the bandwidth, also called the smoothing parameter. $\hat{f}(x)$ is the estimator for the density function $f(x)$. The bandwidth calculation and the form of the kernel function k will be discussed later in this chapter. Generally the kernel k is a symmetric density function, for example the normal density, about zero and has the following properties:

$$\int k(t)dt = 1, \int tk(t)dt = 0, \; and \; \int t^2 k(t)dt = k_2 \neq 0 \tag{2.9}$$

Except otherwise stated, $\int$ will refer to an integral over the range $(-\infty, \infty)$. The kernel estimator can be visualized as the sum of bumps (kernels) placed at each observation. The kernel function k determines the shape of the bumps while the window width h determines their width. The Mean Integrated Square Error (*MISE*) is used as a measure of the discrepancy between the actual and the estimated density. The *MISE* is defined as follows

$$MISE\left(\hat{f}\right) = E\int\left[\hat{f}(x) - f(x)\right]^2 dx. \tag{2.10}$$

The term E is the expectation operator. Since the integral is non-negative, the order of the integration and expectation of the above equation can be reversed. Equation (2.10) can be written as

$$MISE\left(\hat{f}\right) = \int E\left[\hat{f}(x) - f(x)\right]^2 dx. \tag{2.11}$$

The integrand in Eq. (2.11) can be manipulated as

$$
\begin{aligned}
E\left\{\hat{f}(x)-f(x)\right\}^2 &= E\left\{\hat{f}(x)-E\left\{\hat{f}(x)\right\}+E\left\{\hat{f}(x)\right\}-f(x)\right\}^2 \\
&= E\left[\left\{E\hat{f}-f\right\}^2+\left\{\hat{f}-E\hat{f}\right\}^2+2\left\{E\hat{f}\,\hat{f}-\left\{E\hat{f}\right\}^2-f\hat{f}+fE\hat{f}\right\}\right] \\
&= \left\{E\hat{f}-f\right\}^2+E\left\{\hat{f}-E\hat{f}\right\}^2+2\left\{\left\{E\hat{f}\right\}^2-\left\{E\hat{f}\right\}^2-f\,E\hat{f}+f\,E\hat{f}\right\} \quad (2.12)\\
&= \left\{E\hat{f}(x)-f(x)\right\}^2+E\left\{\hat{f}(x)-E\hat{f}(x)\right\}^2 \\
&= bias^2+\operatorname{var}\hat{f}(x).
\end{aligned}
$$

This yields:

$$
MISE(\hat{f})=\int\left\{E\hat{f}(x)-f(x)\right\}^2dx+\int\operatorname{var}\hat{f}(x)dx. \quad (2.13)
$$

Therefore the *MISE* has two components, namely the integrated bias square and the integrated variance.

2.3.1 Bandwidth Calculation

For each x, $\hat{f}(x)$ can be considered as a random variable as it depends on the observations $X_1,.....,X_N$. The expected value and the variance of the estimator are given by

$$
\begin{aligned}
E\hat{f}(x) &= E\left\{\frac{1}{Nh}\sum_{i=1}^{N}k\left(\frac{x-X_i}{h}\right)\right\} \\
&= \frac{1}{N}\sum_{i=1}^{N}E\left\{\frac{1}{h}k\left(\frac{x-X_i}{h}\right)\right\}, \quad (2.14)\\
&= \int\frac{1}{h}k\left(\frac{x-y}{h}\right)f(y)dy
\end{aligned}
$$

and

$$
\operatorname{var}\hat{f}(x)=\frac{1}{N}\left\{\int\frac{1}{h^2}k\left(\frac{x-y}{h}\right)^2f(y)dy-\left\{\frac{1}{h}\int k\left(\frac{x-y}{h}\right)f(y)dy\right\}^2\right\}. \quad (2.15)
$$

The bias in the estimation of $f(x)$ does not depend directly on the sample size (N), but depends on the bandwidth h, which of course is a function of N. That means that the bias will depend indirectly on N through its dependence on h. We can write

$$
\begin{aligned}
bias &= E\hat{f}(x) - f(x) \\
&= \int \frac{1}{h} k\left(\frac{x-y}{h}\right) f(y)dy - f(x) \\
&= \int k(z)\{f(x-hz) - f(x)\}dz, \quad \text{by substituting } z = \frac{x-y}{h}.
\end{aligned} \tag{2.16}
$$

By applying a Taylor series expansion and using the assumptions made about k in Eq. (2.9), we can obtain from Eq. (2.16):

$$
\begin{aligned}
bias &= -hf'(x)\int zk(z)dz + \frac{1}{2}h^2 f''(x)\int z^2 k(z)dz + \text{Higher Order Terms in h} \\
&\approx \frac{1}{2}h^2 f''(x)\int z^2 k(z)dz \\
&= \frac{1}{2}h^2 f''(x)k_2.
\end{aligned} \tag{2.17}
$$

We assume that the unknown density function $f(x)$ has at least continuous derivatives of first and second order. In the similar way, as in the bias, we can obtain an equation for the variance:

$$
\begin{aligned}
\text{var}\,\hat{f}(x) &= \frac{1}{N}\left\{\int \frac{1}{h^2} k\left(\frac{x-y}{h}\right)^2 f(y)dy - \left\{\frac{1}{h}\int k\left(\frac{x-y}{h}\right) f(y)dy\right\}^2\right\} \\
&= \frac{1}{N}\int \frac{1}{h^2} k\left(\frac{x-y}{h}\right)^2 f(y)dy - \frac{1}{N}\{f(x) + bias(x)\}^2, \quad [\text{using Eq. 2.14}] \\
&\approx \frac{1}{Nh}\int f(x-hz)k(z)^2 dz - \frac{1}{N}\{f(x) + O(h^2)\}^2, \quad [\text{using } \frac{x-y}{h} = z, \text{and Eq. 2.17}] \\
&\approx \frac{1}{Nh}\int \{f(x) - hzf'(x) + ...\}k(z)^2 dz + O(\frac{1}{N}), \qquad [\text{By Taylor expansion}] \\
&\approx \frac{1}{Nh} f(x)\int k(z)^2 dz + O(\frac{1}{N}).
\end{aligned} \tag{2.18}
$$

The *MISE* can be approximated easily by combining Eq. (2.13), (2.17), and (2.18) to find:

$$
\begin{aligned}
MISE(\hat{f}) &\approx \int \left\{\frac{1}{2}h^2 f''(x)k_2\right\}^2 dx + \int \frac{1}{Nh} f(x)\int k(z)^2 dz\, dx \\
&= \frac{1}{4}h^4 k_2^2 \int f''(x)^2 dx + \frac{1}{Nh}\int k(z)^2 dz.
\end{aligned} \tag{2.19}
$$

The basic assumption here is that the true density f is such that its second derivative f'' is continuous and square integrable. Equation (2.19) illustrates the fundamental problems in probability density estimation techniques. If we want to reduce the bias, we have to employ a very small bandwidth h. However, for small bandwidth, the integrated variance will become large. On the other hand by choosing a large value of h, the random variation as measured by the variance can be reduces at the expense of introducing systematic bias into the estimation. It should be noted here that whatever method of density estimation is being used, the choice of smoothing parameter (bandwidth for KDE, bin width for histogram etc) implies a trade-off between random and systematic error. This will be discussed again in chapter 3 with a numerical example.

It is straightforward to derive the optimum bandwidth by minimizing *MISE* in Eq. (2.19):

$$\frac{\partial MISE}{\partial h} = h^3 k_2^2 \int f''(x)^2 \, dx - \frac{1}{Nh^2} \int k(z)^2 \, dz = 0$$

$$\Rightarrow h_{opt} = \frac{\left\{ \int k(z)^2 \, dz \right\}^{1/5}}{k_2^{2/5} \left\{ \int f''(x)^2 \, dx \right\}^{1/5}} N^{-1/5}. \tag{2.20}$$

The Eq. (2.20) for optimal bandwidth itself depends on the unknown density function $f(x)$ being estimated. However we can draw some useful conclusions from the optimum bandwidth formula. The optimum bandwidth will go to zero at a very slow rate with the increase the number of observations N. It can be also concluded that smaller value of h will be appropriate for rapidly varying densities, because the term $\int f''(x)^2 \, dx$, in a sense, is the measure of the rapidity of the fluctuations in the unknown density function. A natural approach for calculating the bandwidth *h* is to use a standard family of distributions, like the Gaussian distribution, to obtain a value of the term $\int f''(x)^2 \, dx$ in the Eq. (2.20). We can derive an expression for the bandwidth *h* by assuming a Gaussian function as the kernel function k [5]:

$$h_{opt} = 1.06 \sigma N^{-1/5}. \tag{2.21}$$

The quantity σ^2 is the variance of the normal distribution and can be estimated by the usual standard deviation of the data or by some more robust estimators, for example range or interquartile range [5]. The interquartile range is the difference between the first quartile and third quartile of a set of data. The bandwidth or smoothing parameter h is of crucial importance in density estimation. Several other techniques to calculate the bandwidth exist in the statistical literature [5, 7, 8, 9], for examples least-square cross-validation, maximum likelihood cross-validation, or the local adaptive method. The basic idea behind the adaptive method is to construct a kernel estimate consisting of bumps or kernel functions placed at each observation, but to allow the bandwidth of the kernel to vary from one point to another. This method is based on the intuition that a natural way to deal with long-tailed densities is to employ a wider kernel in regions of low density. But these methods are expensive with respect to computing time as one needs to obtain a rough estimation of the density function in order to compute locally adaptive bandwidths. These bandwidths are then used with the data points to calculate the final estimation of the unknown density function. Therefore, this is a two step process. For application to Monte Carlo radiation transport, an inexpensive and easily obtained bandwidth is crucial. That is why the bandwidth in Eq. (2.21), which is the global bandwidth, is used for all our one dimensional simulations. In Chapter 3 for our KDE estimated Monte Carlo tallies we use a quasi-locally adaptive bandwidth, which we call the region-based bandwidth. For the region-based bandwidths we calculated the σ and N (number of collisions or the number of points extracted from the tracks generated in a particular material region) for each region. Therefore, these region based bandwidths are one step fast calculations, but are not exactly locally adaptive.

2.3.2 Kernel Function

Inserting the value of optimum bandwidth h_{opt} from Eq. (2.20) into Eq. (2.19), we can arrive at the final minimum value expression for $MISE$:

$$\begin{aligned} MISE\left(\hat{f}\right)_{KDE} &= \frac{5}{4} C(k) \left\{ \int f''(x)^2 \, dx \right\}^{1/5} N^{-4/5} \\ &= O\left(N^{-4/5}\right), \quad [\text{in leading order term}] \end{aligned} \tag{2.22}$$

where the constant $C(k)$ is given by

$$C(k) = k_2^{2/5} \left\{ \int k(z)^2 \, dz \right\}^{4/5}. \tag{2.23}$$

The constant $C(k)$ depends on the kernel function k. So the kernel which minimizes $C(k)$ will minimize $MISE$, given that all other things are equal. In a different context, Hodges and Lehmann [12] in 1956 demonstrated that $C(k)$ can be minimized by using the function $k_e(x)$:

$$k_e(x) = \begin{cases} \frac{3}{4\sqrt{5}}\left(1 - \frac{x^2}{5}\right), & |x| \le \sqrt{5} \\ 0, & \text{otherwise} \end{cases}. \tag{2.24}$$

This function $k_e(x)$ was first used by Epanechnikov [13] in 1969 for kernel density estimation and is widely known as the Epanechnikov kernel. This kernel is illustrated in Figure 2.1.

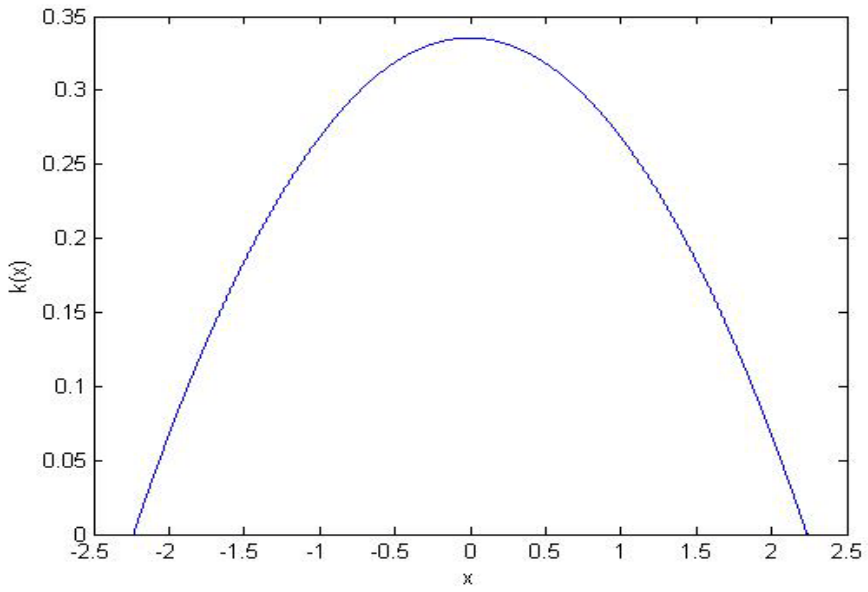

Figure 2.1 The Epanechnikov kernel.

We can define efficiency of any kernel function k by comparing it with the Epanechnikov kernel:

$$\begin{aligned} eff(k) &= \left\{C(k_e)/C(k)\right\}^{5/4} \\ &= \frac{3}{5\sqrt{5}}\left\{\int x^2 k(x)\,dx\right\}^{-1/2}\left\{\int k(x)^2\,dx\right\}^{-1}. \end{aligned} \tag{2.25}$$

Some kernels and their efficiencies are given in Table 2.1.

Table 2.1 Some kernels and their efficiencies [5]

Kernel	$k(x)$	Efficiency
Epanechnikov	$\begin{cases} \frac{3}{4\sqrt{5}}\left(1-\frac{x^2}{5}\right), & \lvert x\rvert \le \sqrt{5} \\ 0, & \text{otherwise} \end{cases}$	1
Biweight	$\begin{cases} \frac{15}{16}\left(1-x^2\right)^2, & \lvert x\rvert \le 1 \\ 0, & \text{otherwise} \end{cases}$	0.9939
Triangular	$\begin{cases} 1-\lvert x\rvert, & \lvert x\rvert \le 1 \\ 0, & \text{otherwise} \end{cases}$	0.9859
Gaussian	$\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$	0.9512
Rectangular	$\begin{cases} \frac{1}{2}, & \lvert x\rvert \le 1 \\ 0, & \text{otherwise} \end{cases}$	0.9295

Efficiency provides a measure to compare different symmetric kernels with the Epanechnikov kernel. Since $C(k_e)$ has the minimum value among all the symmetric kernels, for any other symmetric kernel k, the closer the efficiency to one, the smaller is the *MISE* for that k, given that all other conditions are same. Table 2.1 shows that there is very little difference between the various kernels on the basis of asymptotic *MISE* calculation. Therefore the choice of kernel type should be based on other considerations, such as computational expense and differentiability. For most of our Monte Carlo simulations we used the Epanechnikov kernel, because of its highest efficiency.

2.3.3 Boundary Correction

The performance of the density estimator $\hat{f}(x)$ is poor near boundaries due to the boundary effect that occurs in nonparametric curve estimation problems. This is due to the fact that the bias of $\hat{f}(x)$ is not in the order of $O(h^2)$, which is the bias for the interior points or for cases without boundaries given by Eq. (2.17), at points close to the boundaries. Whenever the data points are near the boundaries the associated kernel will not integrate to unity over the problem domains because it overlaps with the boundary. Simply truncating the estimator at the boundaries and then renormalizing $\hat{f}(x)$ to unity will not entirely solved the problem as noted by Silverman [5] and also tested by the author. There are several boundary correction methods available in the statistical literatures. Some well known methods are listed below.

1. The reflection method [5, 14]
2. The boundary kernel method [8, 14].
3. The transformation method [15].
4. The pseudo-data method [16]
5. The local linear method [17]

If there is one boundary at the origin then the reflection method is specially designed for the case $f'(0)=0$, where f' denotes the first derivative of f. In the context of Monte Carlo radiation transport calculation, the reflection method is useful as reflecting boundaries are common in nuclear engineering applications. The data point is reflected within the boundary region with respect to the boundary. The extent of the boundary region can be defined by the bandwidth which will be discussed later in this section. Assuming the boundary is at the origin, the reflected kernel density estimator for the boundary region is given by

$$\hat{f}(x)=\frac{1}{Nh}\sum_{i=1}^{N}\left\{k\left(\frac{x-X_i}{h}\right)+k\left(\frac{x+X_i}{h}\right)\right\}. \tag{2.26}$$

The boundary kernel method is more general in the sense that it can adapt any shape of the density. This method is very inexpensive and easy to implement. This method will be discussed in detail later in this section. We used this boundary correction (boundary

kernels) method for all our Monte Carlo simulation except for the fission source convergence problem, which is the subject matter of Chapter 5. The local linear method is a special case of the boundary kernel method where local polynomial fitting is used. The pseudo-data method of Cowling and Hall [16] generates some extra data points $X_{(i)}$'s by employing their "three-point-rule", and then combine them with the original data X_i's to form a kernel type estimator. The transformation method proposed by Marron and Ruppert, is another attractive method. But this method is computationally expensive and not so easy to implement. This method comprises of a three step process. First, a transformation g is selected from a parametric family so that the density of $Y_i = g(X_i)$ has a first derivative that is approximately equal to 0 at the boundaries of its support. In the second step, a kernel estimator with reflection is applied to the Y_i's. Finally, the estimator is transformed back to the original domain by the change of variable to obtain an estimate of f. We initially investigated and extended from one to two boundaries another boundary correction method developed by Karunamuni et al. [18]. This boundary method is known as generalized version of the reflection method. Their method is quite similar to the transformation method but the generalized reflection, more specifically the transformation g, also depends on the estimation of the derivatives of the unknown pdf at the boundaries. That means basically we are dealing with two estimations: the unknown density function f and its derivative at the boundaries. The method was sensitive to the derivative estimation. Due to the complexity of this method, especially in the point of view of Monte Carlo calculation, we decided not to use that. In the following section first we will discuss the boundary kernel method followed by a brief introduction of the generalized reflection method.

2.3.4 Boundary Correction by Boundary Kernel Method

To quantify the density estimation boundary problem mathematically let us assume that we have two boundaries at $x\min$ and $x\max$. Let k be a kernel with support confined to $[-1,1]$. Using Eq. (2.14), the expected value of the estimator $\hat{f}(x)$ is given by

$$E\left[\hat{f}(x)\right] = \int_{x\min}^{x\max} \frac{1}{h} k\left(\frac{x-y}{h}\right) f(y)dy \tag{2.27}$$

The quantity x is a representative point in the physical domain with boundaries $x\min$ and $x\max$. After a change of variable and a Taylor series expansion as described before the Eq. (2.27) becomes

$$E\left[\hat{f}(x)\right] = f(x)\int_{\frac{x-x\max}{h}}^{\frac{x-x\min}{h}} k(u)du - hf'(x)\int_{\frac{x-x\max}{h}}^{\frac{x-x\min}{h}} uk(u)du + \frac{h^2}{2} f''(x)\int_{\frac{x-x\max}{h}}^{\frac{x-x\min}{h}} u^2 k(u)du - \ldots\ldots. \tag{2.28}$$

If $x \geq x\min + h$ and $x \leq x\max - h$ then the upper and lower limit of the integrals in Eq. (2.28) can be replaced by 1 and -1, respectively, and then by using Eq. (2.9) we obtain a bias of order $O\left(h^2\right)$, which is the same as before. Note that, for any fixed $x > 0$ and h converging to zero we will always have $x > h$ for sufficiently large N. Hence for this case we can define the boundary region as the region within one bandwidth of a boundary. The observations that are not within one bandwidth from both boundaries are labeled interior observations/points. The basic idea is to use a point wise boundary kernel for the boundary regions which will force the first integral of the right hand side to Eq. (2.28) to be unity and the second one to be 0, so that we will achieve $O\left(h^2\right)$ bias for the boundary region. For the interior points we are using the usual kernel function, such as the Epanechnikov kernel. For the left boundary region ($x\min < x < x\min + h$) we can write from Eq. (2.28)

$$E\left[\hat{f}(x)\right] = f(x)\int_{-1}^{p} k(u)du - hf'(x)\int_{-1}^{p} uk(u)du + \frac{h^2}{2} f''(x)\int_{-1}^{p} u^2 k(u)du - \ldots\ldots, \tag{2.29}$$

where $p = \dfrac{x - x\min}{h}$. Now let us define

$$a_i\left(p\right) = \int_{-1}^{p} u^i k\left(u\right)du. \tag{2.30}$$

One simple boundary kernel can be defined [14] which will force a $O\left(h^2\right)$ bias for the boundary region by a linear multiple of the kernel k:

$$k_b(u) = \frac{\{a_2(p) - a_1(p)u\}k(u)}{a_0(p)a_2(p) - a_1(p)^2}, \text{ for } -1 < u < p \tag{2.31}$$

Similarly for the right hand boundary region defined by $x\max - h < x < x\max$ Eq. (2.28) becomes

$$E\left[\hat{f}(x)\right] = f(x)\int_p^1 k(u)du - hf'(x)\int_p^1 uk(u)du + \frac{h^2}{2}f''(x)\int_p^1 u^2k(u)du - \ldots\ldots, \tag{2.32}$$

where $p = \frac{x - x\max}{h}$. We can use the same boundary kernel for the right boundary region given by Eq. (2.31) by defining

$$a_i(p) = \int_p^1 u^i k(u)du. \tag{2.33}$$

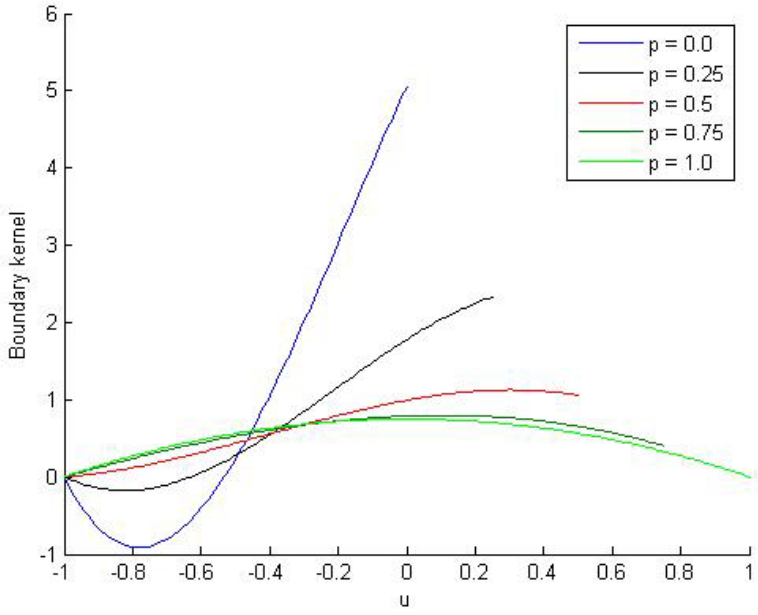

Figure 2.2 Boundary kernel $k_b(u)$ based on rescaled Epanechnikove kernel for different p values.

Figure 2.2 plots the boundary kernel $k_b(u)$ for several values of p, and k equal to the rescaled Epanechnikov kernel, which is given by

$$k(x)=\frac{3}{4}\left(1-x^2\right),\ \text{for}\left|x\right|\le 1 \tag{2.34}$$

A drawback of this boundary kernel method is that the estimates $\hat{f}$ might be negative near the endpoints, especially when $f\left(x\min\right)\approx 0$ and $f\left(x\max\right)\approx 0$. Some remedies have been proposed [14, 19] to correct this deficiency. For most of our Monte Carlo simulation we simply used this boundary kernel without any practical problem as this is very inexpensive, simple to implement and very effective boundary correction method available in statistical literature.

2.3.5 Generalized Reflection

This boundary correction method was initially used. This is not very easy to implement and computationally expensive as the derivatives of the unknown density function need to be estimated at the boundaries in order to estimate the unknown density function itself. In this section the generalized reflection method will be discussed briefly.

Karunamuni and Albert [18] introduced this generalize reflection method in 2005 by using a transformation that preserves an $O\left(h^2\right)$ boundary bias for a single boundary at $x=0$. We have extended this work to a slab of width a with two boundaries at $x=0$ and $x=a$. The resultant estimator is

$$\hat{f}(x)=\begin{cases}\dfrac{1}{Nh}\displaystyle\sum_{i=1}^{N}\left\{k\left(\frac{x-g_1(X_i)}{h}\right)+k\left(\frac{x+g_1(X_i)}{h}\right)\right\}, & 0\le x\le h\\ \dfrac{1}{Nh}\displaystyle\sum_{i=1}^{N}k\left(\frac{x-X_i}{h}\right), & h<x<a-h\\ \dfrac{1}{Nh}\displaystyle\sum_{i=1}^{N}\left\{k\left(\frac{x-g_2(X_i)}{h}\right)+k\left(\frac{x-2a+g_2(X_i)}{h}\right)\right\}, & a-h\le x\le a\end{cases} \tag{2.35}$$

where as usual h is the bandwidth, k is the kernel function with support $\left[-1,1\right]$, and g_1 and g_2 are the boundary transformations. The transformations are given by

$$g_1(y)=y+0.5d_0k_{c_0}^{/}y^2+0.5\left(d_0k_{c_0}^{/}\right)^2y^3, \tag{2.36}$$

where

$$d_0 = \frac{f^{(1)}(0)}{f(0)}, \tag{2.37}$$

$$k'_{c_0} = \frac{2\int_{c_0}^{1}(t-c_0)k(t)dt}{c_0 + 2\int_{c_0}^{1}(t-c_0)k(t)dt}, \tag{2.38}$$

and

$$g_2(y) = (2a-y) + 0.5d_a k'_{c_a}(a-y)^2 + 0.5\left(d_a k'_{c_a}\right)^2 (a-y)^3, \tag{2.39}$$

where

$$d_a = \frac{f^{(1)}(a)}{f(a)}, \tag{2.40}$$

$$k'_{c_a} = -\frac{2\int_{-1}^{c_a}(t-c_a)k(t)dt}{c_a + 2\int_{-1}^{c_a}(t-c_a)k(t)dt}. \tag{2.41}$$

c_o and c_a are given by

$$c_0 = \frac{x}{h}, \text{and } c_a = \frac{x-a}{h} \tag{2.42}$$

The logarithmic derivatives d_0 and d_a are estimated by FET [20], which will be introduced briefly in chapter 3. It was found that the estimation of these terms is important for obtaining good results near the boundaries. Instead of directly estimating the derivatives we estimate d_0 and d_a as

$$\hat{d} = \frac{\log \hat{f}(x+h_1) - \log \hat{f}(x)}{h_1}, \tag{2.43}$$

where $\hat{f}(x)$ and $\hat{f}(x+h)$ are estimated by FET [20] and h_1 is the bandwidth given by [18]

$$h_1 = hN^{-1/100} \tag{2.44}$$

2.3.6 Numerical 1-D KDE example

This is a simple example to demonstrate the ability of the KDE to reproduce a known pdf, in this case a clipped exponential on the interval $[-1,1]$, which is given by

$$f(x) = \frac{e^{-x}}{e - e^{-1}}, \quad x \in [-1,1]. \tag{2.45}$$

To depict this graphically, 10,000 observations are drawn from $f(x)$ and KDE is used with boundary kernel method as the boundary correction to evaluate $\hat{f}(x)$ at 50 equally spaced points between -1 and 1.These points are plotted in Figure 2.3.

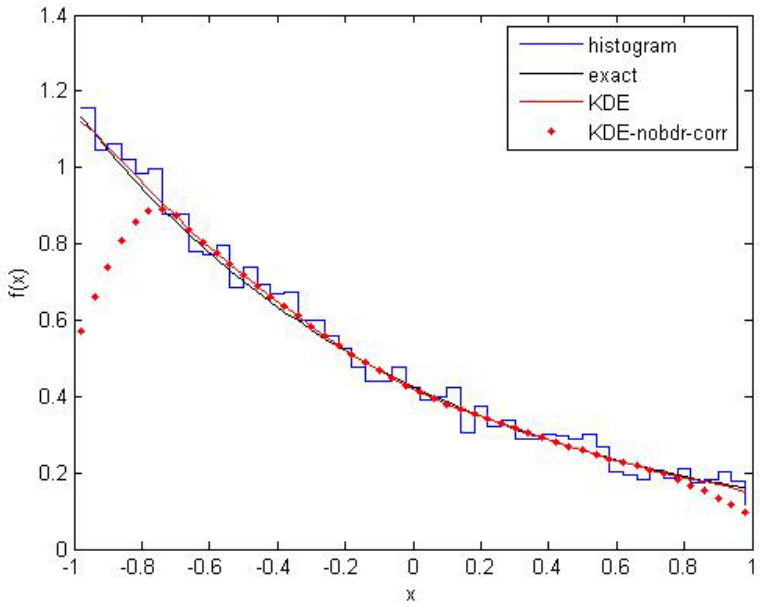

Figure 2.3 Comparison between KDE, KDE without boundary correction, and Histogram.

It is clear from Figure 2.3 that the boundary correction algorithm discussed earlier with boundary kernel method is performing well near the boundaries. A visual comparison suggests the kernel density estimator estimates the clipped exponential better than the histogram. This result will be discussed in a quantitative manner in Chapter 3. Note that, though the bandwidth is calculated assuming the underlying distribution is Gaussian, it seems to work well in estimating the exponential distribution.

Figure 2.3 also illustrates the potential for KDE to represent Monte Carlo radiation transport tallies, since the tally points can be treated as if they were un-normalized samples from a pdf. This is the basic motivation of kernel density estimated Monte Carlo collision and track-length tallies which is the topic of the next chapter.

2.4 Multivariate Kernel Density Estimator

Let $\left\{\left(X_i^1,.....X_i^d\right), i=1,...N\right\}$ be a sample of independent and identically distributed random variables with an unknown density function $f\left(x_1,.....,x_d\right)$. The general multivariate kernel density estimator is given by

$$\hat{f}\left(x_1,.....,x_d\right)=\frac{1}{Nh_1...h_d}\sum_{i=1}^{N}K\left(\frac{x_1-X_i^1}{h_1},.....,\frac{x_d-X_i^d}{h_d}\right), \tag{2.46}$$

where K denotes a multivariate kernel function and $\left[h_1,.....,h_d\right]$ the bandwidth vector. In practice the choice of K is especially difficult in the boundary region. Therefore, for Monte Carlo implementation we propose to use the product kernel estimator. The estimator is defined as

$$\hat{f}(x_1,...,x_d)=\frac{1}{N}\sum_{i=1}^{N}\prod_{l=1}^{d}\frac{1}{h_l}k\left(\frac{x_l-X_l^i}{h_l}\right), \tag{2.47}$$

where $h_1,.....,h_d$ are the bandwidth parameters and k is the usual univariate kernel function, for example the Epanechnikov kernel. In the same fashion as in the univariate case, approximate expressions can be derived for the bias and variance of the multivariate estimator, and these can be used to obtain some guidance for the appropriate choice of the bandwidth parameters. The expected value of the estimator is given by

$$E\left[\hat{f}\left(x_1,...,x_d\right)\right]=\int...\int\left[\prod_{l=1}^{d}\frac{1}{h_l}k_l\left(\frac{x_l-y_l}{h_l}\right)\right]f\left(y_1,...,y_d\right)dy_1...dy_d. \tag{2.48}$$

By changing variables in Eq. (2.48) we can obtain

$$E\left[\hat{f}\left(x_1,...,x_d\right)\right]=\int...\int\left[\prod_{l=1}^{d}k_l\left(z_l\right)\right]f\left(x_1-h_1z_1,...,x_d-h_dz_d\right)dz_1...dz_d \tag{2.49}$$

By expanding $f\left(x_1 - h_1 z_1, ..., x_d - h_d z_d\right)$ in a multivariate Taylor series with respect to all x_i about the point $\left(x_1, x_2, ..., x_d\right)$ we obtain

$$E\left[\hat{f}\left(x_1, ..., x_d\right)\right] = \int ... \int \left[\prod_{l=1}^{d} k_l\left(z_l\right)\right] \left\{ f\left(x_1, ..., x_d\right) - \sum_{j=1}^{d} h_j z_j \frac{\partial f}{\partial x_j} + \frac{1}{2}\sum_{j=1}^{d} h_j^2 z_j^2 \frac{\partial^2 f}{\partial x_j^2} + ... \right\} dz_1 ... dz_d . \tag{2.50}$$

Considering the same kernel function k in all directions and by using Eq. (2.9), the following is obtained:

$$E\left[\hat{f}\left(x_1, ..., x_d\right)\right] \approx f\left(x_1, ..., x_d\right) + \frac{1}{2} k_2 \sum_{j=1}^{d} \frac{\partial^2 f\left(x_1, ..., x_d\right)}{\partial x_j^2} h_j^2 . \tag{2.51}$$

In Eq. (2.51) the higher order terms in h are ignored. For notational simplicity, assume equal bandwidth in all directions, yielding the final expression of the bias of the multivariate kernel density estimator:

$$bias\left(x_1, ..., x_d\right) = E\left[\hat{f}\left(x_1, ..., x_d\right)\right] - f\left(x_1, ..., x_d\right) \approx \frac{1}{2} k_2 h^2 \sum_{j=1}^{d} \frac{\partial^2 f\left(x_1, ..., x_d\right)}{\partial x_j^2} . \tag{2.52}$$

The variance of $\hat{f}\left(x_1, ..., x_d\right)$ is given by

$$\begin{aligned} \operatorname{var} \hat{f}\left(x_1, ..., x_d\right) = & \frac{1}{N} \int ... \int \left[\prod_{l=1}^{d} \frac{1}{h_l} k_l\left(\frac{x_l - y_l}{h_l}\right)\right]^2 f\left(y_1, ..., y_d\right) dy_1 ... dy_d \\ & - \frac{1}{N}\left\{E\left[\hat{f}\left(x_1, ..., x_d\right)\right]\right\}^2 . \end{aligned} \tag{2.53}$$

By changing variable in Eq. (2.53) and ignoring $O\left(1/N\right)$ terms, this becomes

$$\operatorname{var} \hat{f}\left(x_1, ..., x_d\right) = \frac{1}{N}\left[\prod_{l=1}^{d} \frac{1}{h_l}\right] \int ... \int \left[\prod_{l=1}^{d} k_l\left(z_l\right)\right]^2 f\left(x_1 - h_1 z_1, ..., x_d - h_d z_d\right) dz_1 ... dz_d . \tag{2.54}$$

Considering same bandwidth h in all direction as before, and by applying multivariate Taylor series expansion to Eq. (2.54), the final expression for variance is obtained:

$$\operatorname{var} \hat{f}\left(x_1, ..., x_d\right) = \frac{1}{N h^d} f\left(x_1, ..., x_d\right) \beta, \tag{2.55}$$

where

$$\beta = \int ... \int \left[\prod_{l=1}^{d} k_l(z_l) \right]^2 dz_1 ... dz_d . \tag{2.56}$$

Similar to the univariate KDE, Combining Eqs. (2.52), and (2.55) gives the approximate mean integrated square error

$$MISE \approx \frac{1}{4} h^4 k_2^2 \int ... \int \left\{ \sum_{j=1}^{d} \frac{\partial^2 f(x_1, ..., x_d)}{\partial x_j^2} \right\}^2 dx_1 ... dx_d + \frac{1}{Nh^d} \beta. \tag{2.57}$$

Hence the approximately optimal bandwidth for the multivariate KDE is obtained by minimizing the $MISE$ by simple calculus:

$$h_{opt}^{d+4} = d \beta k_2^{-2} \left\{ \int ... \int \left(\sum_{j=1}^{d} \frac{\partial^2 f(x_1, ..., x_2)}{\partial x_j^2} \right)^2 dx_1 ... dx_d \right\}^{-1} N^{-1} \tag{2.58}$$

The optimal bandwidth can be chosen by replacing f in Eq. (2.58) by a standard density function, such as multivariate normal density.Replacing f by multivariate normal density and assuming a normal density function as the kernel k, the bandwidth is given by

$$h_i = \left\{ \frac{4}{(d+2)N} \right\}^{1/d+4} \sigma_i , \tag{2.59}$$

where h_i denotes the optimal smoothing parameter and σ_i the standard deviation in dimension i. For practical implementation, the latter is replaced by the sample estimate. This bandwidth formula is used for all our multidimensional Monte Carlo simulations. It is straightforward to use the boundary kernel method as the boundary correction for the multivariate KDE as we are taking the product of the kernel in each dimension. Therefore, if the point of evaluation at one dimension is within the boundary region, defined by the bandwidth in that dimension, then we just need to replace the kernel in that dimension by the boundary kernel given by Eq. (2.31).

2.4.1 Numerical Multivariate KDE Example

A simple 2 dimensional example is used to demonstrate that the multivariate KDE can reproduce a known density function. The density function used in this example is given by

$$f(x,y)=\frac{1+xy}{a^2+a^2/4},\ 0<x,y<a. \tag{2.60}$$

To plot this graphically, 2 million data points were drawn from $f(x,y)$ and Eq. (2.47) was used to evaluate $\hat{f}(x,y)$ at 20×20 equally spaced points inside a 2×2 square. These points are plotted in Figure 2.5. Figure 2.4 illustrates the exact reference plot obtained by evaluating the Eq. (2.60). The boundary kernel method is used as the boundary correction in this example. Visual inspection suggests that in this case 2-D KDE produces a good estimation of the known pdf f. Figure 2.5 reconfirms the potential for KDE to estimate Monte Carlo radiation transport tallies.

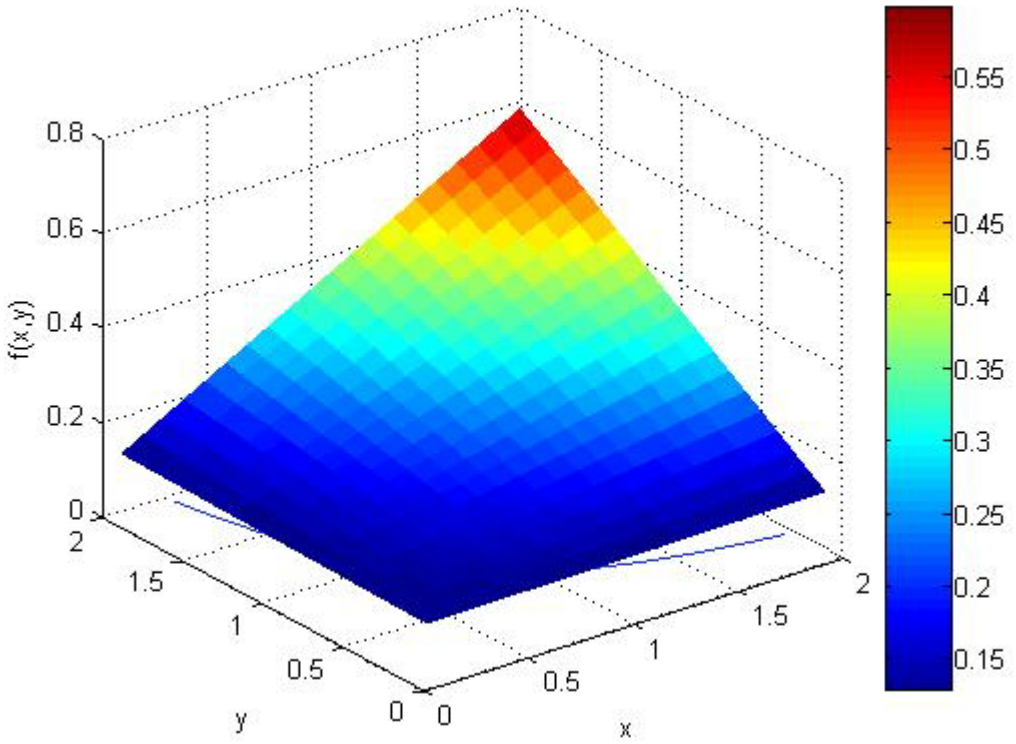

Figure 2.4 Exact graphical representation of the 2-d pdf $f(x,y)$ in Eq. (2.60).

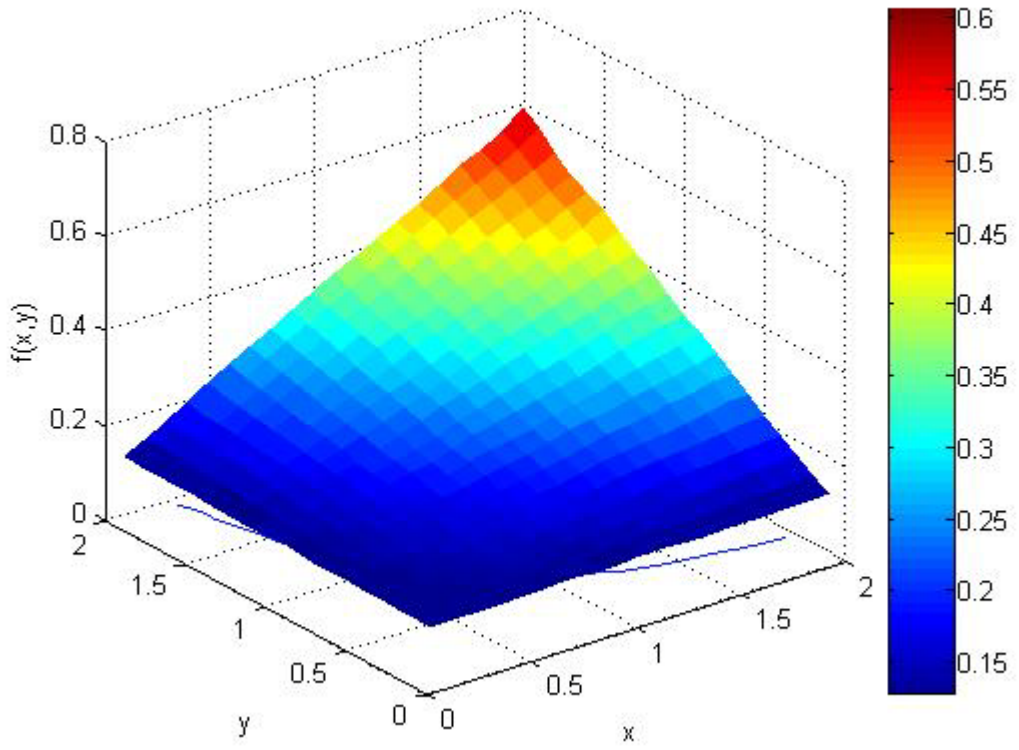

Figure 2.5 Graphical representation of the KDE estimation of function f in Eq. (2.60).

2.5 References

[1] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, Fourth ed, Tata McGraw-Hill, New Delhi (2002).

[2] E. Fix and J. L. Hodges, "Discriminatory Analysis, Nonparametric Estimation: Consistency Properties," *Report No. 4, Project No. 21-49-004*, USAF School of Aviation Medicine, Randolph Field, Texas (1951).

[3] M. Rosenblatt, "Remarks on Some Nonparametric Estimates of a Density Function," *Ann. Math. Statist.*, **27**, 832-837 (1956).

[4] E. Parzen, "On Estimation of a Probability Density Function and Mode," *Ann. Math. Statist.*, **33**, 1065-1076 (1962).

[5] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London (1986).

[6] L. Devroye and L. Györfi, *Nonparametric Density Estimation – The L_1 View*, John Wiley & Sons, New York (1985).

[7] W. Härdle, *Smoothing Techniques with Implementation in S*, Springer – Verlag, New York (1991).

[8] M. P. Wand and M. C. Jones, *Kernel Smoothing*, Chapman & Hall, London, UK (1995).

[9] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis – The Kernel Approach with S-Plus Illustrations*, Clarendon Press, Oxford (1997).

[10] J. Du and W. R. Martin, "Adaptive Kernel Density Estimation and Monte Carlo Sampling," *Trans. Am. Nucl. Soc.*, **74**, 171-171 (1996).

[11] N. Tyagi, W. R. Martin, J. Du, A. F. Bielajew, and I. J. Chetty, "A Proposed Alternative to Phase-space Recycling Using the Adaptive Kernel Density Estimator Method," *Medical Physics*, **33**, 553-560 (2006).

[12] J. L. Hodges and E. L. Lehmann, "The Efficiency of Some Nonparametric Competitors of the t-test," *Ann. Math. Statist.*, **27**, 324-335 (1956).

[13] V. A. Epanechnikov, "Nonparametric Estimation of a Multidimensional Probability Density," *Theor. Probab. Appl.*, **14**, 153-158 (1969).

[14] M. C. Jones, "Simple Boundary Correction for Kernel Density Estimation," *Statistics and Computing,* **3**, pp. 135-146 (1993).

[15] J. S. Marron and D. Ruppert, "Transformations to Reduce Boundary Bias in Kernel Density Estimation," *J. R. Statist. Soc. B*, **56**, 653-671 (1994).

[16] A. Cowling and P. Hall, "On Pseudodata Methods for Removing Boundary Effects in Kernel Density Estimation," *J. R. Statist. Soc. B*, **58**, 551-563 (1996).

[17] M. Y. Cheng, J. Fan, and J. S. Marron, "On Automatic Boundary Corrections," *The Annals of Statistics*, **25**, 1691-1708 (1997).

[18] R. J. Karunamuni and T. Alberts, "On Boundary Correction in Kernel Density Estimation," *Statistical Methodology*, **2**, 191-212 (2005).

[19] M. C. Jones and P. J. Foster, "A Simple Nonnegative Boundary Correction Method for Kernel Density Estimation." *Statistical Sinica*, **6**, 1005-1013 (1996).

[20] D. P. Griesheimer, *Functional Expansion Tallies for Monte Carlo Simulations*, Ph.D. Thesis, Nuclear Engineering and Radiological Sciences, University of Michigan (2005).

CHAPTER 3

Kernel Density Estimated Monte Carlo Tallies

3.1 Introduction

While traditional Monte Carlo tallies work very well for estimating integral quantities, such as average scalar flux over large volumes, they are limited in their abilities to estimate the detailed shape of the distribution of particle flux or current with respect to space, angle, energy or time. There are, however, many applications in which the shape of the flux distributions is desired. The conventional Monte Carlo approach to extract the shape information from the particle histories is to approximate the true distribution by a histogram. These histogram approximations are generated by dividing the phase space into bins and the Monte Carlo simulation yields an estimate of the desired quantity, e.g., the integrated flux, within each bin. When normalized by the bin width, the results generate a step approximation to the true distribution. To achieve better resolution, finer bins are required, resulting in poor statistics for the bins, and potentially leading to an excessive number of histories to reduce the uncertainty in each bin to an acceptable level. Also, finer partitioning of the phase space requires additional surfaces to be tracked in the geometry, which can add significant overhead to the simulation.

Keeping in mind that any positive tally can be viewed as an unnormalized probability density function (pdf), one approach to this problem is to estimate the equivalent pdf with the functional expansion tally (FET) method, which makes use of orthogonal functions (e.g., Legendre polynomials), which was done by Griesheimer [1]. This chapter discusses the application of KDE to estimate the underlying pdf of the Monte Carlo tally. Since KDE is a nonparametric density estimator, there are no a priori assumptions about the functional form of the distribution associated with the random variable. The KDE method is attractive because it can estimate MC tallies in any location within the required domain without any particular bin structure. Post-processing or simultaneous processing of the

KDE tallies is sufficient to extract detailed tally information for an arbitrary grid. In this chapter we introduce the mathematical form of two new scalar flux estimators, viz. KDE-collision and KDE-track-length estimators [2, 3], including their variances. Convergence properties of the KDE tally, the Functional Expansion Tally (FET), and the histogram tally are also investigated. Convergence analyses of these estimators has been reported previously [4, 5, and 6] and a detailed analytical and numerical convergence comparison between the FET and histogram tallies was carried out by Griesheimer et al. [7]. In this chapter the convergence properties of KDE from Chapter 2 are briefly reviewed, mainly for comparison with FET and histogram tallies. This chapter ends with quantitative and qualitative comparison between the KDE, FET, and histogram estimators.

3.2 Conventional Monte Carlo Tallies

In this chapter and throughout the thesis, individual particle histories are identified by the index i. All the events, such as collision, absorption, boundary crossing etc., in a particle's life are identified sequentially by the index c. Values of c can range from 0, the birth of the particle, to C_i, termination of the particle history after C_i events. For all radiation transport problems, the phase space location of particle i during an event c is expressed by the seven-dimensional vector $\vec{\xi}_{i,c} = \{x, y, z, \theta, \psi, E, t\}_{i,c}$. The series of events that occur during a single particle's life is a Markov chain [8], denoted

$$\hat{\xi}_i = \left\{ \vec{\xi}_{i,C_i}, \vec{\xi}_{i,C_i-1}, ..., \vec{\xi}_0 \right\}. \tag{3.1}$$

Every Monte Carlo simulated particle history yields a Markov chain $\hat{\xi}$. Each chain is independent and identically distributed by the parent distribution $f(\xi)$, which describes the relative likelihood of all possible Markov chains [9].

Monte Carlo tallies are functions that operate on a finite set of Markov chain realizations $\left\{ \hat{\xi}_i \right\}_{i=1,N}$, and yield an estimate for some desired unknown parameter μ of the governing distribution $f(\xi)$. The general form of all Monte Carlo tallies can be given by

$$\hat{\mu} = S_\mu \left(\left\{ \hat{\xi}_i \right\}_{i=1,N} \right), \tag{3.2}$$

where $\hat{\mu}$ is the estimator of μ, estimated by the set of particle histories $\left\{\hat{\xi}_i\right\}_{i=1,N}$, and S_μ in Monte Carlo terminology is known as the "tally" or "scoring" function. The scoring function is responsible for extracting desired information from the collection of particle histories. The distribution $f(\xi)$ theoretically describes all particle behaviors that are physically possible in the system. As it contains all information, it is possible to express any quantity of interest as a parameter of $f(\xi)$.

3.2.1 Conventional Monte Carlo Collision Estimator of Flux

In nuclear engineering, the incremental scalar flux $\phi(\underline{x},E,t)dxdE$ is defined as the total distance traveled per unit time by all the particles of energies dE about E in a volume element dx about $\underline{x}$ at time t. The energy dependent flux has the units $\left[cm^{-2}\cdot eV\cdot \sec^{-1}\right]$ and is mathematically expressed as

$$\phi(\underline{x},E,t)=vN(\underline{x},E,t), \tag{3.3}$$

where $N(\underline{x},E,t)$ is the corresponding number density of particles per unit volume at $\underline{x}$ and per unit energy at E at time t. Also, v is the velocity of a particle with energy E. Particle flux is an important quantity in nuclear engineering as it is used to calculate the nuclear reaction rates and leakage rates. A reaction rate $R_\chi(\underline{x},E,t)dxdE$ calculates the expected rate $\left[\text{reactions}\cdot \sec^{-1}\right]$ at which reaction of type χ take place in the elemental volume dx about $\underline{x}$, at time t, due to particles with incident energy dE about E. The reaction rate can be given by

$$R_\chi(\underline{x},E,t)=\Sigma_\chi(\underline{x},E,t)\phi(\underline{x},E,t), \tag{3.4}$$

where Σ_χ is the macroscopic cross section for reaction χ. By integrating Eq. (3.4) over all space, energy and time we can obtain

$$N_\chi=\int_V\int_E\int_t\Sigma_\chi(\underline{x},E,t)\phi(\underline{x},E,t)dtdEdx, \tag{3.5}$$

which is the total number of reactions that occur in the volume V per particle. In Eq. (3.5) the flux $\phi(\underline{x},E,t)$ is normalized by the total number of particle histories, which is a standard practice in nuclear engineering applications. Similarly, N_χ is also normalized.

It is trivial to calculate the average number of reactions in a finite volume by a simple Monte Carlo counting experiment. It is straightforward to show that for N independent particle histories, the quantity

$$\hat{N}_\chi = \frac{\text{Total number of reaction } \chi \text{ in tally volume}}{N} \tag{3.6}$$

is an unbiased estimator of the quantity N_χ. This estimator of the reaction rate can be used to derive an estimator for the average flux in the entire tally volume.

By using the mean value theorem [10], the average flux in an arbitrary special volume V can be expressed as

$$\bar{\phi} = \frac{1}{V}\int_V \phi(\underline{x})dx. \tag{3.7}$$

Now assume that the total macroscopic cross section is constant for all locations, times, and energies inside volume V. After multiplying and dividing the right hand side of Eq. (3.7) by Σ_t, this becomes

$$\bar{\phi} = \frac{1}{V\Sigma_t}\int_V \Sigma_t\phi(\underline{x})\,dx, \tag{3.8}$$

where Σ_t is the total macroscopic cross section. Note that the integral term in Eq. (3.8) is identical to Eq. (3.5). Substituting Eq. (3.5) into Eq. (3.8), and replacing the index χ by t, yields the following expression for the average flux:

$$\bar{\phi} = \frac{N_t}{V\Sigma_t}. \tag{3.9}$$

Using the unbiased estimator of N_t we can write

$$\hat{\phi} = \frac{\text{Total number of reactions in volume } V}{V\Sigma_t N}. \tag{3.10}$$

Equation (3.10) can be written in a more conventional Monte Carlo tally format

$$\hat{\phi} = \frac{1}{NV\Sigma_t} \sum_{i=1}^{N} \sum_{c=1}^{C_i} w_{i,c}, \tag{3.11}$$

where c is a sequential index of collision events for history i and $w_{i,c}$ is the weight of particle i prior to collision c. The weights are important for cases where variance reduction techniques are used [11]. For analog Monte Carlo simulations the weight w is always equal to 1. The average flux estimator in Eq. (3.11) is known as collision estimator as it scores at each collision a particle undergoes during the simulation. The total cross section in Eq. (3.11) can be easily replaced with any other cross section to yield other types of flux tallies, such as an absorption tally. Absorption estimators are not commonly used because they accumulate little information (1 event per history) relative to the collision estimator, and therefore have a large variance.

One very common and widely used collision estimator, especially in shielding applications, is the next event estimator or point detector estimator. These tallies are used to estimate the flux at a single point inside the system. At every collision this estimator scores what would be the contribution of that particular collision towards the detector point. This tally is the subject of the next chapter.

The underlying assumption of the above derivation of the collision flux estimator is the cross section in the tally volume is independent of time, energy, and space. Such an assumption is not physically realistic. It can be shown [9], not in a straightforward way, that if the cross section indeed depends on the phase space vector $\vec{\xi}$, then the collision flux estimator is given by

$$\hat{\phi} = \frac{1}{NV} \sum_{i=1}^{N} \sum_{c=1}^{C_i} \frac{w_{i,c}}{\Sigma_t\left(\vec{\xi}_{i,c}\right)}. \tag{3.12}$$

3.2.2 Monte Carlo Track Length Estimator of Flux

The collision flux estimator directly depends on the number of collisions inside the tally region. If the number of collision is not well resolved, for instance regions with very small cross sections, then the collision estimator can produce unreliable estimates with

high variance. Moreover collision estimates are incapable of estimating flux in a void region. These limitations prompted the development of a completely different flux tally: the track length or path length estimator.

A simple derivation of the track length tally can be achieved from the definition of the scalar flux given by Eq. (3.3). The speed v corresponds to the distance traveled by a single particle per unit time. The $N(\underline{x}, E, t)$ term is the particle density, which gives the total number of particles per unit phase space volume (e.g., per unit volume per unit energy) at time *t*. Hence the scalar flux can be interpreted as the total path length traversed per unit time per unit phase space volume by all the particles. This interpretation of scalar flux can be written mathematically as

$$\phi(\vec{\xi})\,d\xi = \text{Rate at which particles generate path length in } d\xi \text{ about } \vec{\xi}. \tag{3.13}$$

We can integrate Eq. (3.13) over some arbitrary phase space volume $\Re$ to obtain

$$\int_{\Re} \phi(\vec{\xi})\,d\xi = \text{Total path length generated by particles in } \Re. \tag{3.14}$$

By dividing both side of Eq. (3.14) by the volume V of $\Re$ and applying the mean value theorem we can write

$$\bar{\phi} = \frac{1}{V}\int_{\Re} \phi(\vec{\xi})\,d\xi = \frac{\text{Total path length generated by particles in } \Re}{V}. \tag{3.15}$$

The right hand side of Eq. (3.15) can be easily estimated in Monte Carlo simulation by simply accumulating the total distance traveled by the particles in the tally region multiplied by the weight of the particle and dividing by the volume of the tally region. The formal mathematical form of the track length tally is given by

$$\hat{\phi} = \frac{1}{NV}\sum_{i=1}^{N}\sum_{c=1}^{C_i} w_{i,c} d_{i,c,\Re} = \left\{\begin{array}{c}\text{Total path length generated by particles in } \Re \\ \text{per particle history}\end{array}\right\}, \tag{3.16}$$

where $d_{i,c,\Re}$ is the distance traveled by particle i in phase volume $\Re$ as it passes between events $c-1$ and c. This derivation of the path length tally is directly based on the definition of the scalar flux.

3.3 Weighted Kernel Density Estimator

In the previous definition of the kernel density estimator, which is discussed in chapter 2 (Eq. (2.8)), all the observations $(X_1,...,X_N)$ from the unknown density function $f(x)$ are considered equiprobable. That means they all have weight equal 1. If instead, all the observations from the unknown density function $(X_1,...,X_N)$ have corresponding weights $(w_1,...,w_N)$, then one can define the weighted kernel density estimator as

$$\hat{f}(x)=\frac{1}{\sum_{i=1}^{N} w_i}\sum_{i=1}^{N} w_i \frac{1}{h}k\left(\frac{x-X_i}{h}\right), \tag{3.17}$$

where the terms have their usual meaning. This weighted kernel density estimator is used to derive KDE collision and track length tallies. Similarly for three dimensional cases with observations $\left((X_1,Y_1,Z_1),....,(X_N,Y_N,Z_N)\right)$ from the unknown density function $f(x,y,z)$, and their corresponding weights $(w_1,...,w_N)$, the multivariate KDE given in Eq. (2.47) becomes

$$\hat{f}(x,y,z)=\frac{1}{\sum_{i=1}^{N} w_i}\sum_{i=1}^{N} w_i \frac{1}{h_x}k\left(\frac{x-X_i}{h_x}\right)\frac{1}{h_y}k\left(\frac{y-Y_i}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_i}{h_z}\right). \tag{3.18}$$

Equation (3.18) represents weighted 3-D KDE.

3.3.1 1-D Kernel Density Estimated Collision Tally and Its Variance

Deriving the KDE collision tally is a very straightforward exercise. Let $X_{i,c}$ is the location of a collision event during the life of a particle i, denoted by the index c. The collision point $X_{i,c}$ is given a weight equal to $w_{i,c}/\Sigma_t(X_{i,c})$, where $w_{i,c}$ is the weight of the particle i just before the collision event c and $\Sigma_t(X_{i,c})$ is the total cross section at the collision location $X_{i,c}$. By recording all the collision points and their corresponding weight during the Monte Carlo simulation and by using the definition of the weighted

kernel density estimator from Eq. (3.17), the following expression for the normalized flux distribution in the tally region is obtained:

$$\frac{\phi(x)}{\int \phi(x)dx} = \frac{1}{\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c})}} \sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c})}\frac{1}{h}k\left(\frac{x-X_{i,c}}{h}\right). \tag{3.19}$$

The lefthand side of this equation is the scalar flux "pdf", reflecting the fact that the scalar flux tally can be viewed as an unnormalized pdf. The bandwidth h can be calculated from Eq. (2.21) by using all the recorded collision points. By applying the conventional Monte Carlo collision estimator from Eq. (3.12), we can easily obtain an expression for the integrated flux over the tally region (the normalizing constant in the left hand side of Eq. (3.19)):

$$\int \phi(x)dx = \hat{\phi}V = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c})}. \tag{3.20}$$

By combining Eq. (3.19) and Eq. (3.20) one can obtain the final equation for the KDE collision flux tally:

$$\phi(x) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c})}\frac{1}{h}k\left(\frac{x-X_{i,c}}{h}\right). \tag{3.21}$$

The key observation in Eq. (3.21) is that it is independent of the arbitrary volume *R*. In other words, the KDE collision tally is *mesh-free* and provides a pointwise estimate of the scalar flux (and its variance as shown below) throughout the problem geometry.

Let $\hat{x}$ is the estimator of the true mean μ of some unknown density function $p(x)$. Considering $\left(\underline{x} = \{X_1, ..., X_N\}\right)$ are independent and identically distributed observations from the density function $p(x)$ we can write

$$\hat{x} = \frac{1}{N}\sum_{i=1}^{N} X_i. \tag{3.22}$$

By applying elementary statistics the variance of the estimated mean is given by

$$\begin{aligned}\sigma_{\hat{x}}^2 &= \frac{1}{N^2}\sum_{i=1}^{N}\sigma_{\underline{x}}^2 \\ &= \frac{1}{N}\sigma_{\underline{x}}^2.\end{aligned} \tag{3.23}$$

Hence the sample variance $\sigma_{\hat{x}}^2$ can be estimated by

$$\begin{aligned}\hat{\sigma}_{\hat{x}}^2 &= \frac{1}{N(N-1)}\sum_{i=1}^{N}\left(X_i-\hat{x}\right)^2 \\ &= \frac{\frac{1}{N}\sum_{i=1}^{N}X_i^2-\hat{x}^2}{N-1} \\ &= \frac{\frac{1}{N}\sum_{i=1}^{N}X_i^2-\left(\frac{1}{N}\sum_{i=1}^{N}X_i\right)^2}{N-1}.\end{aligned} \tag{3.24}$$

By using Eq. (3.24), we can easily derive an estimator of the variance of the KDE collision flux tally given by Eq. (3.21):

$$\sigma_{\phi(x)}^2 = \frac{\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{c=1}^{C_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c})}\frac{1}{h}k\left(\frac{x-X_{i,c}}{h}\right)\right\}^2-\left\{\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c})}\frac{1}{h}k\left(\frac{x-X_{i,c}}{h}\right)\right\}^2}{N-1}. \tag{3.25}$$

3.3.2 1-D Kernel Density Estimated Track Length Tally and Its Variance

Most modern Monte Carlo codes depend on track length estimators for tallying volumetric flux. This is because for most of the cases the track length estimator will give a lower variance estimate of the scalar flux than the collision estimator.

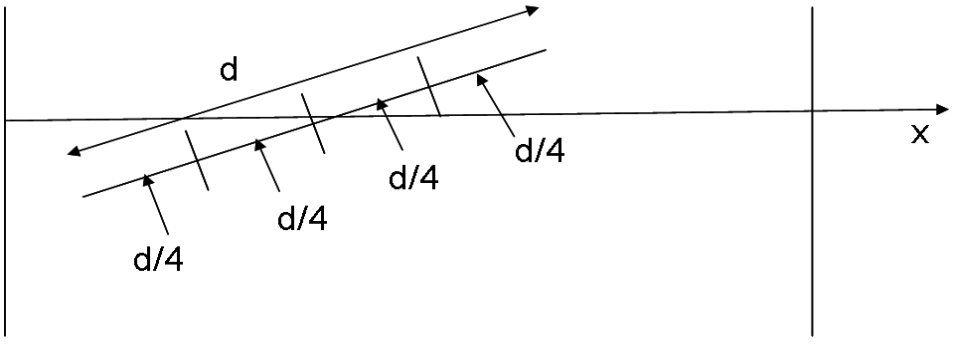

Figure 3.1 Generating sub-tracks from a single track.

Tracks are continuous between two events $c-1$ and c, where the events are collisions or surface crossings. However, for KDE tallying, scores need to be assigned to a point. Since tracks can be long, assigning the entire track to a single point is unrealistic, and it is also not clear where this point should be assigned. Therefore, it was decided to divide each single track into n uniform sub-tracks, where n is a user-defined quantity, and randomly select points from each sub-track and give that point a weight equal to the length of the sub-track. Figure 3.1 illustrates the process, where a track of length d is divided into 4 uniform sub-tracks. One point is extracted randomly from each sub-track and a weight $d/4$ is assigned to that point. If the weight of the particle tracing a track between events $c-1$ and c is $w_{i,c}$, then each extracted point from that track is given a weight $d_{i,c}w_{i.c}/n$, where $d_{i,c}$ is the length of the track between event $c-1$ and c. Proceeding in a similar fashion with the derivation of the KDE collision tally in Section 3.3.1, the KDE track length tally and its variance are obtained:

$$\phi(x)=\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C_i}\frac{w_{i,c}d_{i,c}}{n}\sum_{j=1}^{n}\frac{1}{h}k\left(\frac{x-X_{i,c,j}}{h}\right), \tag{3.26}$$

$$\sigma^2_{\phi(x)}=\frac{\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{c=1}^{C_i}\frac{w_{i,c}d_{i,c}}{n}\sum_{j=1}^{n}\frac{1}{h}k\left(\frac{x-X_{i,c,j}}{h}\right)\right\}^2-\{\Phi(x)\}^2}{N-1}, \tag{3.27}$$

where $X_{i,c,j}$ is the extracted point from the sub-track j of track $d_{i,c}$, between events $c-1$ and c for the i^{th} particle history.

Like the KDE collision estimator, the KDE tracklength estimator is also mesh-free, independent of the material or tally surfaces in the problem geometry.

3.3.3 Multi-D KDE Collision and Track Length Estimators and Their Variances

The 1-D KDE collision and track length tallies can be easily extended to three dimensions by using the 3-D weighted KDE given by Eq. (3.18). The 3-D KDE collision tally and its variance are given by

$$\phi(x,y,z)=\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c},Y_{i,c},Z_{i,c})}\frac{1}{h_x}k\left(\frac{x-X_{i,c}}{h_x}\right)\frac{1}{h_y}k\left(\frac{y-Y_{i,c}}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_{i,c}}{h_z}\right), \tag{3.28}$$

$$\sigma^2_{\phi(x,y,z)}=\frac{1}{(N-1)}\left[\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{c=1}^{c_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c},Y_{i,c},Z_{i,c})}\frac{1}{h_x}k\left(\frac{x-X_{i,c}}{h_x}\right)\frac{1}{h_y}k\left(\frac{y-Y_{i,c}}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_{i,c}}{h_z}\right)\right\}^2-\phi(x,y,z)^2\right]. \tag{3.29}$$

$\left(X_{i,c},Y_{i,c},Z_{i,c}\right)$ is the location of the c^{th} collision of the i^{th} history in a three dimensional region. The bandwidth in each direction can be calculated by using Eq. (2.59) and all the recorded collision locations. For criticality calculations the bandwidth from the previous cycle can be used to eliminate the storage requirement. In a similar exercise we can derive the 3-D KDE track length flux tally and its variance:

$$\phi(x,y,z)=\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}d_{i,c}}{n}\sum_{j=1}^{n}\frac{1}{h_x}k\left(\frac{x-X_{i,c,j}}{h_x}\right)\frac{1}{h_y}k\left(\frac{y-Y_{i,c,j}}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_{i,c,j}}{h_z}\right), \tag{3.30}$$

$$\sigma^2_{\phi(x,y,z)}=\frac{1}{N-1}\left[\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{c=1}^{c_i}\frac{w_{i,c}d_{i,c}}{n}\sum_{j=1}^{n}\frac{1}{h_x}k\left(\frac{x-X_{i,c,j}}{h_x}\right)\frac{1}{h_y}k\left(\frac{y-Y_{i,c,j}}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_{i,c,j}}{h_z}\right)\right\}^2-\left(\phi(x,y,z)\right)^2\right]. \tag{3.31}$$

3.4 Numerical Results for 1-D KDE Collision and Track Length Estimators

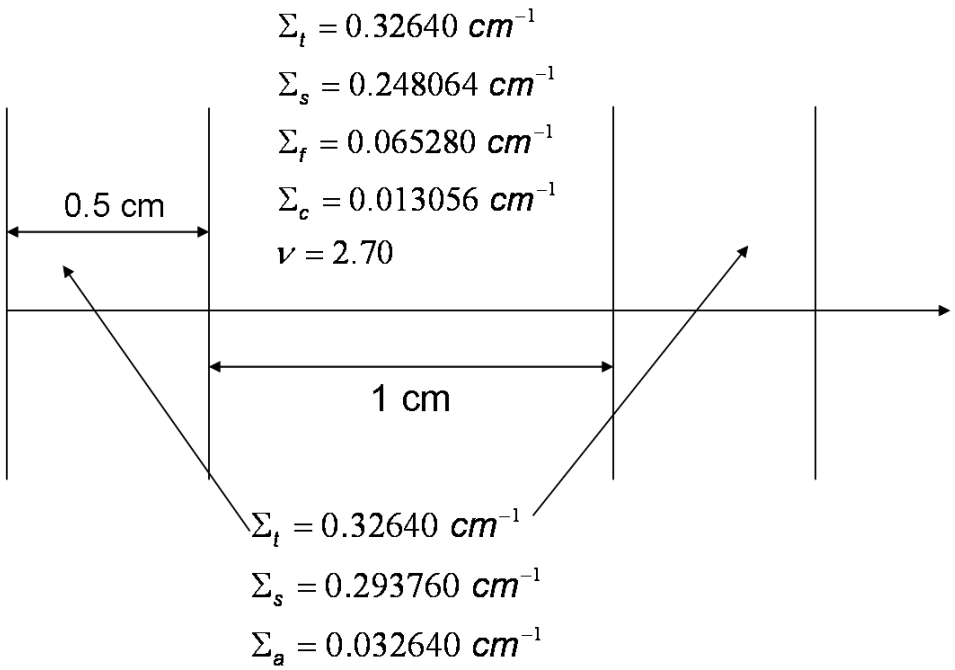

Figure 3.2 1-D example 1: One-dimensional and one energy group representation of a fuel pin.

In all the following 1-D examples, the Epanechnikov kernel is used as the kernel function k. This kernel function is given by Eq. (2.24). The bandwidths for all the following problems are calculated by using Eq. (2.21). All the following numerical examples of KDE tallies are one energy group. The energy variable is not a concern in this research, as the primary goal is to test the effectiveness of KDE tallies for extracting the shape information from Monte Carlo simulation. KDE has problems at boundaries due to the potential overlapping of an individual kernel and the boundary. To remedy this, we have used the boundary kernel method, thoroughly discussed in Chapter 2, as the boundary correction for the KDE tallies. Note that we could also use the reflecting boundary correction method, as all the 1-D examples have reflecting boundaries.

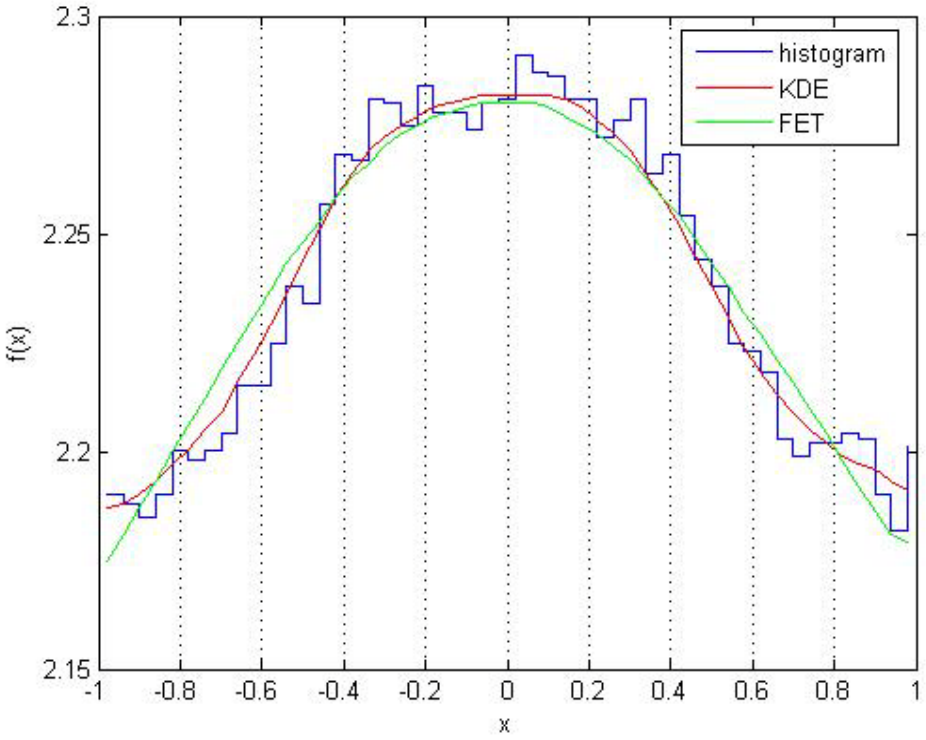

Figure 3.3 One energy group flux distribution inside a 1-D fuel region surrounded by water by collision estimators.

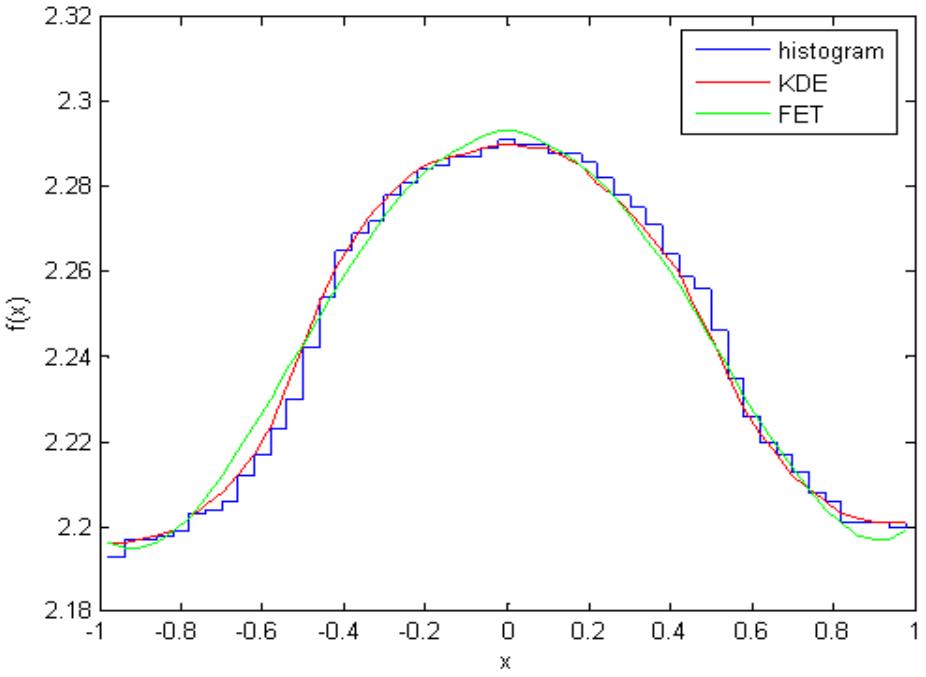

Figure 3.4 One energy group flux distribution inside a 1-D fuel region surrounded by water by track length estimators.

Figure 3.2 illustrates the first 1-D example, where the scalar flux is estimated inside a pure fuel region (material properties are given in Figure 3.2) surrounded by water and with reflecting boundaries to represent an infinite lattice of fuel pins. The simulation used 10,000 particles per batch for 150 batches with the first 50 batches discarded.

The flux in Figure 3.3 is calculated by the KDE collision tally, given by Eq. (3.21), and is averaged over the active cycles. The KDE and FET [1] flux is calculated by banking all the collision points during a cycle and using those points as observations from the actual flux distribution. The basic concepts of FET will be discussed in the convergence analysis section of this chapter. It is clear from Figure 3.3. that qualitatively the performance of KDE is better than FET. The KDE track length estimator is implemented in Figure 3.4 with $n = 2$ (2 sub-tracks) and with Eq. (3.26). Figure 3.4 also confirms the effectiveness of the KDE track length estimator for depicting the flux distribution inside a 1-D fuel pin region.

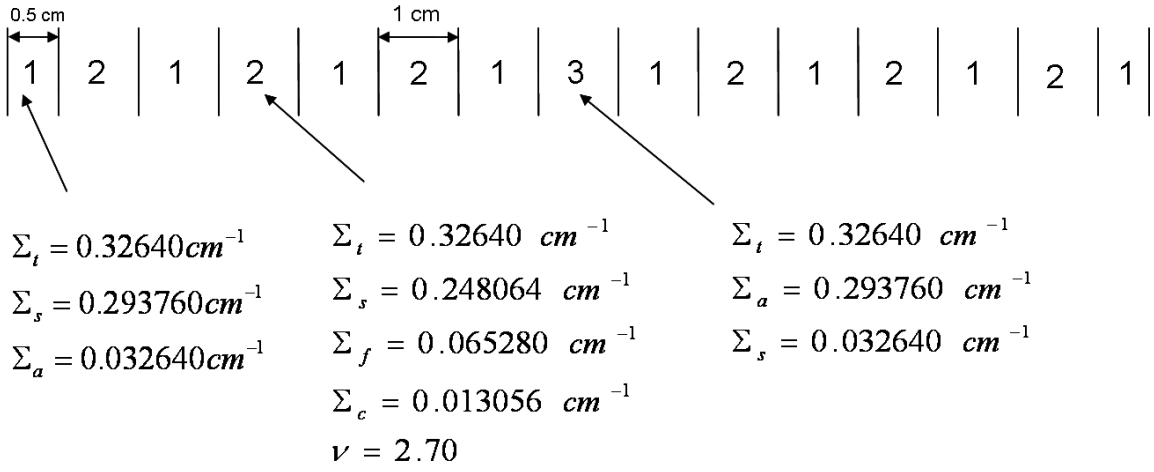

Figure 3.5 1-D example 2: One-dimensional and one energy group representation of a fuel lattice containing a strong absorber.

In the second 1-D example, an array of 6 1-D fuel regions, each surrounded by water and with reflecting external boundaries to represent an infinite lattice, is used to estimate the scalar flux using KDE and FET. At the center of the array there is a strong neutron absorber, which is used to simulate a control pin. Figure 3.5 illustrates the problem geometry and specifies the material properties.

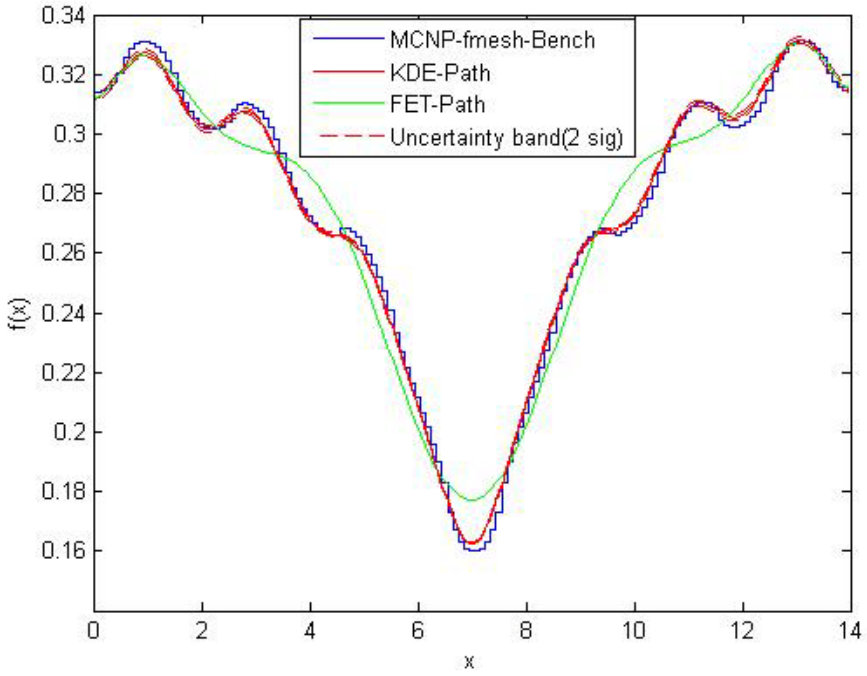

Figure 3.6 One group flux distribution inside a 1-D array of fuel and water with a strong neutron absorber using the track length estimator.

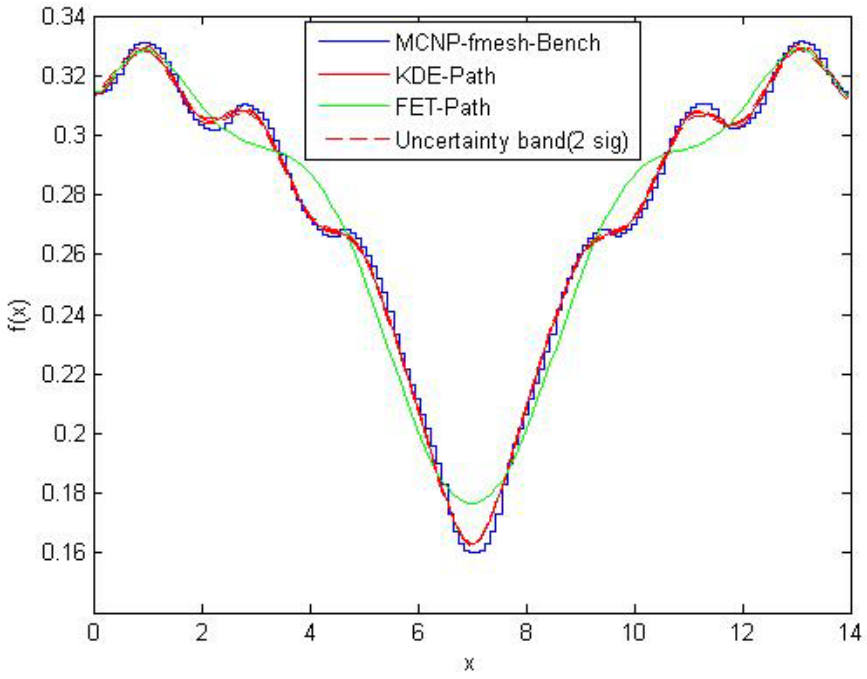

Figure 3.7 KDE track length scalar flux estimator without storing the points from the tracks in a neutron cycle.

For a reference solution, a one energy group MCNP5 [12] kcode calculation is used with 500 cycles (200 inactive) and 200,000 histories per cycle. The scalar flux is estimated by employing the FMESH tally with 140 fine meshes in the x-direction. The KDE and FET estimations employed 30,000 histories per cycle for 200 cycles with the first 100 cycles discarded. The number of sub-tracks n is equal to 4 for all the KDE track length tallies. Twelve Legendre expansion coefficients are used for the FET calculation.

In Figure 3.6 the KDE-track-length estimator is calculated by using a single (global) bandwidth over the entire problem domain. In this case all the points extracted from the tracks during an active cycle need to be stored in order to calculate the bandwidth. It is evident from Figure 3.6 that global KDE tallies can handle material discontinuities inside the tally region.

At the end of each active cycle, the bandwidth is calculated from the stored data and is then used for the subsequent KDE flux calculation for that cycle. The plotted flux is the average over the active cycles. That means at each cycle we need to store the data points, either collision points or points from tracks, and their weights, in order to calculate the bandwidth at the end of the cycle. This storage requirement can be eliminated during a neutron cycle by using the bandwidth from the previous cycle. As the bandwidth depends mainly on the standard deviation of the data and number of data points, the bandwidth can be calculated in a cycle without storing the data points and that bandwidth can be used to score the KDE flux in the next cycle. The KDE-track-length tally without storing the points is illustrated for the same problem in Figure 3.7. The KDE estimators can be slightly improved by using region-based bandwidths instead of one global bandwidth. The KDE collision and KDE track-length estimators with region-based bandwidths, where σ and N in Eq. (2.21) are calculated for each region, are illustrated in Figures. 3.8 and 3.9. The reference solution for the collision tally is obtained with a benchmark calculation with 140 histogram bins in the x-direction, with 100,000 histories per cycle with 400 cycles and 200 discarded cycles. Note that for KDE estimation the internal shape of the flux is not dependent on placing bins within the material regions. The boundary correction for the KDE estimators is only used for the external boundaries and not for the internal material boundaries.

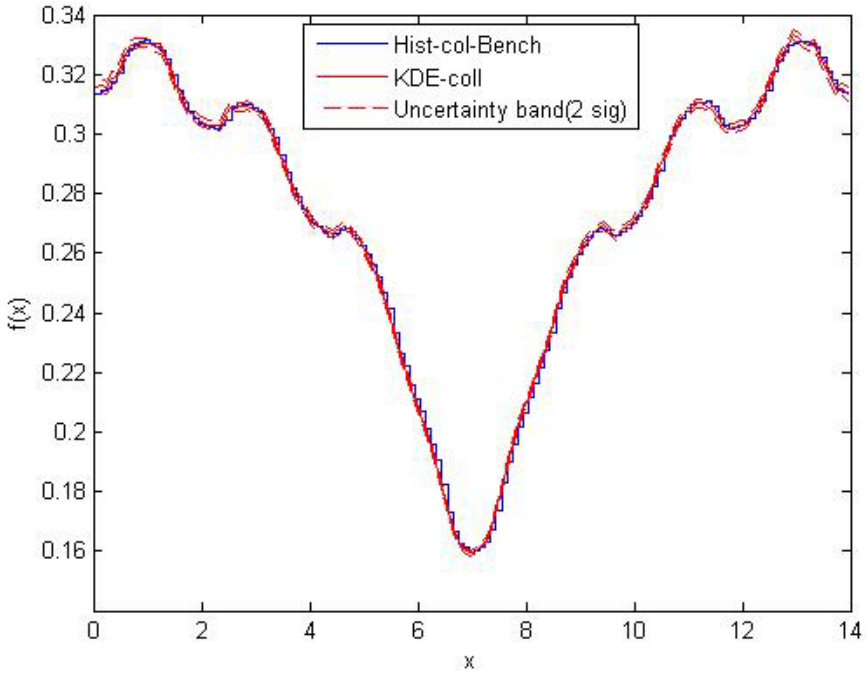

Figure 3.8 KDE collision scalar flux estimator without storing the collision points in a neutron cycle and with region-based bandwidths.

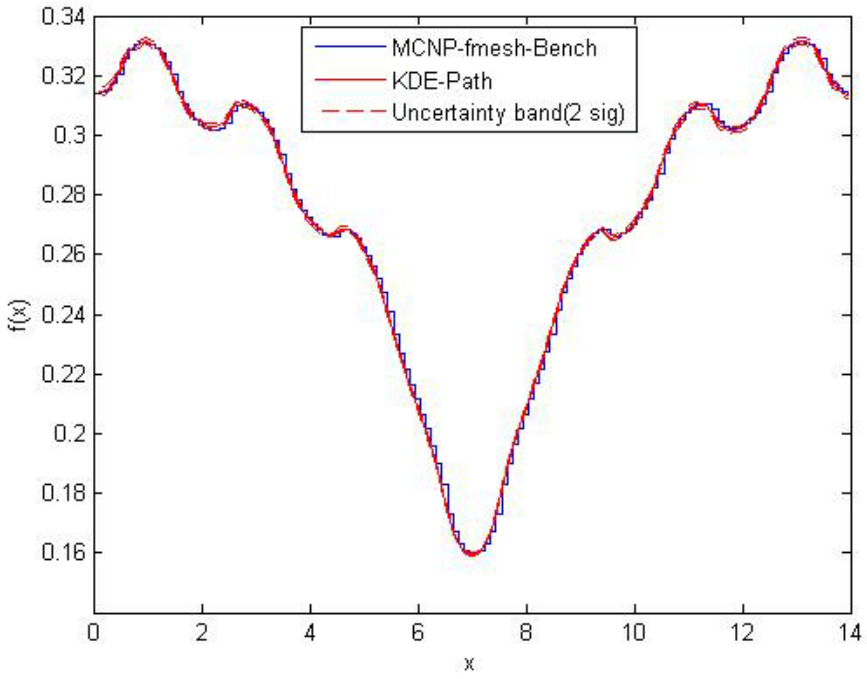

Figure 3.9 KDE track length scalar flux estimator without storing the collision points in a neutron cycle and with region-based bandwidths.

3.5 Numerical Results for 2-D KDE Collision and Track Length Estimators

For testing purposes, both KDE track length and collision estimators are used to create the 2-D flux across the face of a pin cell. Figure 3.10 illustrates the top view of the problem geometry, with a fuel pin of radius $0.603\,\text{cm}$ placed in a square cell of pitch $1.875\,\text{cm}$. The fuel pin has the same fissile material as in the previous 1-D problems and it is surrounded by water with the same cross sections as in the 1-D problems. The Epanechnikov kernel is used as the kernel function in both x and y directions. The bandwidth is calculated by using Eq. (2.59). The KDE collision and track length flux is compared with a highly resolved MCNP5 [12] FMESH calculation with 1600 mesh tally regions (40 in x direction and 40 in y direction). The KDE and MCNP5 simulations employed 20,000 histories per cycle for 400 cycles with the first 200 cycles discarded. The number of sub-tracks n is equal to 4 for the KDE track length tallies. In this case, global KDE is used with one bandwidth for the entire tally region and the boundary kernel method is used as the boundary correction for the four external boundaries.

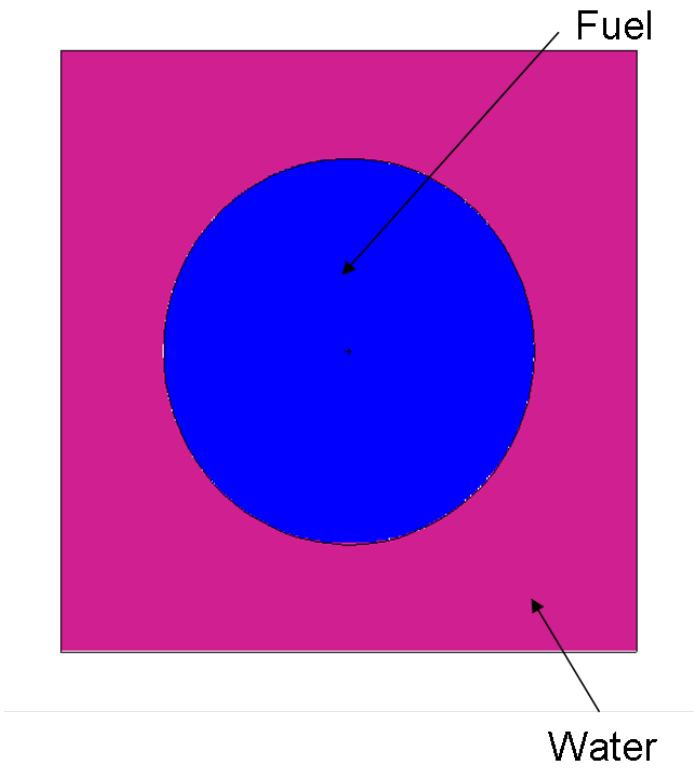

Figure 3.10 2-D Example 1: Two dimensional representation of fuel pin cell.

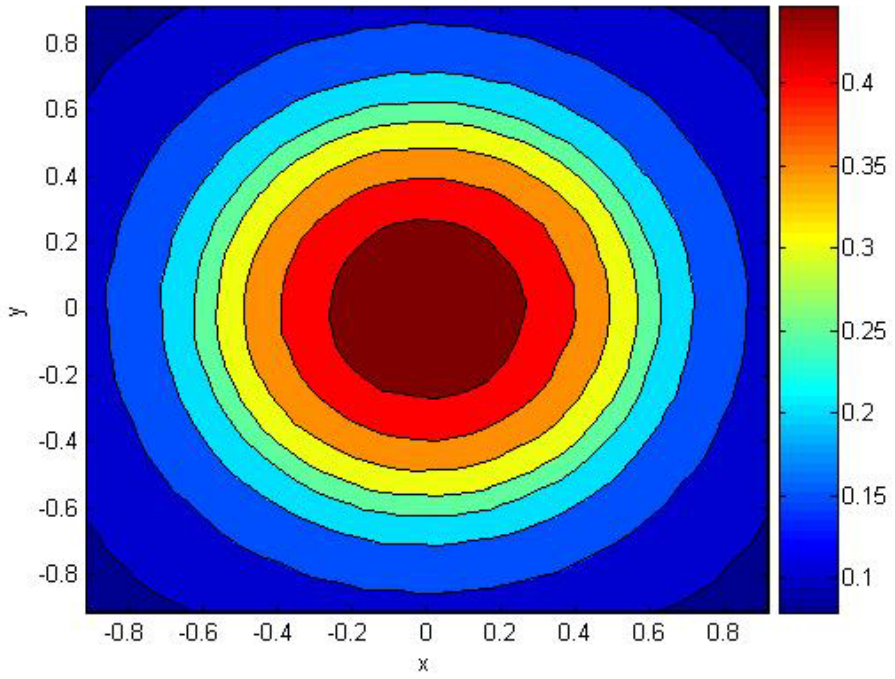

Figure 3.11 MCNP5 flux distribution by plotting isofluxes over the face of the pin cell.

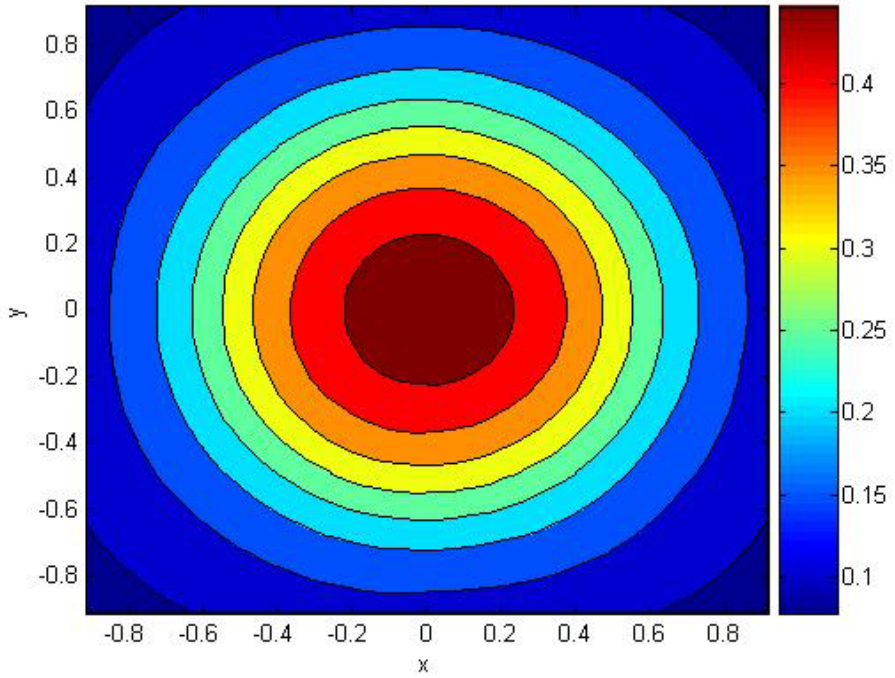

Figure 3.12 KDE collision flux distribution by plotting the isofluxes over the face of the pin cell.

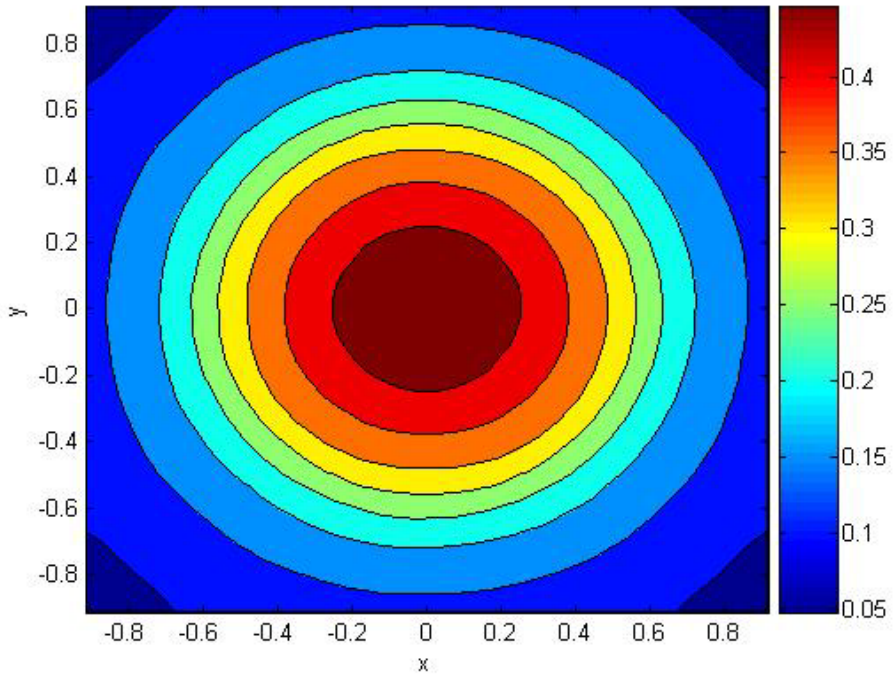

Figure 3.13 KDE track length flux distribution by plotting the isofluxes over the face of the pin cell.

Figure 3.11 illustrates the MCNP5 solution and Figures 3.12 and 3.13 present the KDE collision and track length solutions, respectively. For all three cases MATLAB7 [13] contour plot option is used, which plots the equal flux contours, or isofluxes, from a matrix. From the 2-D flux plot we can conclude, for the same number of histories, a smoother flux distribution is obtained by using global KDE collision and track length tallies. Even the KDE collision tally produces a smoother flux than the MCNP5 FMESH tally, which is basically a histogram tally with track length estimator. It is important to note that the KDE tallies do not require a 2-D bin structure and the KDE tallies are obtained without storing any points and by using a bandwidth from the previous criticality cycle.

In order to provide a more challenging test of the capabilities of the 2-D KDE track length tally, a problem was developed to simulate a realistic PWR fuel assembly. A two-dimensional representation of a quarter 16×16 assembly is shown in Figure 3.14. The fuel assembly shown in Figure 3.14 has 6 strong absorber pins (control elements) that are shown in yellow.

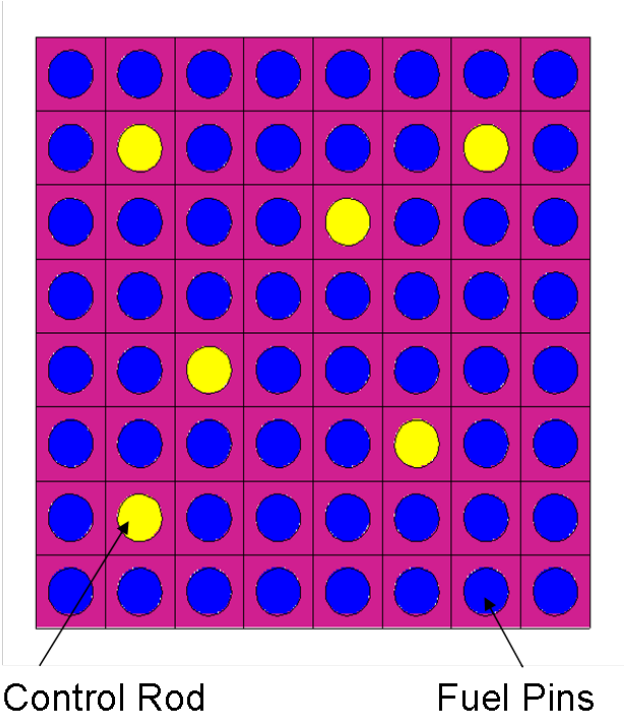

Figure 3.14 Two dimensional representation of a PWR fuel assembly with all 6 control rods inserted. Reflecting boundary conditions are applied to all sides in order to simulate a repeating lattice of assemblies.

The neutron absorber material (yellow pins) has the same properties as the neutron absorber used in the 1-D lattice problem. The fuel pins and the surrounding moderator also have the same cross sections as in the 1-D cases. The quarter assembly illustrated in Figure 3.14 contains 64 individual fuel and poison elements. With this many separate regions, it may not be expected that the global KDE track length tally with a single global bandwidth for the entire assembly will produce an acceptable flux distributions. But surprisingly it does produce good results – with the global KDE it is possible to distinguish the individual pins. Figure 3.15 illustrates the MCNP5 FMESH solution. The KDE flux approximations are shown in Figure 3.16. Figure 3.15 shows the flux depressions inside the control pins and shows the movement of the peak flux away from the control pins. For the last case, which is illustrated by Figure 3.17, 4 of the control rods are removed from the assembly and the guide channels are filled with water. The corresponding MCNP5 and KDE flux plots are shown by Figures 3.18 and 3.19, respectively. All the quarter assembly cases used 50,000 histories per cycle for 400 cycles with the first 200 cycles discarded and the number of sub-tracks n was equal to 4 .

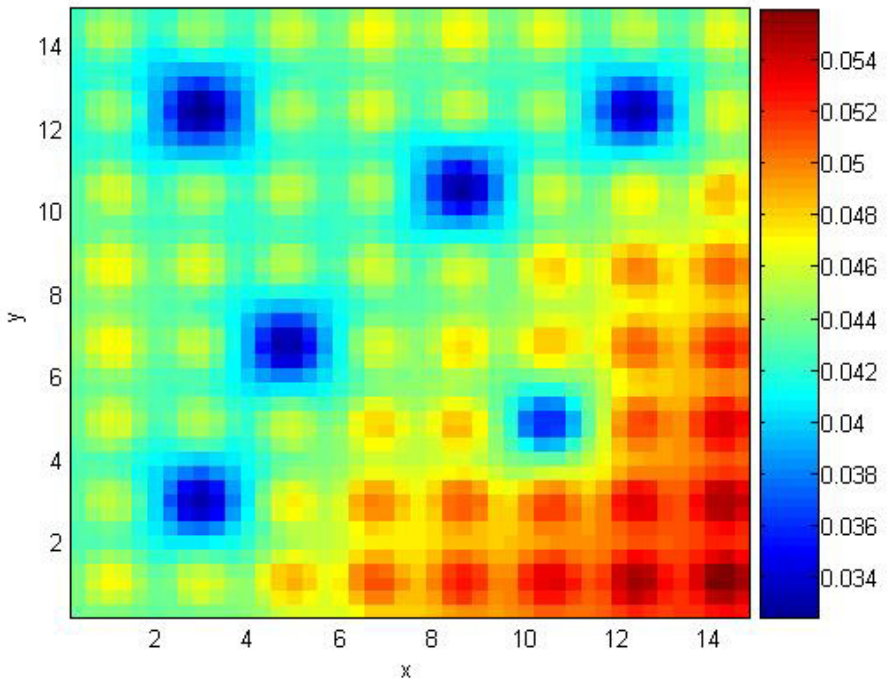

Figure 3.15 2-D MCNP5 flux distribution across the face of the fuel assembly with all the control rods inserted.

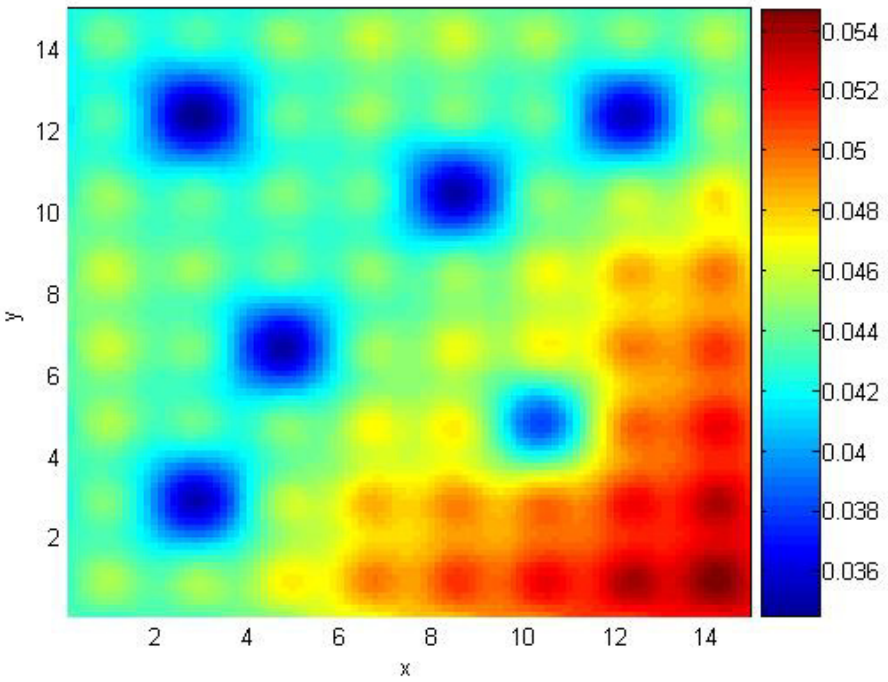

Figure 3.16 2-D global KDE flux distribution across the face of the fuel assembly with all the control rods inserted.

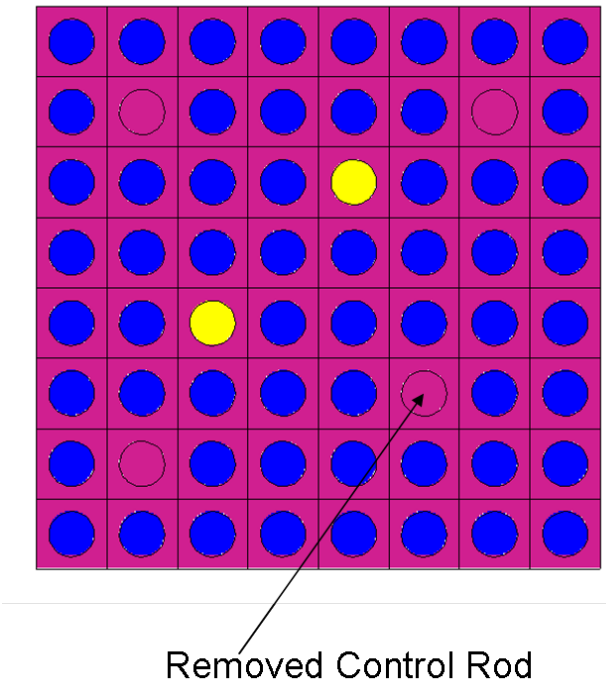

Figure 3.17 Two dimensional representation of a PWR fuel assembly with only 2 control rods inserted. Reflecting boundary conditions are applied to all sides in order to simulate a repeating lattice of assemblies.

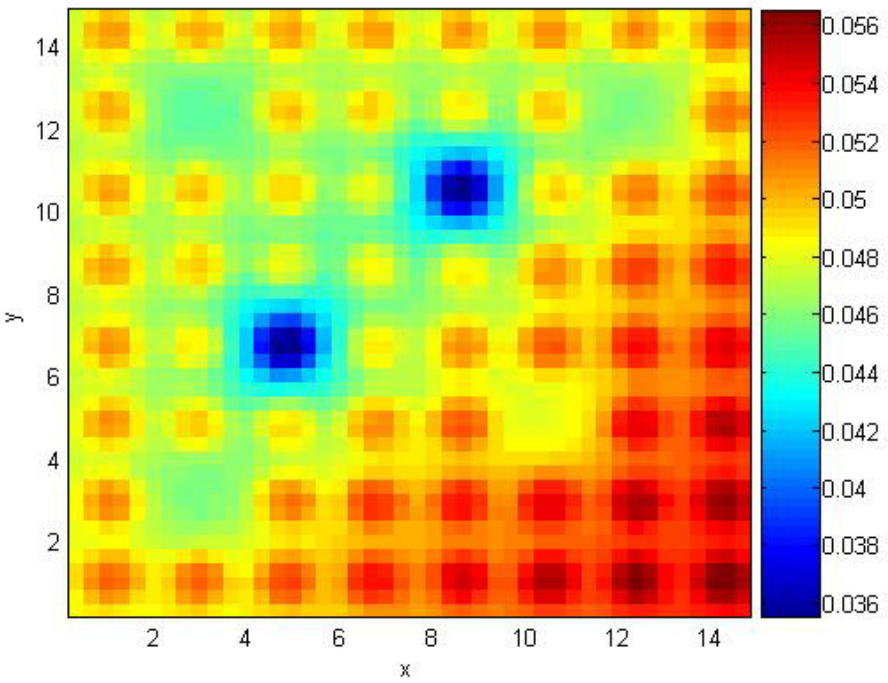

Figure 3.18 2-D MCNP5 flux distribution across the face of the fuel assembly with 4 of the control rods removed from the assembly.

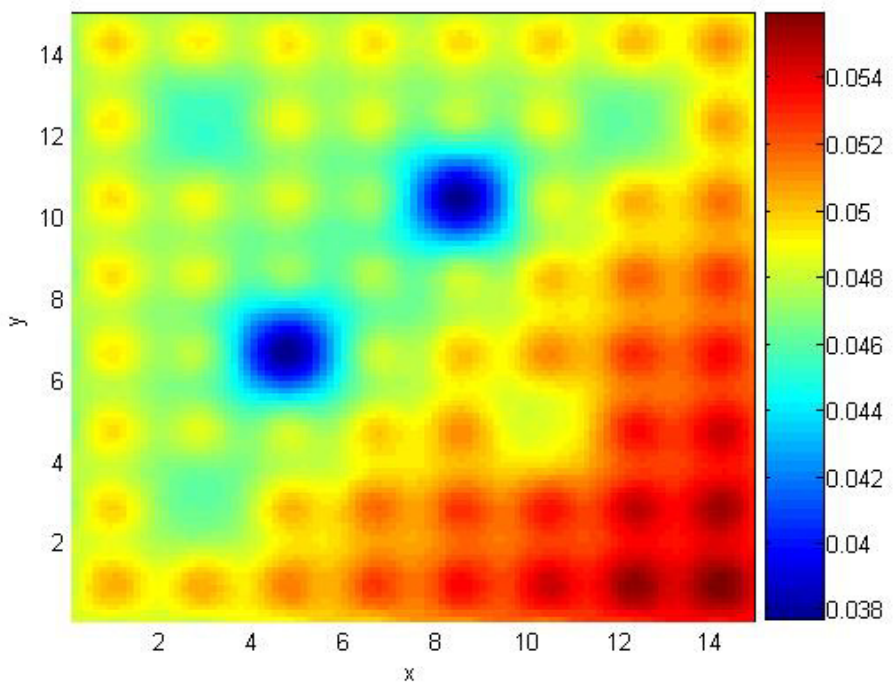

Figure 3.19 2-D global KDE flux distribution across the face of the fuel assembly with 4 of the control rods removed from the assembly.

3.6 Convergence Analysis

The Mean Integrated Square Error (*MISE*) is used as a measure of the discrepancy between the actual and the estimated density. *MISE* will be used in this section to compare the convergence of three Monte Carlo tallying techniques, namely KDE, FET, and histogram. The *MISE* was defined in Chapter 2 as follows

$$MISE(\hat{f}) = E\int\left\{\hat{f}(x)-f(x)\right\}^2 dx = \int\left\{E\hat{f}(x)-f(x)\right\}^2 dx + \int \mathrm{var}\,\hat{f}(x)dx \qquad (3.32)$$

The actual density is given by $f(x)$, and the estimated density is denoted $\hat{f}(x)$. The term E is the expectation operator and the limit of the integration, if not explicitly given, is from negative infinity to positive infinity, which has been assumed for simplicity. All the analyses are also applicable to the boundary kernels used for correcting the boundary bias for KDE. These analyses are easily extended to multi-variate densities as was discussed in Chapter 2.

3.6.1 Convergence Analysis of the Global KDE Tally

Convergence properties for KDE are discussed in Chapter 2. In this section the important results related to the KDE convergence will be reviewed without going into the details.

For each x, $\hat{f}(x)$ in Eq. (2.8) can be considered as a random variable that depends on the observations $X_1, ..., X_N$. The bias and variance of the estimator are given by

$$bias \approx \frac{1}{2} h^2 f''(x) k_2 + O\left(h^3\right), \tag{3.33}$$

$$\text{var}\, \hat{f}(x) \approx \frac{1}{Nh} f(x) \int k(z)^2 dz + O\left(\frac{1}{N}\right). \tag{3.34}$$

k_2 is defined by Eq. (2.9). The *MISE* can be approximated easily by combining Eqs. (3.32), (3.33), and (3.34) to find:

$$MISE\left(\hat{f}\right) \approx \frac{1}{4} h^4 k_2^2 \int f''(x)^2 dx + \frac{1}{Nh} \int k(z)^2 dz. \tag{3.35}$$

The basic assumption here is that the true density f is such that its second derivative f'' is continuous and square integrable. It is straightforward to derive the optimum bandwidth by minimizing the *MISE*:

$$h_{opt} = \frac{\left\{\int k(z)^2 dz\right\}^{1/5}}{k_2^{2/5} \left\{\int f''(x)^2 dx\right\}^{1/5}} N^{-1/5}. \tag{3.36}$$

A natural approach for calculating the bandwidth h is to use a standard family of distributions, like the Gaussian distribution, to obtain a value of the term $\int f''(x)^2 dx$ in the Eq. (3.36). We can derive an expression for the bandwidth h by also using a Gaussian function as the kernel function k:

$$h = 1.06 \sigma N^{-1/5}. \tag{3.37}$$

The quantity σ^2 is the variance of the normal distribution and can be estimated by the usual standard deviation of the data or by some more robust estimators. Inserting the

optimum bandwidth into the *MISE* equation (Eq. (3.35)) we can arrive at the final approximate expression for *MISE*:

$$\begin{aligned} MISE\left(\hat{f}\right)_{KDE} &= \frac{5}{4}C(k)\left\{\int f''(x)^2\,dx\right\}^{1/5} N^{-4/5} \\ &= O\left(N^{-4/5}\right), \end{aligned} \tag{3.38}$$

where

$$C(k) = k_2^{2/5}\left\{\int k(z)^2\,dz\right\}^{4/5}. \tag{3.39}$$

3.6.2 Convergence Analysis of the FET Tally

This analysis is a variation on the analysis performed by Griesheimer et al. [7]. If $[\psi_n]_0^\alpha$ is a complete orthogonal set with respect to some weighting function $\rho(x)$ in $L_\rho^2(\Gamma)$, which is the space defined by all square integrable functions over some bounded domain Γ, then any $f(x) \in L_\rho^2(\Gamma)$ can be written as

$$f(x) = \sum_{n=0}^{\infty} \overline{a}_n k_n \psi_n(x), \tag{3.40}$$

where $\overline{a}_n$ is the nth expansion coefficient defined by (using the orthogonality properties of the basis functions)

$$\overline{a}_n = \int \psi_n(x)\rho(x)f(x)\,dx \tag{3.41}$$

and k_n is the normalization constant for the nth basis function [7]. As it is not possible to calculate an infinite number of terms, the estimator ($\hat{f}(x)$) of the true function f is always truncated to some finite number M :

$$\hat{f}(x) = \sum_{n=0}^{M} \hat{a}_n k_n \psi_n(x), \tag{3.42}$$

where

$$\hat{a}_n = \frac{1}{N}\sum_{i=1}^{N} \psi_n(X_i)\rho(X_i). \tag{3.43}$$

It is easy to derive the first term of the *MISE* expression in Eq. (3.32):

$$\int \left\{ E\hat{f}(x) - f(x) \right\}^2 \rho(x)\, dx = \int \left(\sum_{n=M+1}^{\infty} k_n \bar{a}_n \psi_n(x) \right)^2 \rho(x)\, dx$$
$$= \int \left(\sum_{n=M+1}^{\infty} \sum_{m=M+1}^{\infty} k_n k_m \bar{a}_n \bar{a}_m \psi_n(x) \psi_m(x) \right) \rho(x)\, dx \quad (3.44)$$
$$= \sum_{n=M+1}^{\infty} \bar{a}_n^2 k_n .$$

Equation (3.44) is obtained by using the orthogonality property of the basis functions. The true expansion coefficients converge at a rate determined by κ [1, 14]

$$\left| \bar{a}_n \right| = O\left(\frac{1}{n^{\kappa}} \right). \quad (3.45)$$

The value of κ, which is called the algebraic index of convergence, is generally equal to the number of derivatives of the true density function f that are square integrable [1, 14]. Using Eqs. (3.44) and (3.45) the integral of the square bias term can be written as

$$\int \left\{ E\hat{f}(x) - f(x) \right\}^2 \rho(x)\, d\,\, \text{ø} \sum_{n=M+1}^{\infty} O\left(\frac{1}{n^{2\kappa}} \right) \frac{2n+1}{2}, \left[\text{for Legendre polynomials} \right]$$
$$\approx O\left(\frac{1}{M^{2\kappa-2}} \right). \quad (3.46)$$

The last line in Eq. (3.46) is obtained by using an infinite series summation [15]. The second term of the MISE in Eq. (3.32) can be written as [7]

$$\int \text{var}\, \hat{f}(x) \rho(x) dx \approx O\left(\frac{M}{N} \right). \quad (3.47)$$

Finally, the results from Eqs. (3.46) and (3.47) can then be used to write the *MISE* of FET as

$$MISE\left(\hat{f} \right)_{FET} = O\left(\frac{1}{M^{2\kappa-2}} \right) + O\left(\frac{M}{N} \right). \quad (3.48)$$

The *MISE* of FET can be minimized over M by simple calculus

$$M_{opt} = O\left(N^{1/2\kappa-1} \right). \quad (3.49)$$

Substituting the optimum *M* into Eq. (3.48) we obtain

$$MISE\left(\hat{f}\right)_{FET} = O\left(N^{-\frac{2\kappa-2}{2k-1}}\right). \tag{3.50}$$

3.6.3 Convergence Analysis of the Histogram Tally

Let us assume N_b is the number of observations in bin b and M is the total number of bins. Also, N is the total number of observations and $\Delta x \equiv L/M$ (where L is the length of the domain) is the width of each bin. The width of the bins is taken constant for this analysis. The bias of the histogram estimator for bin b can be written as

$$\begin{aligned} bias_{M,b} &= E\hat{f}^{hist}_{M,b} - f(x) \\ &= \frac{1}{\Delta x}\int_{x_{b-1}}^{x_b} f(x)d \quad \text{-} f(x) \\ &\approx \frac{1}{\Delta x}\Delta x f\left(x_{b-1/2}\right) - \left\{ f\left(x_{b-1/2}\right) + \left(x - x_{b-1/2}\right) f'\left(x_{b-1/2}\right) + \frac{\left(x - x_{b-1/2}\right)^2}{2} f''\left(x_{b-1/2}\right) + \ldots \right\} \\ &\approx O\left(\Delta x\right). \end{aligned} \tag{3.51}$$

It is now simple to obtain an expression for the first term of the MISE in Eq. (3.32) for the histogram estimator:

$$\int bias^2\left(x\right)dx \approx \sum_{b=1}^{M} O\left(\Delta x^2\right)\Delta x \approx \sum_{b=1}^{M} O\left(\frac{L^3}{M^3}\right) \approx \sum_{b=1}^{M} O\left(\frac{1}{M^3}\right) = O\left(\frac{1}{M^2}\right). \tag{3.52}$$

The histogram estimator can be expressed as

$$\hat{f}^{hist}\left(x\right) = \frac{1}{N\Delta x}\sum_{i=1}^{N} I\left\{X_i \in b\right\}, \quad x \in \left[x_{b-1}, x_b\right]. \tag{3.53}$$

The random variable $I\left\{X_i \in b\right\}$ is associated with Bernoulli's distribution, whose variance is given by $p(1-p)$ [8], where p can be written as

$$p = \int_{x_{b-1}}^{x_b} f(x)dx. \tag{3.54}$$

The variance of the histogram estimator can be calculated as

$$\begin{aligned}\operatorname{var}\hat{f}^{hist}(x) &= \frac{1}{N^2\Delta x^2}\sum_{i=1}^{N}\operatorname{var} I \\ &= \frac{1}{N^2\Delta x^2} N\left\{\int_{x_{b-1}}^{x_b} f(x)dx\left(1-\int_{x_{b-1}}^{x_b} f(x)dx\right)\right\} \\ &\approx \frac{1}{N\Delta x} f\left(x_{b-1/2}\right).\end{aligned} \tag{3.55}$$

It is now straightforward to calculate the second term of MISE in Eq. (3.32) for a histogram:

$$\int \operatorname{var}\hat{f}^{hist}(x)dx \approx \int \frac{1}{N\Delta x} f\left(x_{b-1/2}\right)dx \approx O\left(\frac{M}{N}\right). \tag{3.56}$$

Equations (3.32), (3.52) , and (3.56) yield

$$MISE\left(\hat{f}\right)_{hist} = O\left(\frac{1}{M^2}\right) + O\left(\frac{M}{N}\right). \tag{3.57}$$

By simple calculus we can easily derive the optimum number of bins, yielding

$$M_{opt} \approx O\left(N^{1/3}\right). \tag{3.58}$$

Substituting the value of M from Eq. (3.58) into Eq. (3.57) we find

$$MISE\left(\hat{f}\right)_{hist} = O\left(N^{-\frac{2}{3}}\right). \tag{3.59}$$

3.6.4 Comparison Among KDE, FET, and Histogram Results

The KDE, FET, and histogram approximations of the true distribution converge at different rates dependent on the problem parameters. However, for most of the cases the KDE tally outperforms the FET tally, which in turn outperforms the histogram tally. If we consider $\kappa = 2$, that means the underlying true density function *f* has two integrable derivatives, then from Eqs. (3.38) and (3.50) it is easy to conclude that the $MISE\left(\hat{f}\right)_{KDE}$ has a faster convergence rate than $MISE\left(\hat{f}\right)_{FET}$. Although the KDE *MISE* calculation assumed that the second derivative f'' of the true density is continuous and square integrable, it is evident from the figs. 3.6 – 3.9, 3.12, 3.13, 3.14, and 3.16 that a global

KDE tally can estimate the material discontinuity within the domain much more effectively than a global FET tally. For FET estimation the material discontinuity can only be taken care by piecewise FET approximation. A significant feature of the KDE tally, which is shared by the FET tally, is that internal structure within the bin can be obtained without resorting to bin refinement. However, the FET tally requires a knowledge of the bin boundaries since the expansion functions are defined over the bin while the KDE tally is independent of the bin boundaries because the kernels are defined only at the interaction point. Because of this, we believe that the KDE tally will allow the pathlength estimator to be used when Woodcock tracking [16] is employed. Finally, the KDE performance can be improved further by using higher order kernels. In this analysis we assume k_2 (second moment of the kernel k) in Eq. (2.9) is not equal to zero. However if we relax this restriction, it is possible to construct k with $k_2 = 0$, which will further reduce the bias of the estimator. This has not been studied and is a topic for future research.

3.7 Numerical Verification of Convergence

We now present a series of numerical experiments to verify the convergence analyses of the three tallies discussed above. The following trial distribution $f(x)$ is used for each numerical experiment,

$$f(x) = \frac{1}{1.51985}\begin{cases} \cos(x)e^{2x+1}, & x \in [-1,-1/2], \\ \cos(x), & x \in [-1/2,1/2], \\ \cos(x)e^{-x/2+1/4}, & x \in [1/2,1]. \end{cases} \tag{3.60}$$

Figure 3.20 illustrates the shape of the trial distribution $f(x)$. This distribution is specifically chosen to simulate many Monte Carlo particle transport features encountered during the simulation [1]. In the first set of experiments the integrated bias square and the integrated variance for each of the estimators are calculated for the trial distribution $f(x)$. The integrated square bias and the integrated variance for KDE are plotted separately in Figure 3.21 vs. bandwidth h where $10,000$ observations from the trial distribution are used. The entire domain $[-1,1]$ is divided into $B(50)$ bins and the

density is estimated at the mid point of each bin. The calculation is repeated $n(50)$ times to estimate the expectation. The sum of the integrated square bias and the integrated variance, which is the *MISE*, is also plotted in the same plot. The *MISE* of the KDE has a minimum point which corresponds to the optimum bandwidth. The integrated bias for all three estimators is calculated as follows

$$\int \left\{ E\hat{f}(x) - f(x) \right\}^2 d \approx \sum_{i=1}^{B} \left\{ E\hat{f}(x_i) - f(x_i) \right\}^2 \Delta x \tag{3.61}$$

where Δx is the bin width. The variance is calculated as the sample variance.

The same experiment is carried out for FET and histogram tallies and results are illustrated in Figures 3.22 and 3.23. The minimum point of the *MISE* in Figure 3.22 corresponds to the optimum number of expansion coefficients for FET. The optimum number of bins for the histogram is predicated by the minimum *MISE* in Figure 3.23.

Finally, the *MISE* is plotted versus the number of histories in Figure 3.24 and is calculated as follows

$$MISE = \frac{\sum_{j=1}^{n} \sum_{b=1}^{B} \left\{ \hat{f}_{b,j} - f_b \right\}^2 \Delta x}{n} \tag{3.62}$$

The optimum values of the bandwidth (KDE), the number of terms (FET), and the number of bins (histogram), which vary with the number of histories, are used in this plot (Figure 3.24). Figure 3.24 confirms that for the given trial distribution f, $MISE\left(\hat{f}\right)_{KDE}$ converges faster than both FET and histogram tallies.

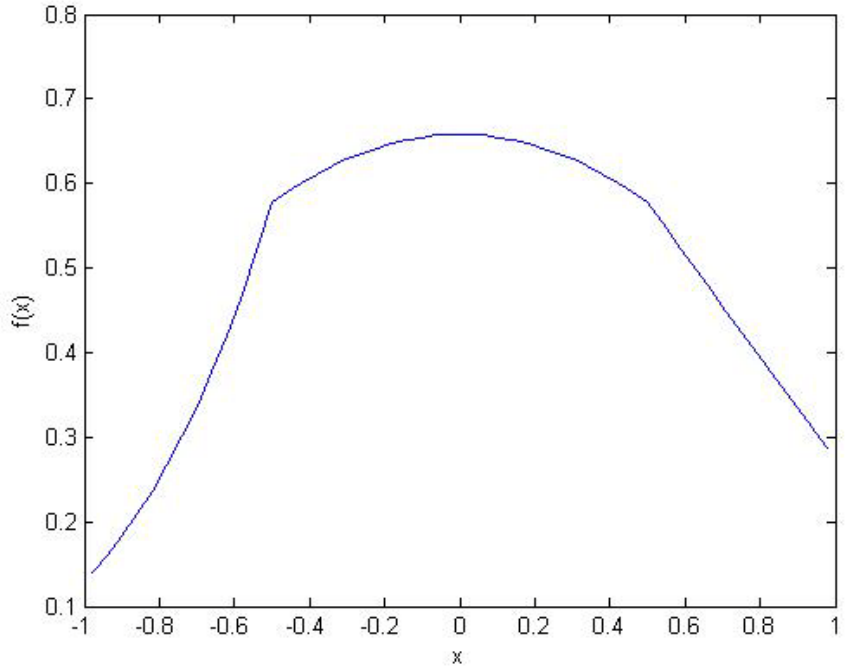

Figure 3.20 Plot of the trial distribution $f(x)$ used for numerical verification of the convergence of KDE, FET, and histogram (from Griesheimer [1]).

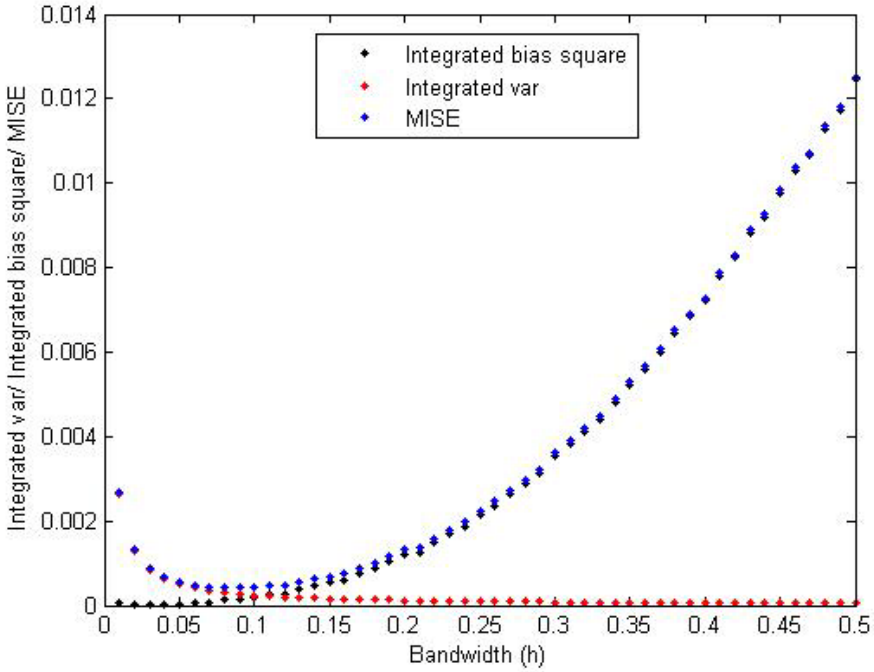

Figure 3.21 Plot of integrated variance, integrated bias square, and *MISE* vs. bandwidth for KDE.

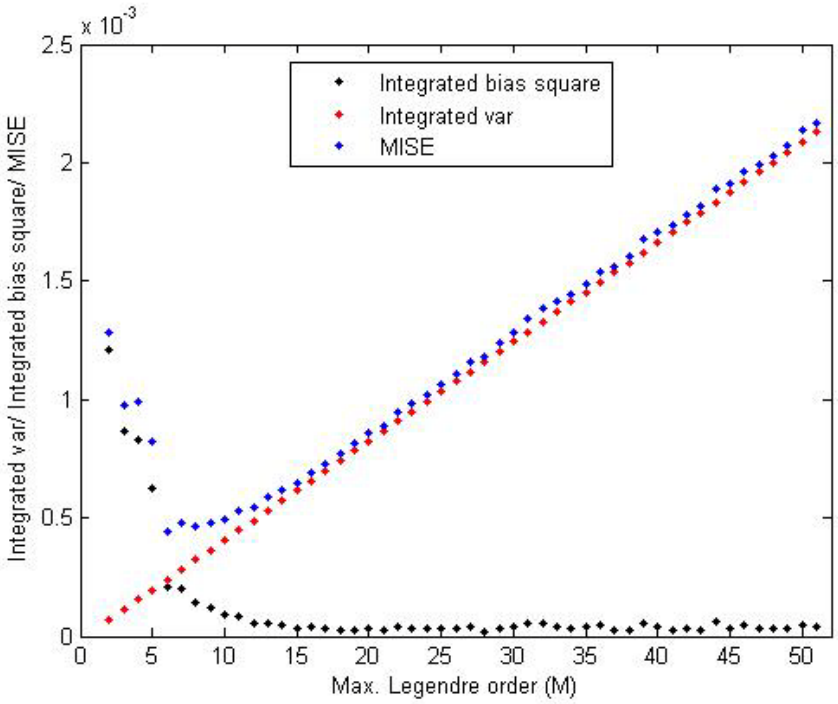

Figure 3.22 Plot of integrated variance, integrated bias square, and *MISE* vs. number of expansion coefficients for FET.

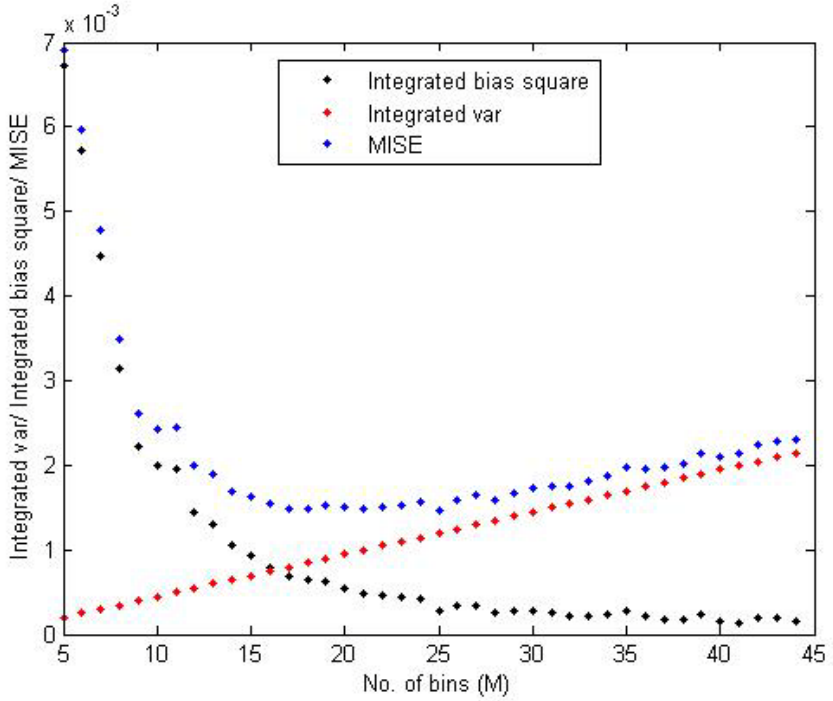

Figure 3.23 Plot of integrated variance, integrated bias square, and *MISE* vs. number of bins for histogram.

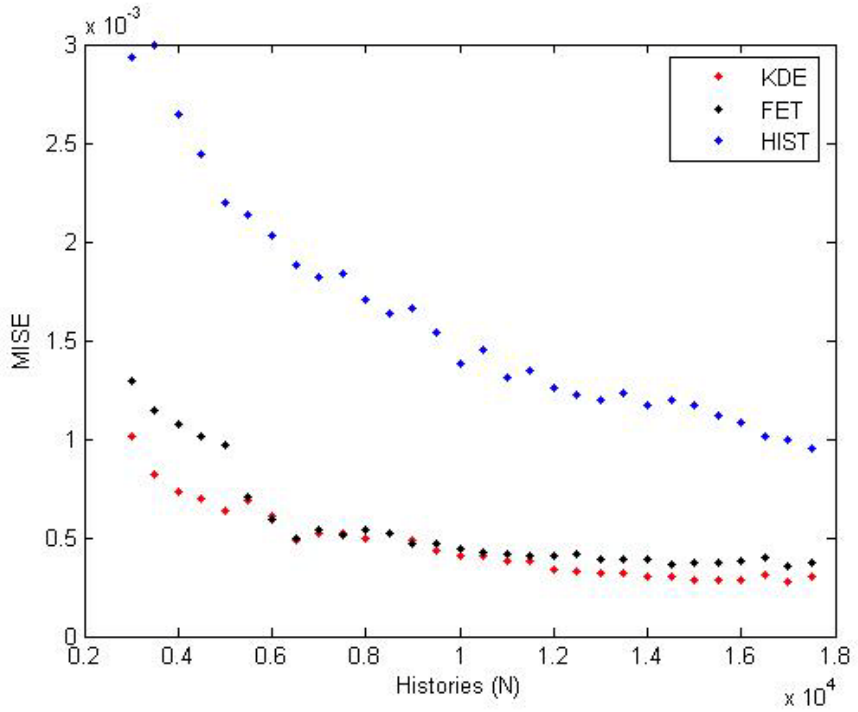

Figure 3.24 Plot of *MISE* vs. number of histories N for KDE, FET, and histogram tallies.

3.8 References

[1] D. P. Griesheimer, *Functional Expansion Tallies for Monte Carlo Simulations*, Ph.D. Thesis, Nuclear Engineering and Radiological Sciences, University of Michigan (2005).

[2] K. Banerjee and W. R. Martin, "Monte Carlo Global Scalar Flux Estimation with Kernel Density Estimator," *Trans. Am. Nucl. Soc.*, Reno, NV, 9-13th November, Vol. **99**, pp. 346-347 (2008).

[3] K. Banerjee and W. R. Martin, "Kernel Density Estimated Global Flux Tallies," *Proc. M&C Topical Meeting*, American Nuclear Society, Saratoga Springs, NY (May 2009).

[4] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London (1986).

[5] P. Hall, "On the Rate of Convergence of Orthogonal Series Density Estimators," *J. R. Statist. Soc. Series B (Methodology)*, **48**, pp. 115-122 (1986).

[6] P. J. Diggle and P. Hall, "The Selection of Terms in an Orthogonal Series Density Estimator," *Journal of the American Statistical Association*, **81**, pp. 230-233 (1986).

[7] D. P. Griesheimer, W. R. Martin, and J. P. Holloway, "Convergence Properties of Monte Carlo Functional Expansion Tallies," *Journal of Computational Physics*, **211**, pp. 129-153 (2006).

[8] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, Fourth ed, Tata McGraw-Hill, New Delhi (2002).

[9] J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*, Dover Publications, Mineola, New York (2007).

[10] S. A. Dupree and S. K. Fraley, *A Monte Carlo Primer – A Practical Approach to Radiation Transport*, Plenum, New York (2002).

[11] L. L. Carter and E. D. Cashwell, *Particle Transport Simulation with the Monte Carlo Method*, Technical Information Center, Energy Research and Development Administration, Oak Ridge (1975).

[12] Monte Carlo Team, "MCNP – A General Monte Carlo N-Particle Transport Code, Version 5," *LA-UR-03-1987*, Los Alamos National Laboratory (2003).

[13] The MathWorks, Inc. *MATLAB – The Language of Technical Computing*, Version 7.0.1.24707 (R14), Service Pack 1 (2004).

[14] J. P. Boyd, *Chebyshev & Fourier Spectral Methods*, Springer-Verlag, New York (1989).

[15] C. M. Bender and S. A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill, New York (1978).

[16] E.R. Woodcock, T. Murphy, P.J. Hemmings, T.C. Longworth, "Techniques Used in the GEM Code for Monte Carlo Neutronics Calculations in Reactors and Other Systems of Complex Geometry," *Proc. Conf. Applications of Computing Methods to Reactor Problems*, ANL-7050, pp. 557, Argonne National Laboratory (1965).

CHAPTER 4

Kernel Density Estimation Method for Monte Carlo Tallies with Unbounded Variance

4.1 Introduction

For many nuclear engineering applications, for example in shielding calculations, it is important to calculate the scalar flux or equivalent quantities at a point. The conventional point detector estimator, also known as the next-event estimator, is an analytical estimator which contributes to a scalar flux estimator at the location of the point detector for every source or collision event [1]. This estimator has a r^2 term in the denominator, where r is the distance from the source or collision point to the detector point, which results in a singularity that makes the theoretical variance of this estimator infinite [2]. The infinite variance implies that the Central Limit Theorem is not valid and cannot be used to form a confidence interval for convergence to the true scalar flux at the detector point. The impact of this singularity on a Monte Carlo simulation is that estimates of the flux and its variance may not be reliable, depending on the number of source particles or collisions close to the detector point.

Similarly, the surface crossing flux tally, which gives an estimate for the average flux over a surface, has a $1/|\mu|$ singularity , where μ is the cosine of the angle between the direction of flight and the surface normal, again yielding an unbounded variance [3].

Kalos [2] proposed the once-more-collided estimator to remedy the unbounded variance in the point detector tally. This method employs the sampling of an imaginary intermediate collision point for every real collision. By forming a suitable probability density function (pdf) for sampling the intermediate points, an estimator with only $1/r$ singularity can be achieved. Kalli and Cashwell [4] studied point detector estimators extensively. Steinberg and Kalos [5] proposed a method to bias the selection of the

collision points in the random walk toward the point detector. Rief et. al. [6] proposed a method of tallying the flux at the detector location using neutron flight paths as line sources, resulting in a $1/r$ singularity for the estimator. The theoretical variance of this estimator is also unbounded due to an angular singularity. However, it was claimed that the convergence of this estimator was similar to a bounded estimator, this is $1/N$.

The conventional approach to estimate the unbounded tallies is to break up the angular domain into two regions and use an approximate estimator near the singularity, yielding a bounded variance. The MCNP5 [7] point detector estimator (F5 estimator) is based on this approach. MCNP5 handles this problem by modifying the conventional point detector tally by defining a fictitious sphere surrounding the point of estimation, where it is assumed the scalar flux is constant and equal to its average value in the sphere. The average flux in this region is then used to estimate the uncollided flux at the point detector, which is known analytically. Defining this sphere requires some experience and experimentation, especially for energy-dependent problems. This method introduces a bias in the tally due to the constant flux approximation in the sphere. Another procedure [8] is to score $e^{-\Sigma_t r}/4\pi r_0^2$ for each collision inside a bounding sphere of radius r_0. The value of the constant r_0 may be estimated based on the expected approximate flux shape inside the bounding sphere. Lux and Koblinger [9] provide a comprehensive review of the point detector estimator in their book.

MCNP5 modifies the surface crossing flux estimator (F2 estimator) by tallying the expected value of $1/|\mu|$ for μ in the range $0 \le |\mu| \le \varepsilon$, where typically $\varepsilon = 0.1$. This method was originally proposed by Clark [10]. This approximate estimator yields a finite variance, and the Central Limit Theorem can be applied. However, this technique introduces a bias into the tally.

This chapter discusses new methods to estimate both flux at a point and the surface crossing flux with KDE [11, 12]. The first part of the chapter discusses the direct application of the KDE tracklength and collision estimators to estimate flux at a point, using the fact that KDE provides a functional estimate of the flux that can be evaluated at

arbitrary points in the problem geometry. The next part of this chapter presents an alternative approach where KDE is incorporated into variance reduction methods to estimate both the point detector flux and the surface crossing flux, without making any approximations near the singularities. Both of these methods yield an unbiased (at least asymptotically) estimate of the mean and a bounded variance. The reason the variance reduction approach works is that KDE provides estimates of the underlying pdf's for the distribution (in r) of collisions and the distribution (in μ) of surface crossing angles and these approximate pdf's enable the variance reduction game to be played. To estimate the point detector scalar flux or the surface crossing flux, samples are drawn from altered (e.g., uniform) distributions (in r or μ) and the scores are modified using the corresponding KDE-estimated pdf's. Both techniques of estimating Monte Carlo tallies with unbounded variance are illustrated with numerical examples.

4.2 Point Detector Estimator

The point detector estimator is a powerful but easily abused estimator for Monte Carlo particle transport. Fundamentally, it uses the point source kernel, which is the Green's function for a point source in an infinite medium, to estimate the scalar flux at a point due to an emission source, consisting of the known external source and a collision source that has itself been generated by the Monte Carlo simulation. The known source and the estimated collision source are used as emission sites for fictitious particles that contribute to the uncollided scalar flux at a point $\underline{r}$. This idea, combining an analytical solution with a Monte Carlo estimate of some intermediate quantity, in this case the collision source, to then estimate another quantity of interest, in this case the scalar flux, may be applicable to other fields in addition to particle transport.

The point detector estimator may be derived in two different ways. The first one is more intuitive and begins with an alternative, but fundamental, definition of scalar flux. The other is a formal mathematical derivation (actually, a correspondence) that makes use of the point source kernel (uncollided Green's function).

4.2.1 Physical Derivation of Point Detector Estimator

The scalar flux $\phi(\underline{r})$ is defined by the distance traveled by all the particles in a unit volume at $\underline{r}$ in a unit time. An alternative definition of scalar flux $\phi(\underline{r})$ at a point $\underline{r}$ is given by

$$\phi(\underline{r}) = \lim_{\varepsilon \to 0} \frac{N_\varepsilon}{\pi \varepsilon^2}, \tag{4.1}$$

where N_ε is the total number of neutrons/second entering a sphere of radius ε centered at $\underline{r}$, schematically illustrated by Figure 4.1. One can view the scalar flux in Eq. (4.1) as an integration of the angular flux over all angles since particles can enter the sphere from any direction.

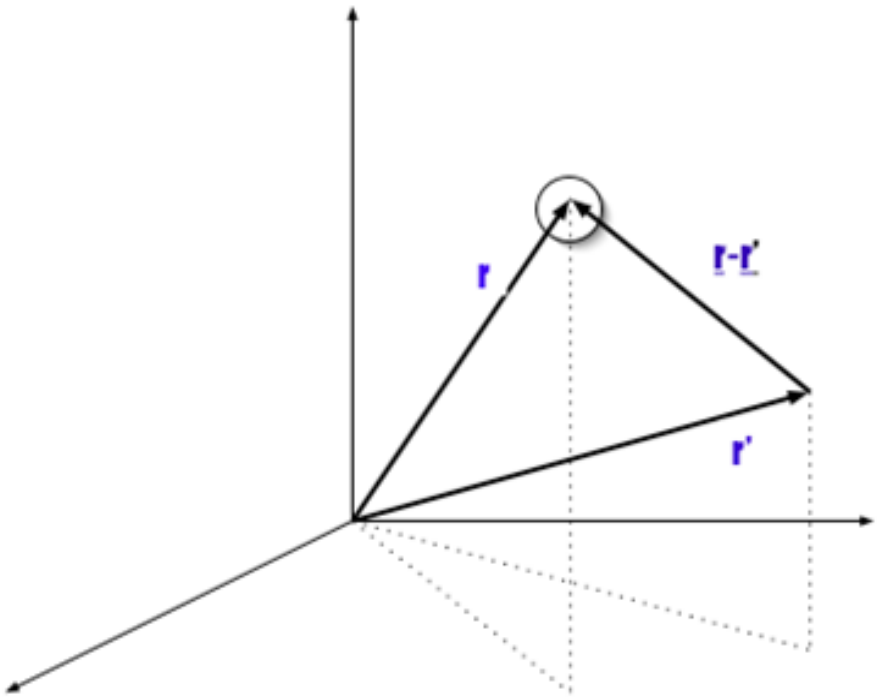

Figure 4.1 Contribution of the source at dr' about $\underline{r}'$ to the sphere of radius ε at $\underline{r}$

The idea is to compute $\phi(\underline{r})$ by adding up all the contributions from the external source and the collision source. By definition of the known source, the number of particles/second emitted in dr' about $\underline{r}'$ is $S(\underline{r}')dr'$. Assuming the source is isotropic, the differential number δN_S of these emitted particles that are emitted in the direction

defined by the sphere of radius ε centered at $\underline{r}$ (Figure 4.1) and make it to the sphere without a collision is given by

$$\delta N_S = \left[S(\underline{r}')dr'\right]\left[\frac{\pi\varepsilon^2}{4\pi\left|\underline{r}-\underline{r}'\right|^2}\right]\left[e^{-\alpha(\underline{r},\underline{r}')}\right]. \tag{4.2}$$

This equation can be expressed in words,

$$\begin{aligned}\delta \mathrm{N_s} = &\left[\text{Number of particles/sec emitted in } dr'\right]\times \\ &\left[\text{Probability of emission into the solid angle subtended by the } \varepsilon \text{ sphere}\right]\times \\ &\left[\text{Probability of getting from } \underline{r}' \text{ to } \underline{r} \text{ without a collision}\right],\end{aligned} \tag{4.3}$$

where $\alpha(\underline{r},\underline{r}')$ is the optical thickness or the number of mean free paths between $\underline{r}$ and $\underline{r}'$,

$$\alpha(\underline{r},\underline{r}') = \int_0^{|\underline{r}-\underline{r}'|} \Sigma_t\left(\underline{r}-s\frac{\underline{r}-\underline{r}'}{\left|\underline{r}-\underline{r}'\right|}\right)ds. \tag{4.4}$$

If the medium is uniform with total cross section Σ_t, then $\alpha(\underline{r},\underline{r}') = \Sigma_t\left|\underline{r}-\underline{r}'\right|$. Now consider the collision source. The collision rate at $\underline{r}'$ is given by $\Sigma_t(\underline{r}')\phi(\underline{r}')$ collisions/cm^3-s. Then the source of scattered neutrons is simply this rate times the probability of scattering at $\underline{r}'$. Treating this like the known source above, the rate at which neutrons collide in dr' and scatter into the direction subtended by the ε sphere, and reach the sphere without collision is given by:

$$\delta N_C = \left[\Sigma_t(\underline{r}')\phi(\underline{r}')dr'\right]\left[\frac{\Sigma_s(\underline{r}')}{\Sigma_t(\underline{r}')}\right]\left[\frac{\pi\varepsilon^2}{4\pi\left|\underline{r}-\underline{r}'\right|^2}\right]\left[e^{-\alpha(\underline{r},\underline{r}')}\right]. \tag{4.5}$$

In this case, the qualitative description becomes

$$\begin{aligned}\delta N_c = &\left[\text{Number of collisions/sec in } dr'\right]\times \\ &\left[\text{Probability that the collision is a scatter}\right]\times \\ &\left[\text{Probability the neutron is scattered into the solid angle subtended by the } \varepsilon \text{ sphere}\right]\times \\ &\left[\text{Probability of getting from } \underline{r}' \text{ to } \underline{r} \text{ without a collision}\right].\end{aligned} \tag{4.6}$$

Then the total number of neutrons/s which make it to the ε sphere is given by the sum of the two terms in Eqs. (4.2) and (4.5):

$$\delta N_{\varepsilon} = \delta N_S + \delta N_C. \tag{4.7}$$

Now associate these terms with the Monte Carlo random walk process and identify the term $S(\underline{r}')dr'$ as the particle emitted by the source and $\Sigma_t(\underline{r}')\phi(\underline{r}')dr'$ as a particle that suffers a collision. To get the total number, sum over all the source particles $i = 1, 2, ..., N$ and all subsequent collisions to obtain the total number of particles per second which end up traveling towards the ε sphere:

$$N_{\varepsilon} = \sum_{i=1}^{N} \left[e^{-\alpha\left(\underline{r},\underline{r}_i\right)} \frac{\pi\varepsilon^2}{4\pi\left|\underline{r}-\underline{r}_i\right|^2} + \sum_{c=1}^{C_i} \left(\frac{\Sigma_s(\underline{r}_{i,c})}{\Sigma_t(\underline{r}_{i,c})} \right) e^{-\alpha(\underline{r},\underline{r}_{i,c})} \frac{\pi\varepsilon^2}{4\pi\left|\underline{r}-\underline{r}_{i,c}\right|^2} \right], \tag{4.8}$$

where C_i is the number of collisions during history i. Then, using the definition of scalar flux from Eq. (4.1), we find

$$\phi(\underline{r}) = \sum_{i=1}^{N} \left[\frac{e^{-\alpha(\underline{r},\underline{r}_i)}}{4\pi\left|\underline{r}-\underline{r}_i\right|^2} + \sum_{c=1}^{C_i} \left(\frac{\Sigma_s(\underline{r}_{i,c})}{\Sigma_t(\underline{r}_{i,c})} \right) \frac{e^{-\alpha(\underline{r},\underline{r}_{i,c})}}{4\pi\left|\underline{r}-\underline{r}_{i,c}\right|^2} \right], \tag{4.9}$$

which is the desired scalar flux at $\underline{r}$.

4.2.2 Mathematical Derivation of Point Detector Estimator

This derivation makes use of the fact that the first flight kernel, or uncollided Green's function, due to a point source in an infinite homogeneous medium, is given by the following:

$$G_{pt}(\underline{r},\underline{r}') = \frac{e^{-\alpha(\underline{r},\underline{r}')}}{4\pi\left|\underline{r}-\underline{r}'\right|^2}. \tag{4.10}$$

Using Eq. (4.10) and the definition of the Green's function, by superposition one can obtain the scalar flux due to the known source term and the resultant emission source due to collisions:

$$\phi(\underline{r}) = \int_V \frac{e^{-\alpha(\underline{r},\underline{r}')}}{4\pi\left|\underline{r}-\underline{r}'\right|^2} S(\underline{r}')dr' + \int_V \frac{e^{-\alpha(\underline{r},\underline{r}')}}{4\pi\left|\underline{r}-\underline{r}'\right|^2} \Sigma_s(\underline{r}')\phi(\underline{r}')dr'. \tag{4.11}$$

This is just the integral transport equation for isotropic scattering, which should be no surprise. This equation is exact for isotropic scattering. Now note that the Monte Carlo simulation yields source particles that estimate the term $S(\underline{r}')dr'$ and collisions that estimate the term $\Sigma_t(\underline{r}')\phi(\underline{r}')dr'$. Therefore the integrals over volume become summations over source and collision sites:

$$\phi(\underline{r}) = \sum_{i=1}^{N}\left[\frac{e^{-\alpha(\underline{r},\underline{r_i})}}{4\pi\left|\underline{r}-\underline{r_i}\right|^2} + \sum_{c=1}^{C_i}\left(\frac{\Sigma_s(\underline{r_{i,c}})}{\Sigma_t(\underline{r_{i,c}})}\right)\frac{e^{-\alpha(\underline{r},\underline{r_{i,c}})}}{4\pi\left|\underline{r}-\underline{r_{i,c}}\right|^2}\right], \tag{4.12}$$

where $\underline{r_{i,c}}$ = location of the c^{th} collision for the i^{th} particle and $\underline{r_i}$ = location of the source emission in history i. Eq. (4.12) indicates that for each history $i = 1, 2, ..., N$, an estimate ϕ_i of the scalar flux $\phi(\underline{r})$ due to the source "emissions" and collision "emissions" can be found:

$$\phi_i = \frac{e^{-\alpha(\underline{r},\underline{r_i})}}{4\pi\left|\underline{r}-\underline{r_i}\right|^2} + \sum_{c=1}^{C_i}\left(\frac{\Sigma_s(\underline{r_{i,c}})}{\Sigma_t(\underline{r_{i,c}})}\right)\frac{e^{-\alpha(\underline{r},\underline{r_{i,c}})}}{4\pi\left|\underline{r}-\underline{r_{i,c}}\right|^2}, \tag{4.13}$$

where C_i is the number of collisions during history i, and "emissions" is in quotes to signify that these are fictitious emissions that are only used to estimate the point flux $\phi(\underline{r})$ and are not followed in the random walk simulation. Every source sample and every collision launches a fictitious particle that results in a score. If a variance reduction game is being played, such as implicit capture or Russian Roulette or splitting [13], then the instantaneous weight $w_{i,c}$ of the particle i at collision c would be included in the partial score for ϕ_i.

4.2.3 Estimating the Uncertainty in the Point Scalar Flux Estimate

To obtain the uncertainty in our estimates, compute the sample mean and variance by first accumulating the sum and the sum of the squares of the observations ϕ_i:

$$\Phi = \sum_{i=1}^{N}\phi_i, \tag{4.14}$$

$$\Phi_{SQ} = \sum_{i=1}^{N} \phi_i^2, \tag{4.15}$$

where one must be careful not to compute ϕ_i^2 until the entire history i is completed, including all progeny from non-analog games such as splitting, forced collisions, and implicit capture. Otherwise the ϕ_i's will not be independent, identically distributed observations.

The scalar flux is then estimated by the sample mean, which is the average of the N observations:

$$\hat{\phi} = \frac{\Phi}{N} = \frac{1}{N} \sum_{i=1}^{N} \phi_i. \tag{4.16}$$

The sample variance for the estimate of the mean is

$$s_N^2 \equiv \frac{1}{N} \frac{1}{N-1} \sum_{i=1}^{N} \left(\phi_i - \hat{\phi} \right)^2 = \frac{1}{N-1} \left[\frac{1}{N} \sum_{i=1}^{N} \phi_i^2 - \hat{\phi}^2 \right] = \frac{1}{N-1} \left[\langle \phi^2 \rangle - \langle \phi \rangle^2 \right], \tag{4.17}$$

which is the familiar "average of the square – square of the average" divided by $N-1$. Using a one-sigma error criterion, Eq. (4.17) results in the following uncertainty in the estimate of the scalar flux:

$$\Delta\hat{\phi} = s_N = \frac{1}{\sqrt{N-1}} \sqrt{\langle \phi^2 \rangle - \langle \phi \rangle^2}. \tag{4.18}$$

This yields the following estimate of the scalar flux at $\underline{r}$ using the point source estimator:

$$\phi(\underline{r}) \approx \hat{\phi} \pm \Delta\hat{\phi}, \tag{4.19}$$

where $\hat{\phi}$ is given by Eq. (4.16)and $\Delta\hat{\phi}$ is given by Eq. (4.18).

4.2.4 Understanding the Point Detector Problem with an Example

If the point detector is within a region that scatters or emits particles due to a source term, the $1/\left|\underline{r} - \underline{r}'\right|^2$ dependence of the point detector estimator will result in a theoretical variance which is infinite. It is shown in this section that the mean is finite but the variance is infinite. This implies that the Central Limit Theorem [14] is not valid and one

cannot use estimates of the variance to deduce convergence to the true mean. It may be the case that sample variances will be finite for finite numbers of histories, but eventually collision or source point will be sampled arbitrarily close to the detector point and the estimate of the variance will diverge.

While the point detector estimator is an unbiased estimator for the scalar flux at a point, the infinite variance does limit its usefulness in regions where collisions may occur. To illustrate this phenomenon, let us examine the behavior of the point detector estimator for the case of a purely absorbing sphere of radius R with constant cross section Σ_a and a uniform volumetric source of neutrons emitting $S_0 = 1$ neutron/s. The goal is to estimate the scalar flux $\phi_0 \equiv \phi(0)$ at the center of the sphere. The analytical solution to this problem is easily obtained from Eq. (4.11):

$$\phi(\underline{r}) = \int_V S(\underline{r}') \frac{e^{-\alpha(\underline{r},\underline{r}')}}{4\pi \left|\underline{r} - \underline{r}'\right|^2} dr' = \int_0^R \left(\frac{3}{4\pi R^3}\right)\left(\frac{e^{-\Sigma_a r}}{4\pi r^2}\right) 4\pi r^2 dr = \left(\frac{3}{4\pi R^3}\right)\left(\frac{1}{\Sigma_a}\right)\left(1 - e^{-\Sigma_a R}\right). \quad (4.20)$$

This is the exact solution to the problem and is the true mean for the experiment consisting of sampling a neutron uniformly within the sphere and estimating the scalar flux at $r = 0$:

$$\bar{\phi}_0 = \left(\frac{3}{4\pi R^3}\right)\left(\frac{1}{\Sigma_a}\right)\left(1 - e^{-\Sigma_a R}\right). \quad (4.21)$$

The Monte Carlo simulation will consist of N histories, and each history yields an independent estimate of ϕ_0. Since the neutrons are sampled uniformly within the sphere, the pdf for the radial position of the sampled neutron is given by

$$f(r) = \frac{3r^2}{R^3}. \quad (4.22)$$

From Eq. (4.13), the point detector estimator will score the quantity $\phi_{pt}(r) = \dfrac{e^{-\Sigma_a r}}{4\pi r^2}$ for every source neutron that is sampled. Since there are no further collisions for this pure absorber, this is the only contribution to the point detector score. Therefore, the mean point detector score is given by:

$$\overline{\phi_0} = \int_0^R f(r)\phi_{pt}(r)dr = \int_0^R \frac{3r^2}{R^3}\left(\frac{e^{-\Sigma_a r}}{4\pi r^2}\right)dr = \left(\frac{3}{4\pi R^3}\right)\left(\frac{1}{\Sigma_a}\right)\left(1-e^{-\Sigma_a R}\right), \tag{4.23}$$

which is identical to the analytical solution given in Eq. (4.20), confirming that the point detector estimator is unbiased. The variance is given by

$$\sigma^2 = \overline{\phi_0^2} - \overline{\phi_0}^2 = \int_0^R f(r)\left(\frac{e^{-\Sigma_a r}}{4\pi r^2}\right)^2 dr - \overline{\phi_0}^2. \tag{4.24}$$

But the integral is unbounded, hence $\sigma^2 = \infty$. For this simple case, the point detector estimator is unbiased but has an infinite variance. The general case with scattering is likely to exhibit behavior similar to this idealized case.

4.3 Estimation of Scalar Flux at a Point by KDE tallies

The mesh-free KDE tallies, which are introduced in Chapter 3, are capable of estimating flux at a point by simply evaluating the KDE tally expressions at the detector point and the convergence within a scattering region was shown to be substantially better than the conventional point detector estimator. Moreover, in Chapter 2 it was found that the variance of the KDE tally is bounded and an asymptotic variance bound was derived.

4.3.1 KDE Collision and Track Length Tallies

Three dimensional KDE Monte Carlo collision and track length tallies were derived in Chapter 3. The KDE tallies are directly employed in this chapter to evaluate flux at a point. In this section the equations for 3-D KDE collision and track length tallies and their variances will be reviewed with their asymptotic variance bound. The 3-D kernel density estimated collision and track length tallies and their variances are given by

$$\phi(x,y,z) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c},Y_{i,c},Z_{i,c})}\frac{1}{h_x}k\left(\frac{x-X_{i,c}}{h_x}\right)\frac{1}{h_y}k\left(\frac{y-Y_{i,c}}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_{i,c}}{h_z}\right), \tag{4.25}$$

$$\sigma^2_{\phi(x,y,z)} = \frac{1}{(N-1)}\left[\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{c=1}^{c_i}\frac{w_{i,c}}{\Sigma_t(X_{i,c},Y_{i,c},Z_{i,c})}\right.\right.$$
$$\left.\left.\frac{1}{h_x}k\left(\frac{x-X_{i,c}}{h_x}\right)\frac{1}{h_y}k\left(\frac{y-Y_{i,c}}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_{i,c}}{h_z}\right)\right\}^2\right. \tag{4.26}$$
$$\left.-\phi(x,y,z)^2\right],$$

$$\phi(x,y,z) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}d_{i,c}}{n}\sum_{j=1}^{n}\frac{1}{h_x}k\left(\frac{x-X_{i,c,j}}{h_x}\right)$$
$$\frac{1}{h_y}k\left(\frac{y-Y_{i,c,j}}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_{i,c,j}}{h_z}\right), \tag{4.27}$$

$$\sigma^2_{\phi(x,y,z)} = \frac{1}{N-1}\left[\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{c=1}^{c_i}\frac{w_{i,c}d_{i,c}}{n}\sum_{j=1}^{n}\frac{1}{h_x}k\left(\frac{x-X_{i,c,j}}{h_x}\right)\right.\right.$$
$$\left.\left.\frac{1}{h_y}k\left(\frac{y-Y_{i,c,j}}{h_y}\right)\frac{1}{h_z}k\left(\frac{z-Z_{i,c,j}}{h_z}\right)\right\}^2-\left(\phi(x,y,z)\right)^2\right]. \tag{4.28}$$

All the notations in the above equations have their usual meaning. The Epanechnikov kernel is used for all our simulations and the bandwidth is calculated by using Eq. (2.59). The asymptotic variance bound of KDE is derived in chapter 2 (Eq. (2.55)), which is

$$\begin{aligned}\operatorname{var}\hat{f}(x_1,...,x_d) &\approx \frac{1}{Nh^d}f(x_1,...,x_d)\beta \\ &\approx \frac{1}{Nh^d}.\end{aligned} \tag{4.29}$$

4.3.2 Numerical Example

Three concentric spheres with radii 0.1, 0.5 and 1.0 cm respectively are used to demonstrate the ability of KDE flux tally to estimate flux at a point, that may be within a scattering region. The problem geometry is illustrated in Figure 4.2. For simplicity the detector point is placed at the center of the sphere. One of the reasons for selecting this simple problem is that it is easy to calculate the analytical flux at the center of the spheres if the three regions contain purely absorbing materials. A uniform isotropic source and isotropic scattering are assumed for this problem.

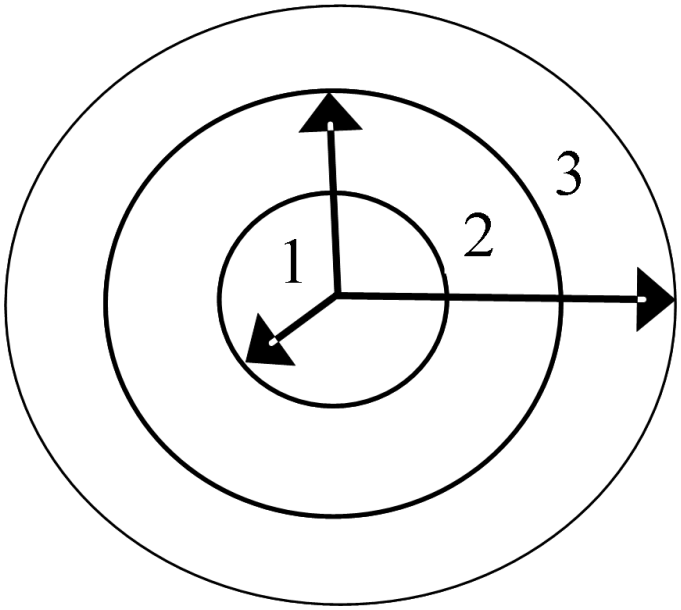

Figure 4.2 Three concentric spheres with uniform volumetric source.

Table 4.1 Description of all the numerical cases

Case	Region 1	Region 2	Region 3	# histories
1	$\Sigma_a = 1;\ \Sigma_s = 0$	$\Sigma_a = 1;\ \Sigma_s = 0$	$\Sigma_a = 1;\ \Sigma_s = 0$	15,000
2	$\Sigma_a = 2;\ \Sigma_s = 0$	$\Sigma_a = 3;\ \Sigma_s = 0$	$\Sigma_a = 4;\ \Sigma_s = 0$	30,000
3	$\Sigma_a = 0;\ \Sigma_s = 0$	$\Sigma_a = 1;\ \Sigma_s = 0$	$\Sigma_a = 1;\ \Sigma_s = 0$	15,000
4	$\Sigma_a = 0;\ \Sigma_s = 0$	$\Sigma_a = 0;\ \Sigma_s = 0$	$\Sigma_a = 1;\ \Sigma_s = 0$	30,000
5	$\Sigma_a = 0;\ \Sigma_s = 5$	$\Sigma_a = 0;\ \Sigma_s = 4$	$\Sigma_a = 1;\ \Sigma_s = 4$	10^6

Table 4.2 Results for cases 1 and 2

Estimators	Case 1	Case 2
Exact	0.15091	0.06982
Conventional Point Detector	0.14374 ± 0.00580	0.07734 ± 0.00527
KDE Collision	0.15188 ± 0.01085	0.07030 ± 0.00413
KDE Track *(n=3)*	0.15129 ± .00918	0.06998 ± 0.00447

Table 4.3 Results for cases 3 and 4

Estimators	Case 3	Case 4
Exact	0.14181	0.10735
Conventional Point Detector	0.14051 ± 0.00273	0.10765 ± 0.00036

KDE Collision	0.13754 ± 0.01008	0.0 ± 0.0
KDE Track *(n=4)*	0.14246 ± 0.00938	0.10714 ± 0.00655

Cases 1-4 are different purely absorbing (and void) configurations while case 5 includes scattering. Specific details of each case are documented in Table 4.1. Cases 1-4 compare the conventional point detector and KDE tallies with the exact solution. The results are tabulated in Tables 4.2 and 4.3. The exact solutions are obtained by solving the first integral of Eq. (4.11). Cases 3 and 4 are used to demonstrate that the KDE track length estimator compares well with the point detector estimator even when the detector is placed in a void region, a configuration that is ideal for the conventional point detector estimator.

Highly scattering materials (case 5) are also used to fill the three spherical shells. A sphere of radius 0.01 cm surrounding the detector at the center is used with 10^9 particles and the conventional track-length estimator is employed to obtain an estimate of the flux. This flux is used as a benchmark solution for comparison with the point detector and KDE flux estimators. Results are tabulated in Table 4.4.

Table 4.4 Results for case 5

Estimators	**Case 5**
Benchmark	0.42650±0.00122
Conventional Point Detector	0.41623± 0.00291
KDE Collision	0.41978± 0.00285
KDE Track *(n=3)*	0.42436± .00228

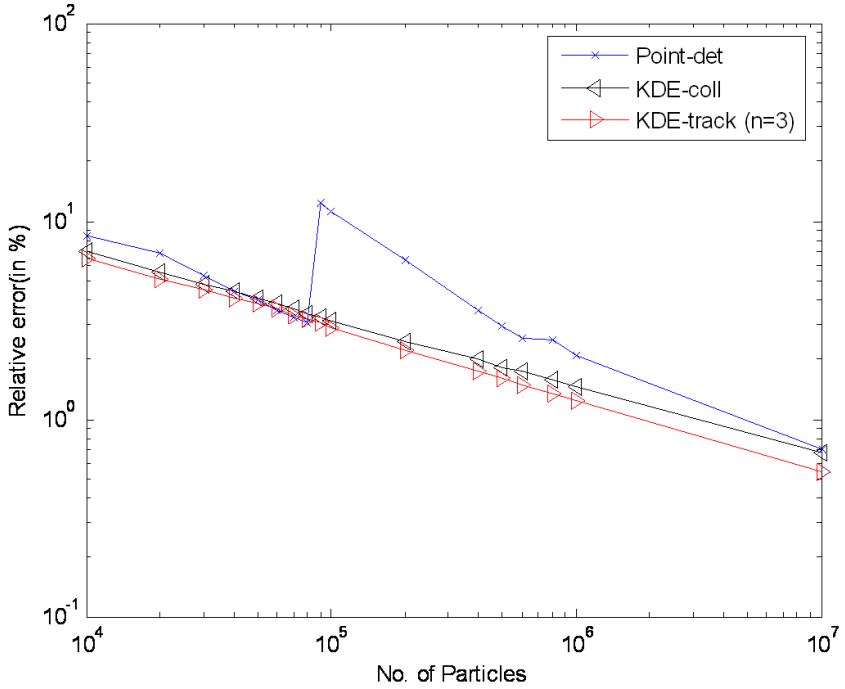

Figure 4.3 Relative error vs. number of particles for case 5.

It is evident from all the above cases that KDE tallies are very effective for estimating flux at a point, especially the KDE track length tally. The relative error for case 5 as a function of number of particles is plotted in Figure 2. The estimated relative error is larger and its behavior is more irregular for the conventional point detector case than the KDE estimators, a consequence of the infinite variance for the point detector estimator. The variances of the KDE estimators converge as

$$\begin{aligned} \operatorname{var}\hat{f}(x_1,.....,x_d) &\approx \frac{1}{Nh^d} \\ &= \frac{1}{Nh^3} \\ &\approx \frac{1}{N\left(N^{-1/7}\right)^3} \\ &= \frac{1}{N^{4/7}}. \end{aligned} \tag{4.30}$$

The bandwidth in Eq. (4.30) is replaced by the number of observations from Eq. (2.59). The Figure 4.3 is confirming the convergence rate obtained by Eq. (4.30). Note that, as the kernel function has a finite support (for Epanechnikov kernel $\left[-\sqrt{5},\sqrt{5}\right]$), only

collision points in the neighborhood of the detector points are required to estimate the flux. Global bandwidth can be calculated "online" without storing all the collision points. Local bandwidth, calculated based on the collision points in the vicinity of the detector point, can also be employed. We used global bandwidth for the above numerical example. Boundary correction is only required for this application of KDE tallies for estimating flux at a point if the detector point is very close to the external boundary of the system. In Chapter 3, we have seen KDE tallies perform well near the internal material boundaries without any boundary correction.

The result in Eq. (4.30) should not be compared with the usual 1/N convergence for variance of an estimator because this is a pointwise evaluation of the actual functional estimate of the flux. If one were to average this variance over a finite region, the resultant convergence will be 1/N.

4.4 Alternative KDE approach for Point Detector Estimator

In the previous chapter (Chapter 3), it has been shown that KDE can be used to estimate the scalar flux using modified collision and pathlength estimators, yielding continuous representations of the scalar flux without the need to impose a mesh. In the previous sections of this chapter, these mesh-free KDE tallies were shown to be capable of estimating flux at a point by simply evaluating the KDE tally expressions at the detector point and the convergence within a scattering region was shown to be substantially better than the conventional point detector estimator. However, the convergence rate for the variance of the pointwise evaluation of the KDE flux estimator was less than 1/N, so it did not compare well with the convergence of the variance for the conventional point detector estimator, which is 1/N for point detectors outside a scattering or source region. The desire to have a faster converging point estimate of the flux motivated this study.

The rest of this chapter is devoted to an alternative approach in which KDE is combined with a variance reduction method to estimate the point detector flux and also the surface crossing flux, yielding an unbiased estimate of the mean and a bounded variance for both estimators. The important observation that enables our approach: KDE yields an implicit (functional) estimate of the probability density function (pdf) that governs each estimator.

This can be used in a variance reduction game by sampling from an altered pdf (with an associated weight that depends on the estimated pdf) that preserves the mean but yields a bounded variance.

4.4.1 Methodology for Alternative Approach

The idea is to apply the concept of importance sampling [1], which is used as a variance reduction tool. Let us consider **x** is a random variable with probability density function (pdf) $f(x)$. Let $g(x)$ be a function of the random variable **x**. By definition, the expected value and the variance of $g(x)$ are given by

$$E\left[g(x)\right] = \mu = \int g(x) f(x) d\ ,x \tag{4.31}$$

$$\text{var}\, g(x) = \sigma^2 = \int g(x)^2 f(x) d\ \text{ ⁄} \mu^2. \tag{4.32}$$

In Monte Carlo μ can be estimated by drawing samples from $f(x)$ and scoring $g(x)$. If $X_1, ..., X_N$ are independent and identically distributed observations from $f(x)$

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} g(X_i), \tag{4.33}$$

$$\sigma_{\hat{\mu}}^2 = \frac{1}{N-1}\left\{\frac{1}{N}\sum_{i=1}^{N} g(X_i)^2 - \hat{\mu}^2\right\}, \tag{4.34}$$

where $\hat{\mu}$ is the estimator for μ and $\sigma_{\hat{\mu}}^2$ is the variance of $\hat{\mu}$. We do not change the expected value in Eq. (4.31) by multiplying the integrand by unity, so we can re-cast Eq. (4.31) as

$$\mu = \int g(x)\frac{f^*(x)}{f^*(x)} f(x)\, dx = \int \left[g(x)\frac{f(x)}{f^*(x)} \right] f^*(x)\, dx, \tag{4.35}$$

where $f^*(x)$ is a pdf, defined in the same domain as $f(x)$. μ can also be estimated by drawing samples from $f^*(x)$ and scoring the function $\left[g(x)\frac{f(x)}{f^*(x)} \right]$. Therefore, we can estimate the expected value of $g(x)$ by sampling from any suitable pdf, f^*. Although

this process does not affect the expected value, which is evident from Eq. (4.35), it does affect the variance. For the altered scheme, the variance is given by

$$\sigma^2 = \int \left[g(x) \frac{f(x)}{f^*(x)} \right]^2 f^*(x)\, dx - \mu^2, \tag{4.36}$$

which is not same as the variance given by (4.32).

This importance sampling concept is applied to the point detector estimator. First, the point detector equation is reviewed. The uncollided point flux at a point $(\underline{r})$ due to a point source at $\underline{r}'$ is given by

$$\phi_{pt}(\underline{r}, \underline{r}') = \frac{e^{-\tau(\underline{r}, \underline{r}')}}{4\pi |\underline{r} - \underline{r}'|^2}, \tag{4.37}$$

where $\tau(\underline{r}, \underline{r}')$ is the optical thickness between $\underline{r}$ and $\underline{r}'$. Using Eq. (4.37), the scalar flux at $\underline{r}$ due to a known source term $S(\underline{r}')$ and resultant emission source due to scatters can be obtained:

$$\begin{aligned} \phi(\underline{r}) &= \int_V \frac{e^{-\tau(\underline{r}, \underline{r}')}}{4\pi |\underline{r} - \underline{r}'|^2} S(\underline{r}')\, dV + \int_v \frac{e^{-\tau(\underline{r}, \underline{r}')}}{4\pi |\underline{r} - \underline{r}'|^2} \Sigma_s(\underline{r}') \phi(\underline{r}')\, dV \\ &= I_1 + I_2. \end{aligned} \tag{4.38}$$

As discussed before, this is nothing but the integral transport equation for isotropic scattering. The conventional Monte Carlo point detector estimator generates realizations from the source distribution $S(\underline{r}')dV$ and scores $\phi_{pt}(\underline{r}, \underline{r}')$. If there is a collision at $\underline{r}'$, where we note that $\underline{r}'$ is distributed according to $f(\underline{r}')dV' \sim \Sigma_t(\underline{r}')\phi(\underline{r}')dV'$, then $\left(\Sigma_s(\underline{r}')/\Sigma_t(\underline{r}')\right)\phi_{pt}(\underline{r}, \underline{r}')$ is scored. We take advantage of the fact that KDE provides a functional estimate of the pdf governs the distribution of collisions. Assuming that the point detector is located at the origin, Equation (4.38) can be written as

$$
\begin{aligned}
\phi\left(\underline{0}\right) &= \int_0^R \frac{e^{-\tau\left(\underline{0},\underline{r}'\right)}}{4\pi r'^2} S\left(r'\right) 4\pi r'^2 dr' + \int_0^R \frac{e^{-\tau\left(\underline{0},\underline{r}'\right)}}{4\pi r'^2} \Sigma_s\left(r'\right)\phi\left(r'\right) 4\pi r'^2 dr' \\
&= \int_o^R \left[\frac{e^{-\tau\left(\underline{0},\underline{r}'\right)}}{4\pi r'^2} \frac{S\left(r'\right) 4\pi r'^2}{\tilde{f}_1\left(r'\right)} \right] \tilde{f}_1\left(r'\right) dr' + \int_0^P \left[\frac{e^{-\tau\left(\underline{0},\underline{r}'\right)}}{4\pi r'^2} \frac{\Sigma_s\left(r'\right)\phi\left(r'\right) 4\pi r'^2}{\tilde{f}_2\left(r'\right)} \right] \tilde{f}_2\left(r'\right) dr' \\
&\qquad + \int_P^R \frac{e^{-\tau\left(\underline{0},\underline{r}'\right)}}{4\pi r'^2} \Sigma_s\left(r'\right)\phi\left(r'\right) 4\pi r'^2 dr' \\
&= I_1 + I_{21} + I_{22}
\end{aligned}
\tag{4.39}
$$

For the first integral $\left(I_1\right)$, instead of generating samples from $S\left(r'\right)4\pi r'^2 dr'$ and scoring the function $\phi_{pt}\left(\underline{0},\underline{r}'\right)$, the samples are drawn from the altered pdf $\tilde{f}_1\left(r'\right)$, where $0 < r' < R$, and then those samples are used to score the entire function within the square bracket. This approach is independent of KDE and could be done for conventional Monte Carlo tallies. It works because the distribution of source particles, $S\left(r'\right)4\pi r'^2 dr'$, is known. This is of course not known for particles emitted as a result of collisions, but KDE yields an estimate of this distribution and this can be used in a similar fashion as the source distribution. Specifically, the second integral $\left(I_2\right)$ is broken into two integrals (I_{21} and I_{22}). For the integral containing the singularity (I_{21}), instead of scoring the point kernel $\phi_{pt}\left(\underline{0},\underline{r}'\right)$ at every collision event within P (P is an arbitrary point between 0 and R), the collision points are stored. Then at the end of the simulation Monte Carlo scores the entire function within the square bracket by generating samples from the altered pdf $\tilde{f}_2\left(r'\right)$, where $0 < r' < P$. An estimate of $\phi\left(r'\right)4\pi r'^2$ is obtained by using Eq. (3.21), which is the equation for the KDE collision flux tally, for all the stored collision points within radius P surrounding the point detector. The non-singular portion of the integral, I_{22}, is estimated using conventional methods. This method of altering the sampling distribution yields a bounded variance by removing the $1/r^2$ singularity from the Monte Carlo scoring function. The distance P is a user-defined parameter, like the radius of the fictitious sphere for the MCNP5 F5 tally, and can be used as a variance reduction tool.

In this case, the first derivative of the unknown function ($f(r) = 4\pi r^2 \phi(r)$) at 0 (detector point) is 0. This means that a reflection boundary correction should be sufficient for the points close to the detector. However, it has been found that the reflecting boundary correction very close to the detector point ($r \to 0$) yields nearly constant estimations of the unknown distribution, which converges to 0 with $r \to 0$ very slowly. Therefore, a simple quadratic least squares fit is used as a boundary correction for KDE for points close to the detector point. The quadratic fit is used, because the shape of the unknown function $f(r)$ near $r \to 0$ is dominated by the r^2 factor. The fit is given by

$$y = ar^2. \tag{4.40}$$

First, y 's are calculated at r 's (for our numerical case 70 equally spaced points on r axis at an interval 0.001, starting from $r = 0.01$ are used) by using usual KDE estimator with reflection boundary correction. Those y 's and r 's are then used to calculate the least square coefficient a. For all $r < 0.01$, the least square fit is used as the boundary correction. For $r \geq 0.01$, usual KDE estimator with reflection boundary correction (if required depending on the bandwidth) is used. For points close to P boundary kernels are used to take care of the boundary bias [15].

4.4.2 MCNP5 Methodology for Point Detector Estimator

MCNP5 point detector methodology is introduced in this section for the purpose of comparing the MCNP5 and KDE point detector tallies. MCNP5 uses a specified average flux region close to the detector to remove the singularity from the estimator. This region is defined by a fictitious sphere of radius R_0 surrounding the point detector. Any collisions that occur outside this sphere are scored in the conventional way. For collisions inside the sphere, the factor $\phi_{pt}(\underline{0}, \underline{r}')$ is scored using the assumption the collisions are uniformly distributed inside the sphere. The resultant score is therefore the analytical solution for the uncollided flux at the center of a sphere with a uniform source:

$$\frac{\int_0^{R_0} e^{-\Sigma_t r'} dr'}{\frac{4}{3}\pi R_0^3} = \frac{1-e^{-\Sigma_t R_0}}{(4/3)\pi R_0^3 \Sigma_t}. \tag{4.41}$$

Isotropic scattering is assumed. Since uniform collisions are assumed within a sphere, it is not advisable to choose a value of R_0 that encompasses more than one material. This constraint is not an issue for the KDE point detector tally.

4.4.3 Adjoint Calculation for Estimating Flux at a Point

The adjoint calculation is used as a reference solution for the numerical examples. For most of the cases this is not the method of choice, because if there are several point detectors, the adjoint game would need to be repeated for each detector. Furthermore, the adjoint calculation is not possible at the current time for continuous energy Monte Carlo. Finally, if point sources are present, this method leads to the same kind of singularity as in the forward calculation.

Let $\psi(\underline{x},\underline{\Omega},E)$ satisfy the forward transport problem:

$$\begin{aligned} L\psi &= Q, \quad && \underline{x}\in V,\ \underline{\Omega}\in 4\pi,\ 0<E<\infty, \\ \psi &= \psi^b, && \underline{x}\in \partial V,\ \underline{\Omega}.\underline{n}<0,\ 0<E<\infty, \end{aligned} \tag{4.42}$$

and let $\psi^*(\underline{x},\underline{\Omega},E)$ satisfy the adjoint transport problem:

$$\begin{aligned} L^*\psi^* &= Q^*, \quad && \underline{x}\in V,\ \underline{\Omega}\in 4\pi,\ 0<E<\infty, \\ \psi^* &= \psi^{b*}, && \underline{x}\in \partial V,\ \underline{\Omega}.\underline{n}>0,\ 0<E<\infty. \end{aligned} \tag{4.43}$$

The operators L and L^* are forward and adjoint Boltzmann transport operators [16, 17] respectively. $\psi(\underline{x},\underline{\Omega},E)$ and $\psi^*(\underline{x},\underline{\Omega},E)$ are forward and adjoint angular fluxes and Q and Q^* are forward and adjoint sources. The superscript b stands for boundary. From the definition of adjoint the following identity can be derived:

$$\begin{aligned}&\left(\psi^*,Q\right)+\int_{\partial V}\int_0^\infty\int_{\underline{\Omega}.\underline{n}<0}\left|\underline{\Omega}.\underline{n}\right|\psi^*\left(\underline{x},\underline{\Omega},E\right)\psi^b\left(\underline{x},\underline{\Omega},E\right)d\Omega dEdS\\&=\left(Q^*,\psi\right)+\int_{\partial V}\int_0^\infty\int_{\underline{\Omega}.\underline{n}>0}\underline{\Omega}.\underline{n}\psi^{b*}\left(\underline{x},\underline{\Omega},E\right)\psi\left(\underline{x},\underline{\Omega},E\right)d\Omega dEdS.\end{aligned}\tag{4.44}$$

Here $(\ )$ represents inner product. The goal is to calculate the flux at the center of a sphere of radius R with a uniform, isotropic source. For this particular case the forward and adjoint problems become:

$$\begin{aligned}&L\psi=Q,\qquad \underline{x}\in V,\ \underline{\Omega}\in 4\pi,\ 0<E<\infty,\\&\psi=0,\qquad \underline{x}\in\partial V,\ \underline{\Omega}.\underline{n}<0,\ 0<E<\infty,\end{aligned}\tag{4.45}$$

$$\begin{aligned}&L^*\psi^*=\frac{1}{4\pi}\delta\left(\underline{x}-\underline{x}_0\right),\qquad \underline{x}\in V,\ \underline{\Omega}\in 4\pi,\ 0<E<\infty,\\&\psi^*=0,\qquad\qquad\qquad\ \underline{x}\in\partial V,\ \underline{\Omega}.\underline{n}>0,\ 0<E<\infty,\end{aligned}\tag{4.46}$$

where δ is the delta function [16]. For one energy group, combine Eqs. (4.44), (4.45), and (4.46) to find:

$$\int_V\int_{4\pi}\psi^*\left(\underline{x},\underline{\Omega}\right)Q\left(\underline{x},\underline{\Omega}\right)d\Omega dV=\frac{1}{4\pi}\int_{4\pi}\psi\left(\underline{x}_0,\underline{\Omega}\right)d\Omega.\tag{4.47}$$

The source Q and scalar flux for this spherical problem can be defined by

$$Q\left(\underline{x},\underline{\Omega}\right)=\frac{1}{4\pi}Q\left(\underline{x}\right)=\frac{1}{4\pi}\frac{3}{4\pi R^3},\tag{4.48}$$

$$\phi\left(\underline{x}\right)=\int_{4\pi}\psi\left(\underline{x},\underline{\Omega}\right)d\Omega.\tag{4.49}$$

Combine Eqs. (4.47), (4.48), and (4.49) to find

$$\phi\left(\underline{x}_0\right)=\int_V\phi^*\left(\underline{x}\right)Q\left(\underline{x}\right)dV.\tag{4.50}$$

Multiplying the integrand of Eq. (4.50) by unity yields:

$$\phi\left(\underline{x}_0\right)=\int_V\frac{Q\left(\underline{x}\right)}{\Sigma_t\left(\underline{x}\right)}\Sigma_t\left(\underline{x}\right)\phi^*\left(\underline{x}\right)dV.\tag{4.51}$$

Hence the Monte Carlo collision estimate of the scalar flux at location $\underline{x}_0$ is given by

$$\hat{\phi}\left(\underline{x}_0\right)=\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C_i}\frac{w_{i,c}}{\Sigma_t\left(X_{i,c}\right)}\frac{3}{4\pi R^3}. \tag{4.52}$$

All the symbols have their usual meaning and $w_{i,c}$ is the weight of the i^{th} adjoint particle before the c^{th} collision at $X_{i,c}$. In a similar fashion the adjoint track length estimator can be derived for the scalar flux at location $\underline{x}_0$:

$$\hat{\phi}\left(\underline{x}_0\right)=\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C_i}w_{i,c}d_{i,c}\frac{3}{4\pi R^3}, \tag{4.53}$$

where $d_{i,c}$ is the track length between events $c-1$ and c for the adjoint Monte Carlo particle. Expand the adjoint operator in Eq. (4.46) to find

$$\begin{aligned}&-\underline{\Omega}.\underline{\nabla}\psi^*\left(\underline{x},\underline{\Omega}\right)+\Sigma_t\left(\underline{x}\right)\psi^*\left(\underline{x},\underline{\Omega}\right)\\&\qquad=\int_{4\pi}\Sigma_s\left(\underline{x},\underline{\Omega}.\underline{\Omega}'\right)\psi^*\left(\underline{x},\underline{\Omega}'\right)d\Omega'+\frac{1}{4\pi}\delta\left(x-\underline{x}_0\right),\ \underline{x}\in V,\ \underline{\Omega}\in 4\pi\\&\psi^*\left(\underline{x},\underline{\Omega}\right)=0,\qquad\qquad\qquad\qquad\qquad\qquad\underline{x}\in\partial V,\ \underline{\Omega}.\underline{n}>0.\end{aligned} \tag{4.54}$$

Now replace $\underline{\Omega}$ by $-\underline{\Omega}$, use the identity

$$\int_{4\pi}f\left(-\underline{\Omega}.\underline{\Omega}'\right)\psi\left(\underline{\Omega}'\right)d\Omega'=\int_{4\pi}f\left(\underline{\Omega}.\underline{\Omega}'\right)\psi\left(-\underline{\Omega}'\right)d\Omega', \tag{4.55}$$

assume isotropic scattering, and substitute $\psi^*\left(x,-\underline{\Omega}\right)=\Psi\left(\underline{x},\underline{\Omega}\right)$ into Eq. (4.54), to find

$$\begin{aligned}&\underline{\Omega}.\underline{\nabla}\Psi\left(\underline{x},\underline{\Omega}\right)+\Sigma_t\left(\underline{x}\right)\Psi\left(\underline{x},\underline{\Omega}\right)\\&\qquad=\frac{1}{4\pi}\int_{4\pi}\Sigma_s\left(\underline{x}\right)\Psi\left(\underline{x},\underline{\Omega}'\right)d\underline{\Omega}'+\frac{1}{4\pi}\delta\left(\underline{x}-\underline{x}_0\right).\end{aligned} \tag{4.56}$$

Equation (4.56) is identical to a forward transport equation. Moreover

$$\begin{aligned}\phi^*\left(\underline{x}\right)&=\int_{4\pi}\psi^*\left(\underline{x},\underline{\Omega}\right)d\Omega\\&=\int_{4\pi}\Psi\left(\underline{x},-\underline{\Omega}\right)d\Omega\\&=\int_{4\pi}\Psi\left(\underline{x},\underline{\Omega}\right)d\Omega\\&=\Phi\left(\underline{x}\right).\end{aligned} \tag{4.57}$$

Therefore, the same Monte Carlo forward transport routine can be used to solve this adjoint problem. The differences are a point source at $\underline{x}_0$ instead of a volumetric source, and the tally region is now the entire sphere. Hence for this case the adjoint point flux calculation has no singularity since the scoring is over the entire sphere.

4.4.4 Numerical Examples

Two concentric spheres with radii $0.5\,\text{cm}$, and $1.0\,\text{cm}$ respectively are used to demonstrate the ability of the KDE point detector tally within a scattering region. For simplicity the detector point is placed at the center of the sphere. The inner sphere consists of a purely scattering material with $\Sigma_s = 10.0$, and the outer spherical shell contains highly scattering material with $\Sigma_s = 4.0$, and $\Sigma_a = 1.0$. Uniform volumetric source and isotropic scattering are assumed.

An adjoint MC calculation is used as a reference solution since the adjoint solution for this simple spherical case has no singularity. The flux at the center of the sphere as a function of the number of histories is shown in Figures 4.4, 4.5, and 4.6 for the reference adjoint solution, the KDE modified tally with different P values($P = 0.1,\ 0.15,\ 0.2\,\text{cm}$), and the MCNP tally with different R_0 (radius of the fictitious sphere surrounding the detector point $= 0.1, 0.15, 0.2\,\text{cm}$). Figures 4.7 and 4.8 depict the relative error as a function of number of histories with different P values and different spheres of exclusion for KDE and MCNP respectively.

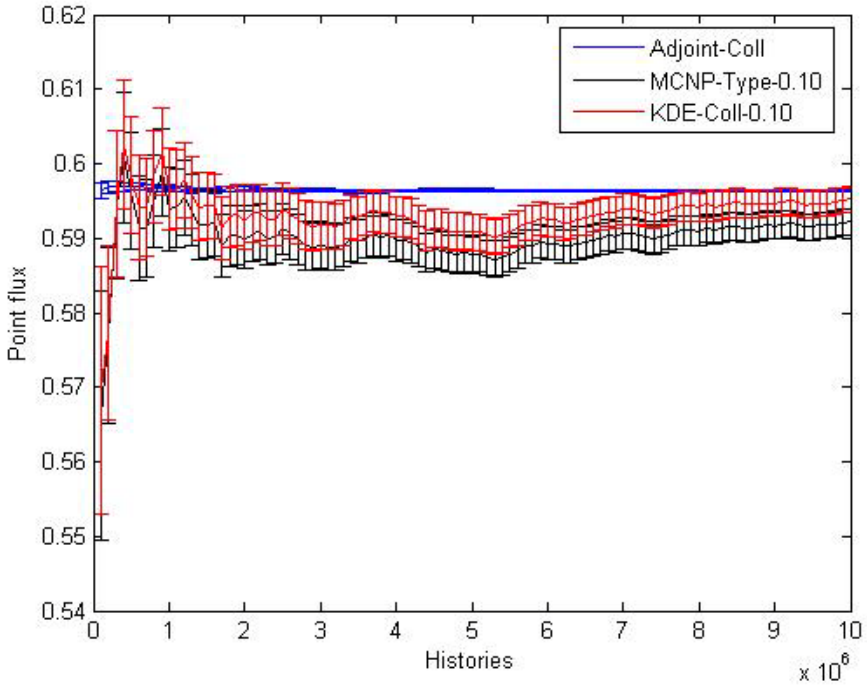

Figure 4.4 Flux at the center of the problem sphere by adjoint (collision tally), MCNP ($R_0 = 0.1\,\text{cm}$) and KDE ($P = 0.1\,\text{cm}$).

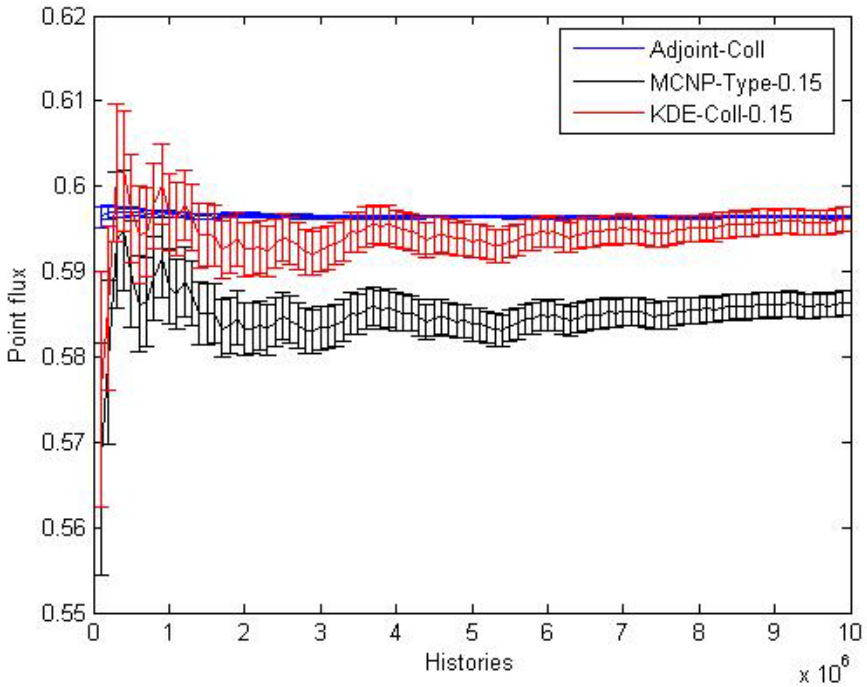

Figure 4.5 Flux at the center of the problem sphere by adjoint (collision tally), MCNP ($R_0 = 0.15\,\text{cm}$) and KDE ($P = 0.15\,\text{cm}$).

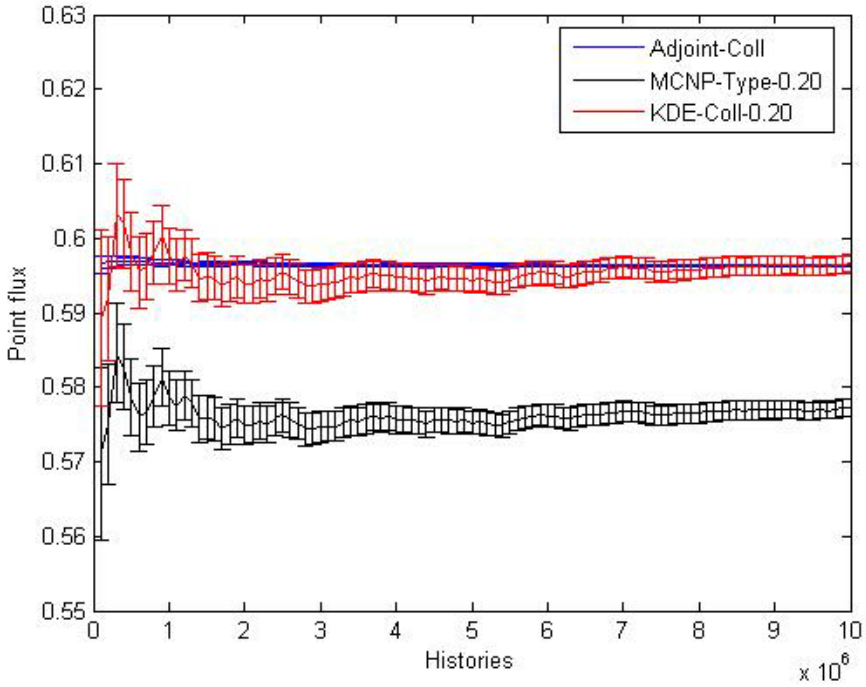

Figure 4.6 Flux at the center of the problem sphere by adjoint (collision tally), MCNP ($R_0 = 0.2\,\text{cm}$) and KDE ($P = 0.2\,\text{cm}$).

The following conclusions are based on the results shown in Figures 4.4-4.8. First, it is evident that for the KDE point detector tally, the variance will decrease with increasing P but the mean will remain the same. However for the MCNP point detector tally, if the radius of the sphere of exclusion increases, the variance will decrease but with a corresponding increase in the bias in the mean. Therefore the P parameter for the KDE point detector tally can be used as a variance reduction tool, which is not the case for the MCNP tally. Note that, both for KDE with $P > 0.0$ and MCNP with $R_0 > 0.0$ the variance converges as $1/N$, which is higher than the direct evaluation of the KDE flux tallies for estimating flux at a point. Figure 4.8 also illustrates the consequence of using the unmodified MCNP point detector estimator (the yellow line – MCNP-Type-0.00) in a highly scattering medium.

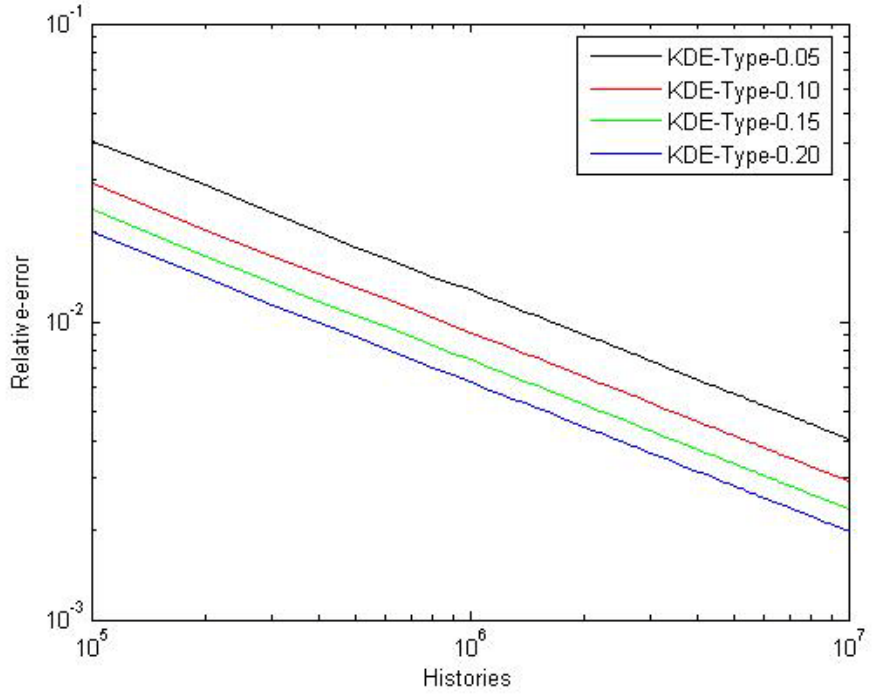

Figure 4.7 Relative error vs. number of histories for KDE point detector tally with different P values.

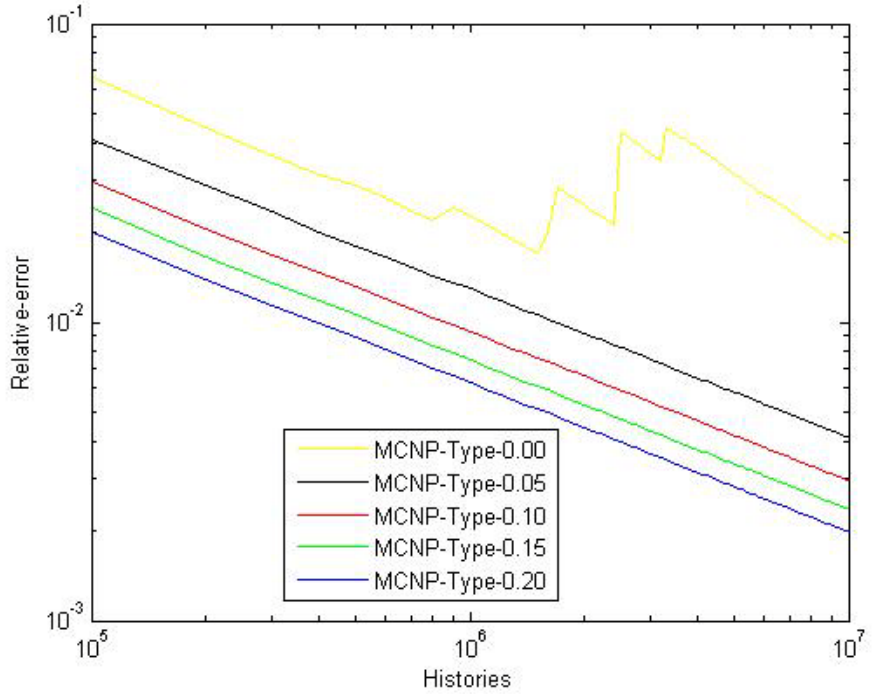

Figure 4.8 Relative error vs. number of histories for MCNP point detector tally with different spheres of exclusion.

4.4.5 Improvement in Computational Performance by Interpolation

In the previous section, it was shown for the KDE point detector tally that the parameter P has the effect of a variance reduction tool. Figure 4.7 illustrates that increase in P will decrease the variance without changing the mean of a point detector estimator in a highly scattering media. This is a unique and valuable feature of KDE point detector tally, which other proposed modifications are not able to achieve.

There is one drawback of this feature. In a highly scattering medium, if the parameter P is gradually increased, there will be more and more collisions inside the sphere defined by P surrounding the point detector. The KDE tally depends on the KDE estimation at all those collision locations inside P. Therefore, if P is sufficiently large in a highly scattering media, there will be hundreds of thousands of collisions inside P, and the KDE point detector tally requires estimation of the quantity $\phi(r')4\pi r'^2$ for each collision and this in turn may increase computation time. An alternative approach is to estimate the quantity $\phi(r')4\pi r'^2$ beforehand, by using KDE tally and the stored collision points inside P at a few hundred equally spaced points between 0 and P. The KDE point detector tally can be implemented by generating samples from the altered pdf, $\tilde{f}_2(r')$, and by evaluating the quantity $\phi(r')4\pi r'^2$ at r' by applying linear interpolation using the already banked $\phi(r')4\pi r'^2$ values in the equally spaced points between 0 and P. The linear interpolation scheme used is given by

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0}, \tag{4.58}$$

where y is evaluated at x from the given coordinates (x_0, y_0) and (x_1, y_1). The quantity $\phi(r')4\pi r'^2$ is calculated between 0 and P at $P/0.0005$ equally spaced points for the following numerical examples. Figures 4.9 - 4.13 demonstrate that for the interpolation scheme the same accuracy is achieved as in the direct case without any interpolation.

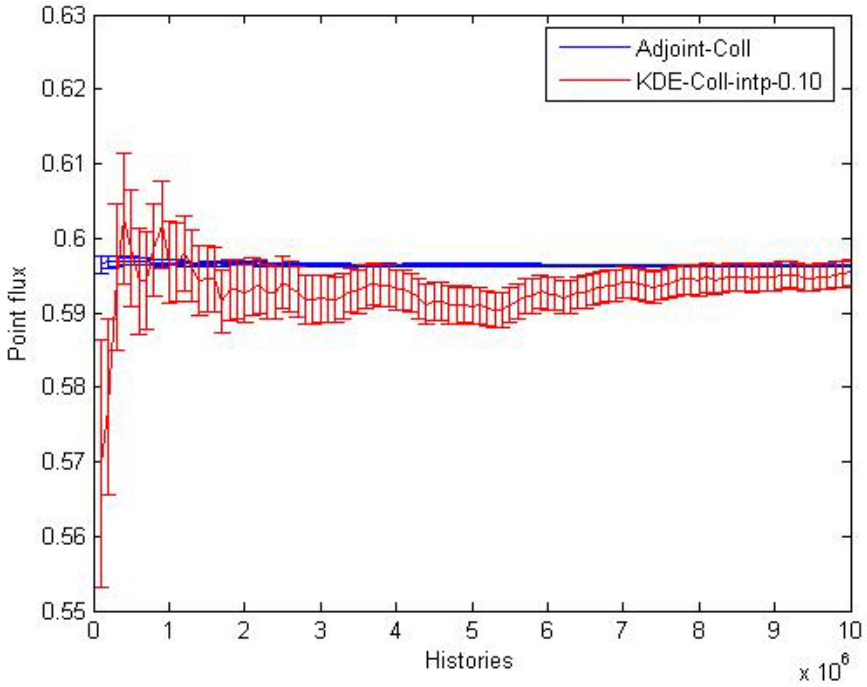

Figure 4.9 Flux at the center of the problem sphere by adjoint (collision tally), and KDE point detector tally ($P = 0.1\,\text{cm}$) with interpolation scheme.

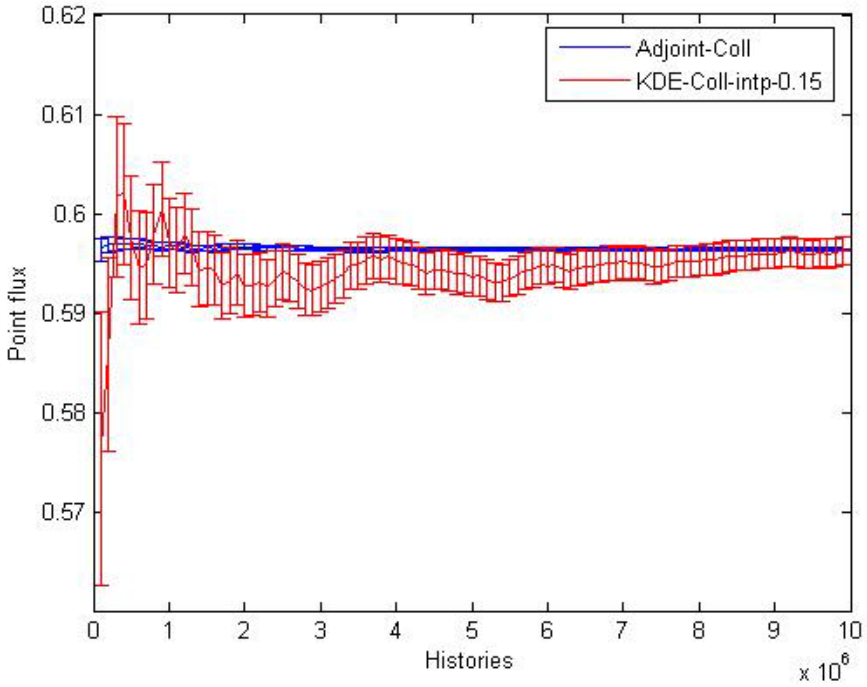

Figure 4.10 Flux at the center of the problem sphere by adjoint (collision tally), and KDE point detector tally ($P = 0.15\,\text{cm}$) with interpolation scheme.

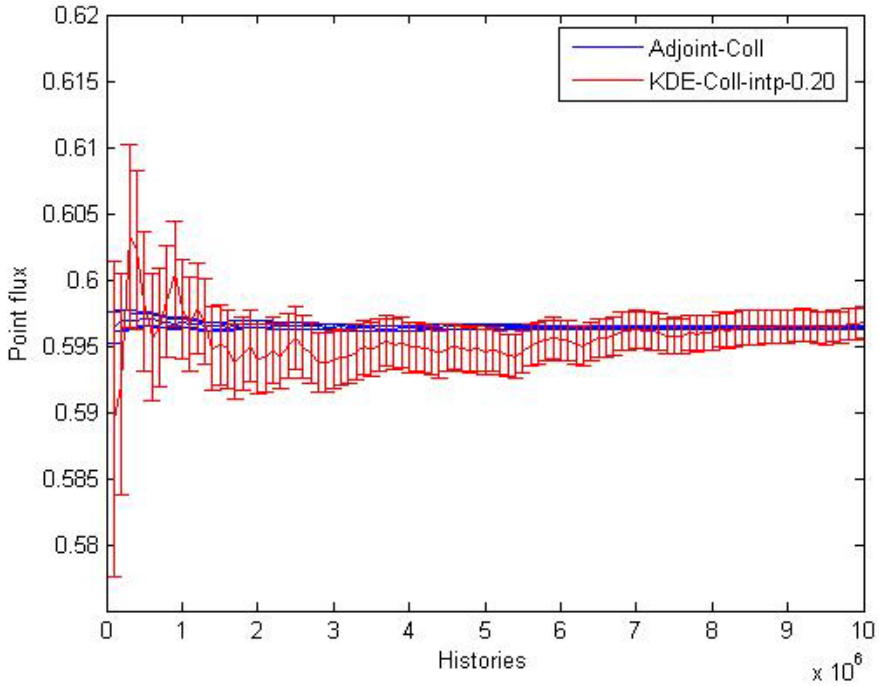

Figure 4.11 Flux at the center of the problem sphere by adjoint (collision tally), and KDE point detector tally ($P = 0.20\,\text{cm}$) with interpolation scheme.

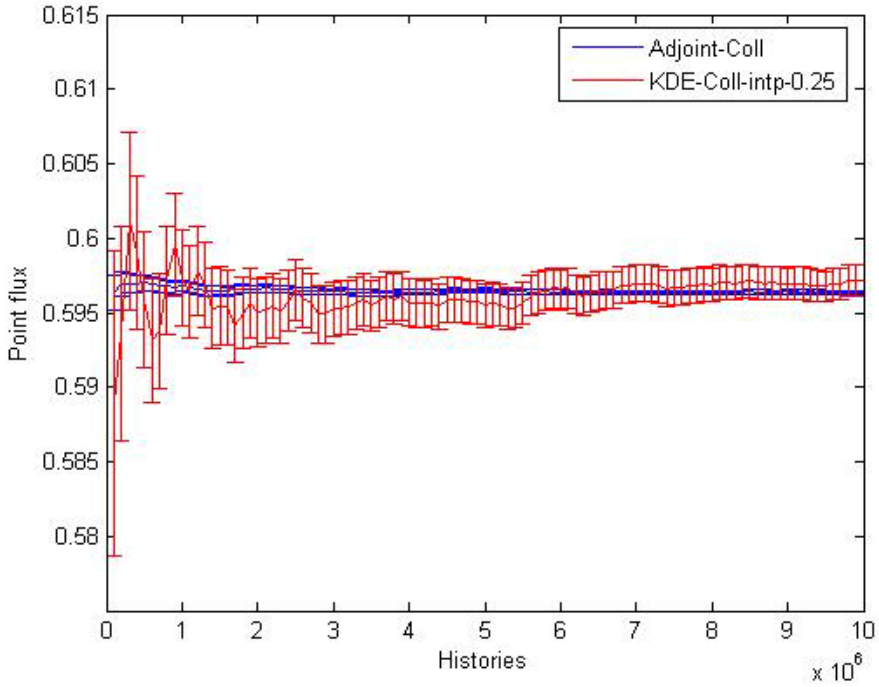

Figure 4.12 Flux at the center of the problem sphere by adjoint (collision tally), and KDE point detector tally ($P = 0.25\,\text{cm}$) with interpolation scheme.

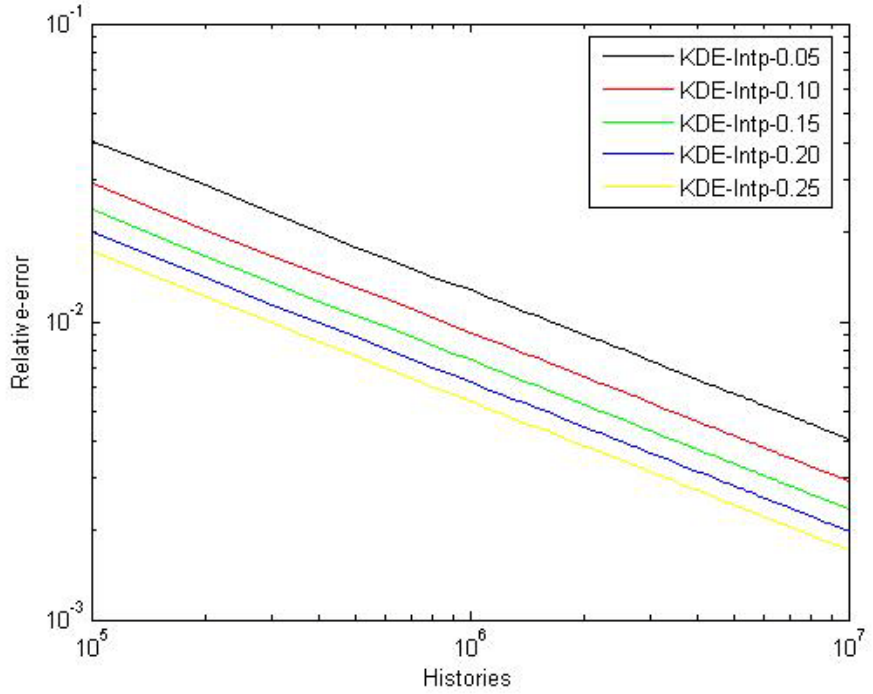

Figure 4.13 Relative error vs. number of histories for KDE point detector tally (with interpolation scheme) with different P values.

4.5 Surface Crossing Flux Tally

The surface crossing flux tally is used to estimate the average flux over a surface. This estimator, like the point detector estimator, has a singularity of the form $1/|\mu|$, where μ is the cosine of the angle between the particle direction $\underline{\Omega}$ and the surface normal $\underline{n}$. It is easy to see that the mean is preserved but the variance is infinite. As a result, the estimate of the variance for this tally is unreliable and cannot be used to gauge convergence to the true mean. The singularity is due to the fact that the score of a single particle event may approach infinity as the particle's direction becomes tangent to the surface, $|\underline{\Omega}.\underline{n}| \to 0$. However, the contribution to the average flux over a surface from particles traveling tangent to the surface is generally small. An exception could be a parallel beam incident onto a surface at grazing angle. For convex surfaces on the exterior of the system geometry with vacuum boundary conditions, the flux tangent to the surface will be zero [3]. On the other hand, it would be preferable to have a surface crossing tally that is known to work for all situations and this motivated the application of KDE to this estimator.

4.5.1 Derivation of the Surface Crossing Flux Tally

Consider the i^{th} Monte Carlo particle traveling in direction $\underline{\Omega}_{i,k}$, that crosses a surface of area A at an angle $\theta_{i,k}$ with respect to the local surface normal $\underline{n}$, for the k^{th} time. Assume that the thickness t of the surface is sufficiently small, so that no particle will suffer a collision within the volume V. Then as illustrated by the Figure 4.14, the track length $d_{i,k}$ of the i^{th} particle during k^{th} crossing in the layer is

$$d_{i,k} = \frac{t}{\left|\mu_{i,k}\right|}, \tag{4.59}$$

where

$$\left|\mu_{i,k}\right| = \left|\cos\theta_{i,k}\right| = \left|\underline{\Omega}_{i,k}.\underline{n}\right|. \tag{4.60}$$

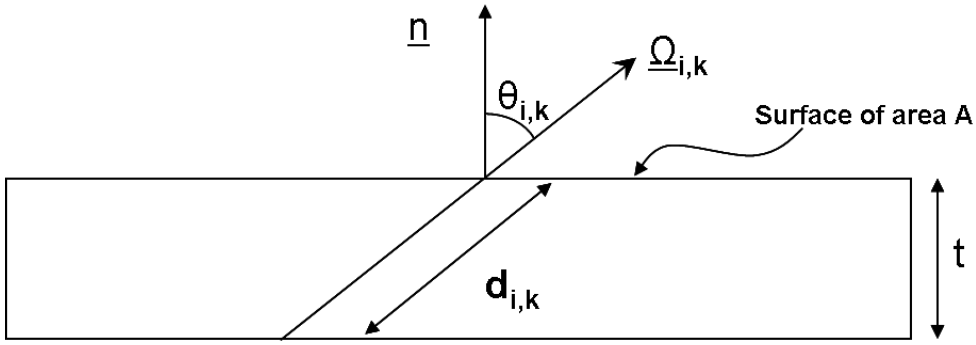

Figure 4.14 Surface crossing flux tally geometry.

Because of the absolute value of the cosine used in Eq. (4.59), the direction of the surface normal $\underline{n}$ (inner or outer) is irrelevant. By using Eq. (4.59), the track length estimator of the flux in the small box for N particle histories is given by

$$\begin{aligned}\hat{\phi} &= \frac{1}{NAt}\sum_{i=1}^{N}\sum_{k=1}^{K_i} w_{i,k} d_{i,k} \\ &= \frac{1}{NAt}\sum_{i=1}^{N}\sum_{k=1}^{K_i} w_{i,k} \frac{t}{\left|\mu_{i,k}\right|},\end{aligned} \tag{4.61}$$

where $w_{i,k}$ is the weight of the i^{th} particle just before k^{th} crossing of surface A. K_i is the total number of crossing of surface A for the i^{th} particle. By taking the limit of $t \to 0$ (ensures no collision inside the region defined by t)

$$\hat{\phi} = \frac{1}{NA} \sum_{i=1}^{N} \sum_{k=1}^{K_i} \frac{w_{i,k}}{|\mu_{i,k}|}, \tag{4.62}$$

which is the surface crossing flux estimator. In conventional Monte Carlo simulation, this estimator is executed by observing surface crossings during the random walk process, and scoring $w/A|\mu|$.

A more formal and mathematical derivation of the surface crossing flux tally is presented in the book by Dupree and Fraley [3] and will be outlined here. The scalar flux can be defined from the angular flux, $\Psi(\mu,\varphi)$, by

$$\phi = \int_{-1}^{1} \int_{0}^{2\pi} \Psi(\mu,\varphi) d\mu d\varphi, \tag{4.63}$$

Where μ and φ are the cosine of the polar angle the azimuthal angle respectively. Defining the azimuthally integrated flux, $\psi(\mu)$, by

$$\psi(\mu) = \int_{0}^{2\pi} \Psi(\mu,\varphi) d\varphi, \tag{4.64}$$

gives

$$\phi = \int_{-1}^{1} \psi(\mu) d\mu. \tag{4.65}$$

The particle current $J^+(\mu)$ is the number of particles per unit area crossing x-y plane in a positive z direction and is given by

$$J^+(\mu) = \mu\psi(\mu) = |\mu|\psi(\mu), \quad 0 < \mu < 1. \tag{4.66}$$

Similarly $J^-(\mu)$ is given by

$$J^-(\mu) = -\mu\psi(\mu) = |\mu|\psi(\mu), \quad -1 < \mu < 0. \tag{4.67}$$

Combining Eqs. (4.65), (4.66), and (4.67) we find

$$\begin{aligned}\phi &= \int_{-1}^{1} \psi(\mu)\, d\mu \\ &= \int_{-1}^{0} \frac{J^{-}(\mu)}{|\mu|} d\mu + \int_{0}^{1} \frac{J^{+}(\mu)}{|\mu|} d\mu \\ &= \int_{-1}^{1} \frac{1}{|\mu|} |\mu| \psi(\mu)\, d\mu.\end{aligned} \tag{4.68}$$

This result must be normalized by the area of the surface in concern.

4.5.2 MCNP5 Approach

For small $|\mu|$, where $|\mu| < \varepsilon$ (ε is small), MCNP assumes that the flux is linearly anisotropic and scores the expected values of $1/|\mu|$, which is $2/\varepsilon$, instead of $1/|\mu|$. A proof of this will be outlined here from Dupree and Fraley [3] and Clark [10]. Assuming the azimuthally integrated angular flux, $\psi(\mu)$, is linearly anisotropic:

$$\psi(\mu) \approx a_0 + a_1 \mu. \tag{4.69}$$

The expected value of $1/|\mu|$, for $|\mu| < \varepsilon$, is given by

$$E\left[\frac{1}{|\mu|}\right] = \frac{\int_{-\varepsilon}^{\varepsilon} \frac{1}{|\mu|} |\mu| \psi(\mu)\, d\mu}{\int_{-\varepsilon}^{\varepsilon} |\mu| \psi(\mu)\, d\mu}. \tag{4.70}$$

Replacing $\psi(\mu)$ in Eq. (4.70) by Eq. (4.69) and solving the integrals yields

$$E\left[\frac{1}{|\mu|}\right] = \frac{2}{\varepsilon}. \tag{4.71}$$

Therefore, when a particle strikes a surface detector at grazing angle with $-\varepsilon < \mu < \varepsilon$, MCNP scores $2w/A\varepsilon$, instead of $w/A|\mu|$, where w is the weight of the particle just before the surface crossing. MCNP uses $\varepsilon = 0.1$.

4.5.3 KDE Approach

For the surface crossing flux tally, score $1/|\mu|$, where μ is the angle between the direction of flight and the surface normal. These $|\mu|$'s are in the range between 0 and 1, and are distributed by $|\mu|\psi(|\mu|)d\mu$, an implicit distribution function that is a consequence of the Monte Carlo simulation. The surface crossing flux estimator in Eq. (4.68) can be recast into the following form:

$$\begin{aligned}\hat{\phi} &= \int_0^1 \frac{1}{|\mu|}|\mu|\psi(|\mu|)d\mu \\ &= \int_0^{\varepsilon}\left[\frac{1}{|\mu|}\frac{|\mu|\psi(|\mu|)}{f^*(|\mu|)}\right]f^*(|\mu|)d\mu + \int_{\varepsilon}^1 \frac{1}{|\mu|}|\mu|\psi(|\mu|)d\mu \\ &= I_1 + I_2.\end{aligned} \tag{4.72}$$

This approach is similar to that taken for the KDE point detector tally. As before, the non-singular portion of the integral, I_2, is estimated using conventional scoring method. For the integral containing the singularity, I_1, instead of scoring $1/|\mu|$, where $0<|\mu|<\varepsilon$, the $|\mu|$'s are stored. Then at the end of the simulation Monte Carlo scores the entire function within the square bracket by generating samples from the altered pdf, $f^*(|\mu|)$, where $0<|\mu|<\varepsilon$, and using the KDE estimate of $|\mu|\psi(|\mu|)$ (Eq. (2.8)) for the stored $|\mu|$ between 0 and ε. This removes the $1/|\mu|$ singularity from the Monte Carlo scoring function and yields a bounded variance. As with the KDE point detector tally, ε for the KDE surface crossing flux tally is user-defined and can be used as a variance reduction tool.

The boundary kernel method is used as a boundary correction for $|\mu|'s$ close to 0 and ε. As $f(|\mu|)=|\mu|\psi(|\mu|)$ is 0 at 0, boundary kernels may produce negative values, and more importantly the estimated function $f(|\mu|)$ converges slowly to 0 as $|\mu|\to 0$.

Therefore, a simple linear least squares fit is used as a boundary correction for KDE for points close to $|\mu| = 0$. The fit is given by

$$y = a|\mu|. \tag{4.73}$$

The y 's are calculated at $|\mu|$'s by using KDE with boundary kernels as boundary correction. For this numerical study, 25 equally spaced points on the $|\mu|$ axis with spacing 0.0005, starting from 0.005 are used. Those y 's and $|\mu|$'s are then used to calculate the least square coefficient a. For all $|\mu| < 0.005$ the least square fit is used as boundary correction. For $|\mu| \geq 0.005$, the usual KDE with boundary kernels as boundary correction (if required depending on the bandwidth) is used.

4.5.4 Numerical Example 1 for Surface Crossing Flux Estimators

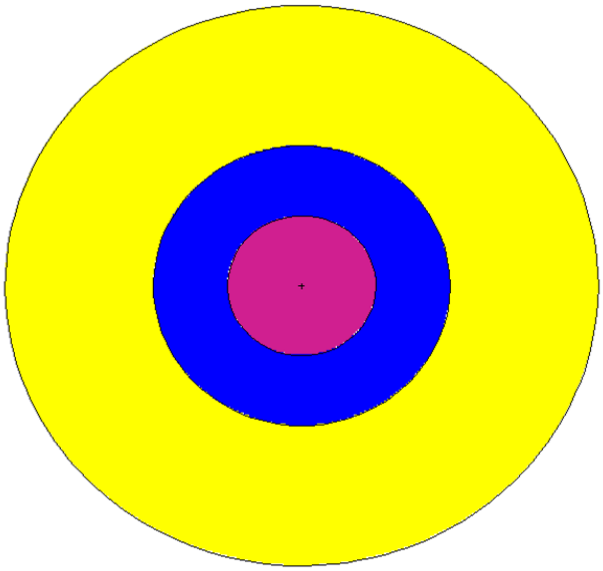

Figure 4.15 Spherical problem geometry for the surface crossing flux example 1.

Three concentric spheres with radii 0.5, 1.0, and 2.0 cm are used in example 1 to demonstrate the ability of the KDE surface crossing flux tally. Figure 4.15 illustrates the problem geometry. The inner most sphere (purple) consists of a purely scattering material with $\Sigma_s = 10.0\,\text{cm}^{-1}$, the first spherical shell (blue) also contains purely scattering material with $\Sigma_s = 8.0\,\text{cm}^{-1}$, and the outer spherical shell (yellow) contains highly scattering

material with $\Sigma_s = 4.0\,\text{cm}^{-1}$, and $\Sigma_a = 2.0\,\text{cm}^{-1}$. A uniform isotropic source inside the inner sphere (purple) and isotropic scattering are assumed. The average flux on the boundary surface between the blue and yellow shell is estimated.

An adjoint MC calculation (see Section 4.4.3) is used as a reference solution as the adjoint solution for this simple spherical case with a volumetric source has no singularity and can be used for the surface crossing flux tally. The flux on the boundary surface between the blue and yellow shell is estimated by using that boundary surface as a source and the inner sphere (which is the source for the forward calculation) as the detector. The average surface flux as a function of the number of histories is shown in Figure 4.16 for the reference adjoint solution, the KDE surface crossing tally ($\varepsilon = 0.1$), and the MCNP tally ($\varepsilon = 0.1$). Figure 4.17 depicts the same case for $\varepsilon = 0.15$. Figures 4.18 and 4.19 illustrate the relative error as a function of number of histories with different ε values for MCNP and KDE tallies, respectively. It is evident from Figures 4.16 – 4.19 that for the KDE surface crossing tally, the variance will decrease with increasing ε but the mean will remain the same. However for the MCNP type surface crossing tally, if ε increases, the variance will decrease but with a corresponding increase in the bias in the mean. Therefore we can use the ε parameter for the KDE surface crossing tally as a variance reduction tool, which is not the case for the MCNP tally. Note the above results indicate ε is not as powerful a variance reduction tool as is P in the KDE point detector tally.

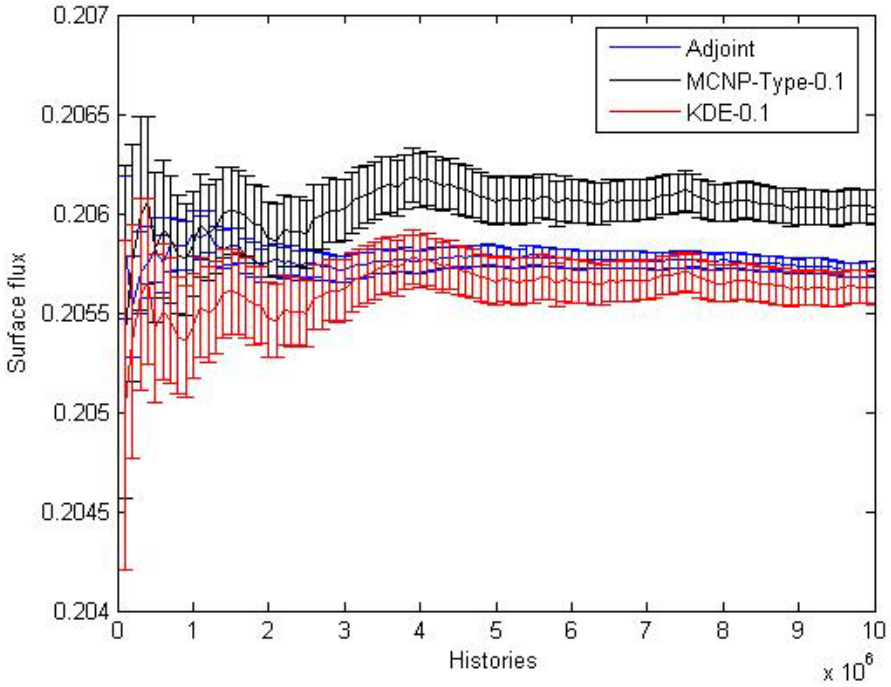

Figure 4.16 Surface crossing flux for example 1 by adjoint, MCNP ($\varepsilon = 0.1$), and KDE ($\varepsilon = 0.1$) surface crossing tally.

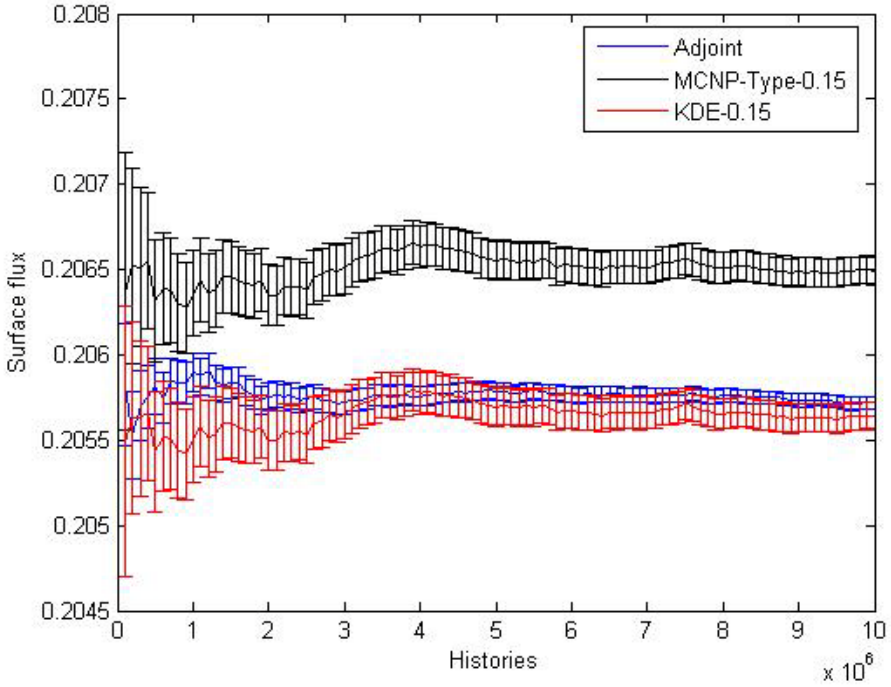

Figure 4.17 Surface crossing flux for example 1 by adjoint, MCNP ($\varepsilon = 0.15$), and KDE ($\varepsilon = 0.15$) surface crossing tally.

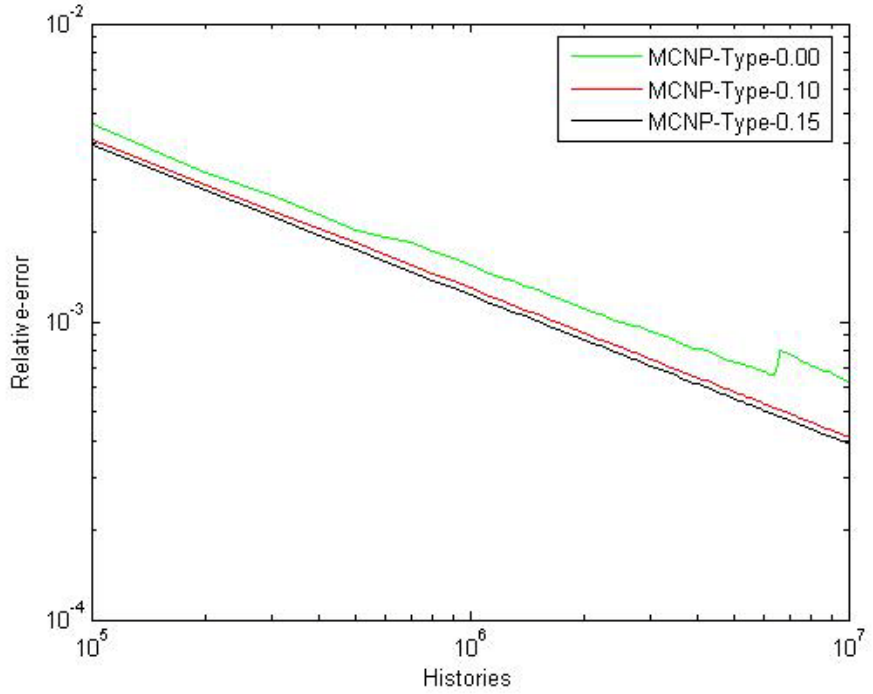

Figure 4.18 Relative error vs. number of histories for MCNP surface crossing flux tally (F2 tally) for example 1 with different ε values.

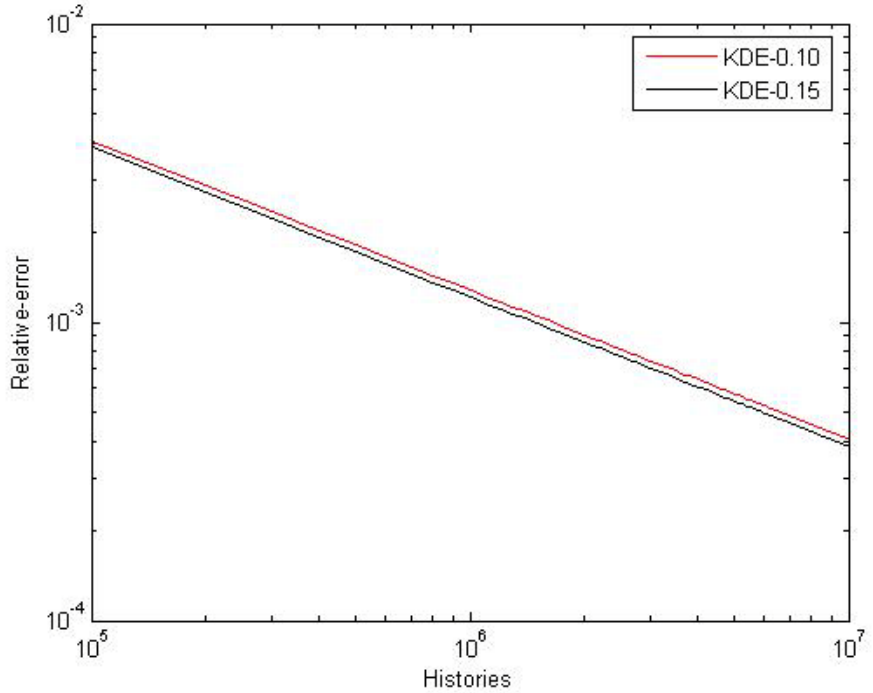

Figure 4.19 Relative error vs. number of histories for KDE surface crossing flux tally for example 1 with different ε values

4.5.5 Numerical Example 2 for Surface Crossing Flux Estimators

A $1\,\text{cm}^3$ cube is used to test the surface crossing flux tally for numerical example 2. The averaged flux over a surface, which is parallel to the x-y plane with $z = 0.5\,\text{cm}$, is estimated. A very thin box source extending from $z = 0.495\,\text{cm}$ to $z = 0.505\,\text{cm}$, $x = 0.0\,\text{cm}$ to $x = 0.3\,\text{cm}$, and $y = 0.0\,\text{cm}$ to $y = 1.0\,\text{cm}$ surrounding the tally surface is assumed with highly scattering material with $\Sigma_s = 7.0\,\text{cm}^{-1}$, and $\Sigma_a = 3.0\,\text{cm}^{-1}$ inside the cube. The problem geometry is illustrated by Figure 4.20. Isotropic scattering is also assumed. An adjoint calculation is used as a reference solution for this problem as the adjoint calculation has no singularity. For the adjoint calculation the average flux over the surface is estimated by using that surface as the source and the thin box surrounding that surface (source for the forward calculation) as the detector. The average surface flux on the desired surface as a function of the number of histories is shown in Figure 4.21 for the reference adjoint solution, the KDE surface crossing tally ($\varepsilon = 0.1$), and the MCNP F2 tally ($\varepsilon = 0.1$). Figure 4.22 depicts the same case for $\varepsilon = 0.15$. Figures 4.23 and 4.24 illustrate the relative error as a function of number of histories with different ε values for MCNP and KDE respectively. Note this example case also indicates that the ε parameter is not as powerful variance reduction tool as is P in the KDE point detector tally. An interpolation scheme as discussed for the KDE modified point detector tally could be employed for the KDE surface crossing flux tally to improve the computational speed.

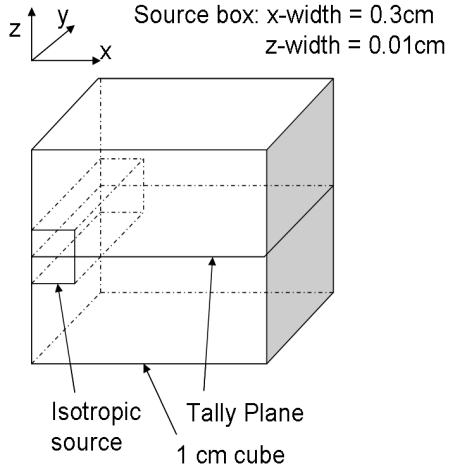

Figure 4.20 Cubical problem geometry for the surface crossing flux example 2.

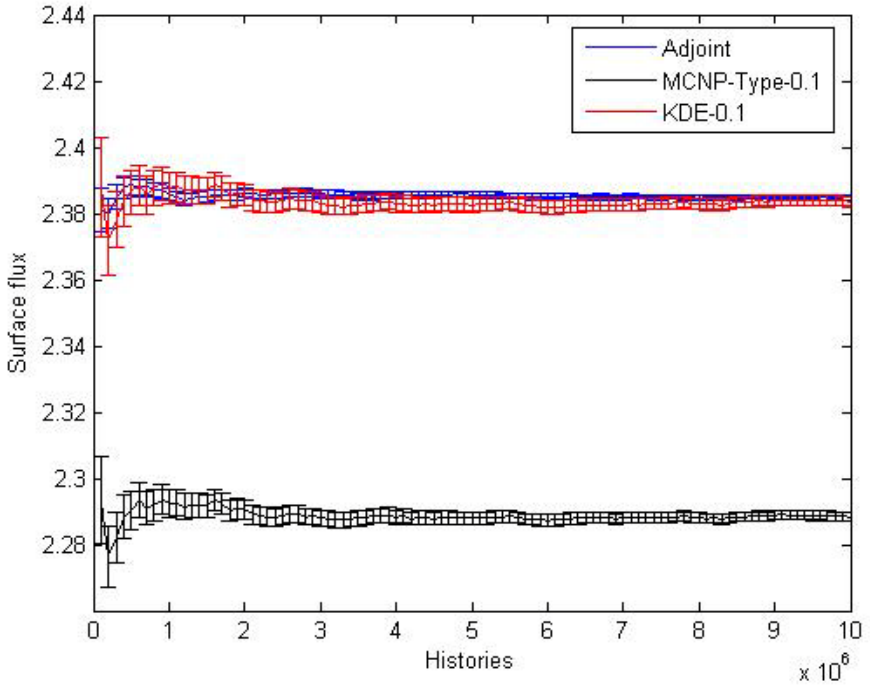

Figure 4.21 Surface crossing flux for example 2 by adjoint, MCNP ($\varepsilon = 0.1$), and KDE ($\varepsilon = 0.1$) surface crossing tally.

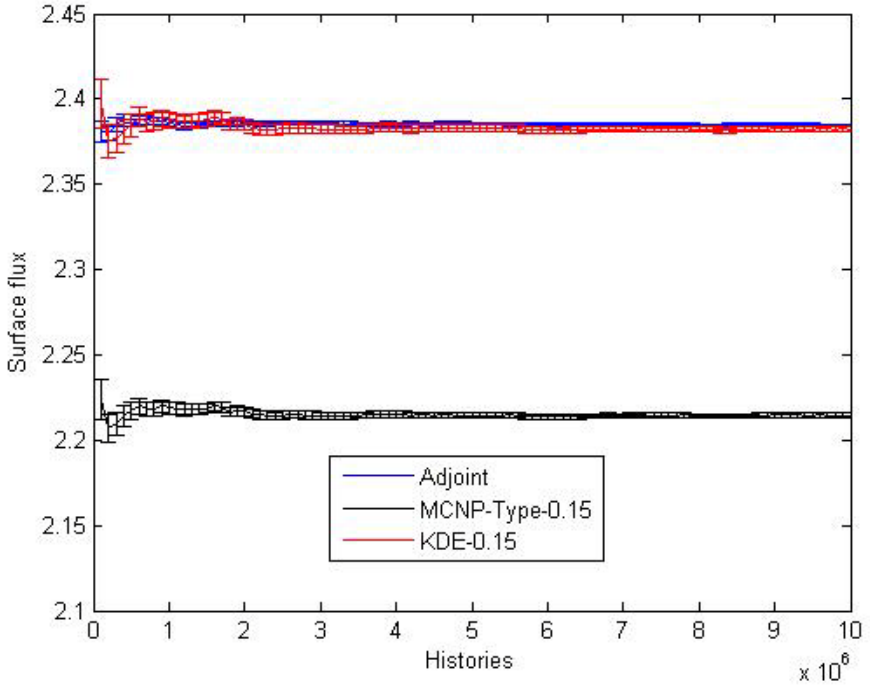

Figure 4.22 Surface crossing flux for example 2 by adjoint, MCNP ($\varepsilon = 0.15$), and KDE ($\varepsilon = 0.15$) surface crossing tally.

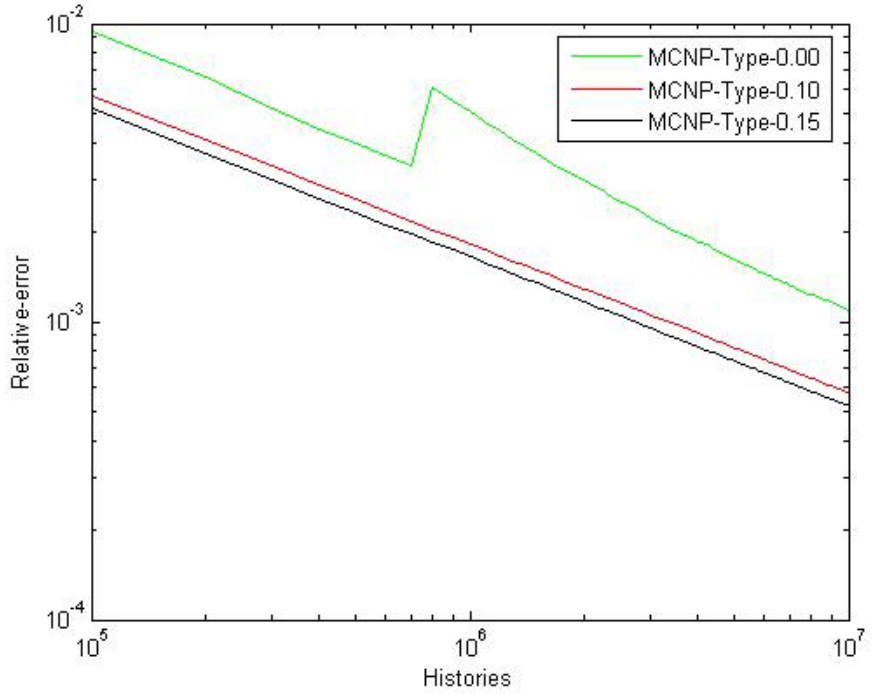

Figure 4.23 Relative error vs. number of histories for MCNP surface crossing flux tally (F2 tally) for example 2 with different ε values.

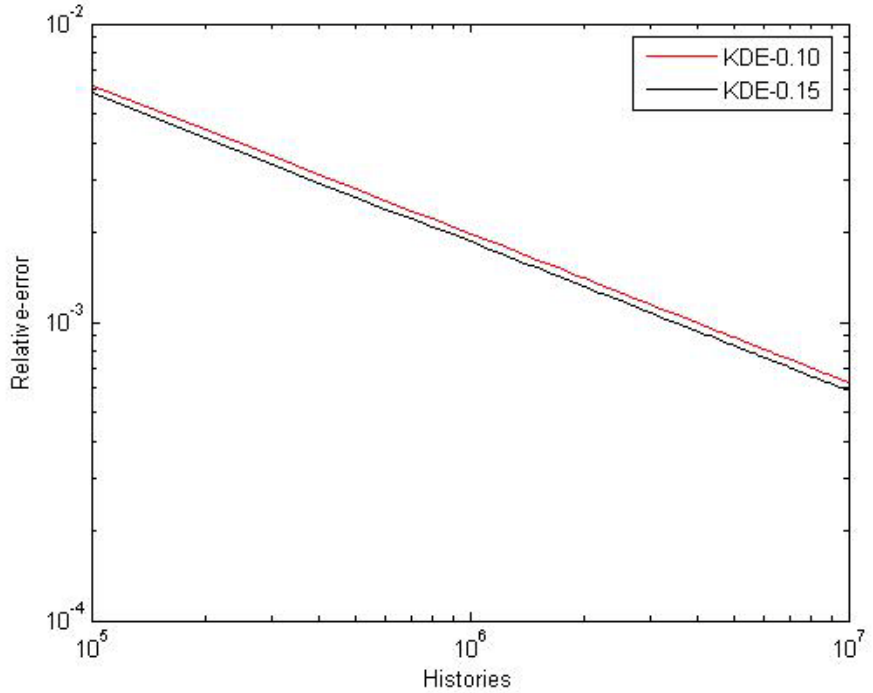

Figure 4.24 Relative error vs. number of histories for KDE surface crossing flux tally for example 2 with different ε values

4.6 References

[1] L. L. Carter and E. D. Cashwell, *Particle Transport Simulation with the Monte Carlo Method*, Technical Information Center, Energy Research and Development Administration, Oak Ridge (1975).

[2] M. H. Kalos, "On the Estimation of Flux at a Point by Monte Carlo," *Nuclear Science and Engineering*, **16**, 111-117, (1963).

[3] S. A. Dupree and S. K. Fraley, *A Monte Carlo Primer – A Practical Approach to Radiation Transport*, Plenum, New York (2002).

[4] H. J. Kalli and E. D. Cashwell, "Evaluation of Three Monte Carlo Estimation Schemes for Flux at a Point," *LA-6865-MS*, Los Alamos Scientific Laboratory, Los Alamos, New Mexico (1977).

[5] H. A. Steinberg and M. H. Kalos, "Bounded Estimator for Flux at a Point in Monte Carlo," *Nucl. Sci. Eng.*, **44**, 406-412 (1971).

[6] H. Reif, A. Dubi, and T. Elperin, "Track Length Estimation Applied to Point Detectors," *Nucl. Sci. Eng.*, **87**, 59-71 (1984).

[7] Monte Carlo Team, "MCNP – A General Monte Carlo N-Particle Transport Code, Version 5," *LA-UR-03-1987*, Los Alamos National Laboratory (2003).

[8] S. K. Fraley and T. J. Hoffman, "Bounded Flux-at-a-Point Estimators for Multigroup Monte Carlo Computer Code," *Nucl. Sci. Eng.*, **70**, 14-19 (1979).

[9] I. Lux and L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, CRC Press, Boca Raton, Florida (1991).

[10] F. H. Clark, "Variance of Certain Flux Estimators Used in Monte Carlo Calculations," *Nucl. Sci. Eng.*, **27**, 235-239 (1967).

[11] K. Banerjee and W. R. Martin, "Kernel Density Estimation Method for Monte Carlo Tallies with Unbounded Variance," *Trans. Am. Nucl. Soc.*, Atlanta, GA, Vol. **101**, pp. 430-432 (November 2009).

[12] K. Banerjee and W. R. Martin, "Applying the Kernel Density Flux Estimator to Estimate Flux at a Point," *Trans. Am. Nucl. Soc.*, Atlanta, GA, Vol. **100**, pp. 294-296 (June 2009).

[13] J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*, Dover Publications, Mineola, New York (2007).

[14] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, Fourth ed, Tata McGraw-Hill, New Delhi (2002).

[15] M. C. Jones, "Simple Boundary Correction for Kernel Density Estimation," *Statistics and Computing,* **3**, pp. 135-146 (1993).

[16] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*, John Wiley & Sons (1976).

[17] E. Larsen, NERS 644, *Classroom Notes*, University of Michigan, Fall Semester (2008).

CHAPTER 5

Kernel Density Estimation Method for Monte Carlo Eigenvalue Calculations

5.1 Introduction

The multiplication factor of a system containing fissionable materials is used to describe the criticality of the system. The multiplication factor is the ratio of the number of neutrons in one generation to the number of neutrons in the previous generation. A generation is defined by the lifetime of a neutron. For finite systems, the multiplication factor is known as the effective neutron multiplication factor, k_{eff}. k_{eff} can also be defined as a ratio of neutron production to neutron loss in the system. $k_{eff} < 1$ denotes that the system loses more neutrons that it produces. In this case, the system is subcritical and the neutron population dies off in time. The system is critical when $k_{eff} = 1$ and it maintains a constant average neutron population in time. In other words, the system is in steady state. A supercritical system with $k_{eff} > 1$ produces more neutrons than it loses. The neutron population will grow in time without any bound.

Knowledge of criticality is essential for designing and operating nuclear reactors as well as dealing with criticality issues associated with fuel fabrication, fuel reprocessing, and spent fuel storage. Experiments or prototypes are the best way to gather knowledge about the criticality of a system. However, experiments and prototypes are expensive and system specific. On the other hand, with the advent of modern computers, numerical methods, both deterministic and Monte Carlo (stochastic), are inexpensive and used frequently for criticality calculations. The Monte Carlo method is an emerging tool for performing reactor calculations because of its ability to handle complex geometry and complex physics.

However, for some systems Monte Carlo criticality calculations may take exorbitant computing time to complete. Monte Carlo criticality calculations depend on source or power iterations, and tallies should not be accumulated until the fission source has converged to its fundamental distribution within statistical fluctuations. Any tallies taken before the source distribution has converged to the fundamental mode will be biased. The rate of convergence of the source to the fundamental mode depends on the dominance ratio of the system, which is defined as the ratio of the second and first eigenvalues of the system. If the dominance ratio is close to unity, which is the case for large reactors with small leakage, heavy-water moderated or reflected reactors, and loosely-coupled systems, the convergence to the fundamental eigenfunction [1] is slow and the source iteration error will decrease slowly.

This chapter discusses the application of KDE to Monte Carlo eigenvalue calculations for reactor analyses. KDE is used to estimate the shape of the fission source at the end of each neutron generation, and samples from this estimated source distribution are used as the starting particles for the next generation. The results for a large, loosely coupled system in 1D slab geometry and in 3D checkerboard geometry are promising: the converged fission source distribution is satisfactory, and there is a substantial increase in fission source convergence as measured by the relative source entropy [2].We also note that traditional sampling of the fission source introduces correlation among the source points between neutron generations, and it is thought that the KDE method will reduce this correlation by decreasing the propagation of the correlated source particles from generation to generation.

Over the years the problems associated with fission source convergence have been studied extensively. Whitesides [3] in 1971 identified this source convergence issue in his famous "k_{eff} of the world" paper. Swaja [4] successfully accelerated Monte Carlo fission source convergence using source extrapolation. Urbatsch [5] in 1995 proposed three methods for improving source convergence in systems with dominance ratio close to unity, including a fission matrix approach and a diffusion synthetic method, as well as a hybrid method for source convergence.

Kuroishi and Nomura [6] postulated a new source acceleration method based on matrix eigenvector calculations with different zoning method. Yamamoto and Miyoshi [7] has proposed a new algorithm for applying Wielandt's method to Monte Carlo criticality calculations. Later in 2005, Yamamoto [8] has proposed a conjugate gradient scheme to accelerate Monte Carlo eigenvalue problems. Finch et al [9] has developed another new method, known as the vacation matrix method, to estimate the source distribution in Monte Carlo criticality calculations although definitive results were not presented. Another effective method to estimate the fission source distribution is developed by Larsen and Yang [10]. Their method is known as the functional Monte Carlo method.

Griesheimer and Toth [11] proposed a novel technique for accelerating the convergence of iterative power method by modifying the fixed-parameter source extrapolation method. Their technique is known as smoothed residual acceleration. Another method by Griesheimer et al [12] applies functional expansion tallies (FET) to estimate functional approximations for the shape of the fission source distribution. The KDE method [13], which is the topic for this chapter, is based on the same philosophy as the FET approach of estimating the shape of the fission source distribution.

5.2 Monte Carlo Criticality Calculation

For steady-state reactor calculations, the neutron transport equation is given by

$$\begin{aligned}
&\underline{\Omega}.\underline{\nabla}\psi\left(\underline{x},\underline{\Omega},E\right)+\Sigma_t\left(E\right)\psi\left(\underline{x},\underline{\Omega},E\right)\\
&\qquad=\int_0^\infty\int_{4\pi}\Sigma_s\left(\underline{\Omega}'.\underline{\Omega},E'\to E\right)\psi\left(\underline{x},\underline{\Omega}',E'\right)d\Omega'dE'\\
&\qquad+\frac{\chi_p\left(E\right)}{4\pi k_{eff}}\int_0^\infty\int_{4\pi}\upsilon\Sigma_f\left(E'\right)\psi\left(\underline{x},\underline{\Omega}',E'\right)d\Omega'dE', \qquad (5.1)\\
&\qquad\underline{x}\in V,\ \underline{\Omega}\in 4\pi,\ 0<E<\infty,\\
&\psi\left(\underline{x},\underline{\Omega},E\right)=0,\quad \underline{x}\in\partial V,\ \underline{\Omega}.\underline{n}<0,\ 0<E<\infty,
\end{aligned}$$

where ψ is the angular neutron flux, which is defined by the product of the neutron speed and the angular number density, $\underline{x}$ is the spatial coordinate vector, $\underline{\Omega}$ is unit vector towards the direction-of-flight, E is the energy, Σ_t is the macroscopic total cross section,

Σ_s is the macroscopic differential scattering cross section, Σ_f is the macroscopic fission cross section, $\chi_p(E)$ is the prompt fission energy spectrum, υ is the average number of neutrons generated per fission event. These equations always have the zero solution: $\psi = 0$. The goal here is to find out the largest value of k_{eff} such that a nonzero solution of ψ exists. The largest k_{eff} is called the fundamental eigenvalue of the system and the corresponding ψ is called the fundamental eigenfunction. The steady-state k-eigenvalue neutron transport equation can be written in operator notation:

$$(L + T - S)\psi = \frac{1}{k_{eff}} M\psi, \tag{5.2}$$

where

$$Lf = \underline{\Omega}.\underline{\nabla} f, \tag{5.3}$$

$$Tf = \Sigma_t(E) f, \tag{5.4}$$

$$Sf = \int_0^\infty \int_{4\pi} \Sigma_s(\underline{\Omega}'.\underline{\Omega}, E' \to E) f d\Omega' dE', \tag{5.5}$$

$$Mf = \frac{\chi_p(E)}{4\pi} \int_0^\infty \int_{4\pi} \upsilon\Sigma_f(E') f d\Omega' dE', \tag{5.6}$$

for any function $f(\underline{x}, \underline{\Omega}, E)$. L is the leakage, T is the loss operator due to collisions and S is the in-scattering operator. Define the transport-collision operator $A = L + T - S$ and the fission source $Q(\underline{x}) = M\psi$. Manipulate Eq. (5.2) and use the transport-collision operator and fission source to obtain

$$Q = \frac{1}{k_{eff}} MA^{-1}Q. \tag{5.7}$$

Equation (5.7) is in a form that directly corresponds to Monte Carlo fission source iteration. Monte Carlo criticality calculations consist of outer iterations, which is nothing but a fixed source calculation. These outer iterations are called cycles or batches or generations. In Monte Carlo criticality procedures, the total number of neutron histories simulated is divided into M batches, and each batch consists of N neutrons. The neutron

histories in one batch are used to estimate the neutron source in the next batch and the k_{eff}. The typical criticality calculation begins with an initial fission source guess, for example a uniform distribution over the entire system, a point source, or a source that could take advantage of the prior information about the system, and run enough batches to converge to the true fission source distribution within statistical fluctuations. These initial batches are called inactive batches. Monte Carlo criticality procedures start accumulating useful data only after the source is converged. The batches, run after the source convergence, are known as active batches. In this chapter KDE is applied to fission source iteration to reduce the number of inactive batches, yielding faster convergence to the true fission source distribution.

In some Monte Carlo codes, the number of particles starting each batch is a constant. In other codes, the total weight W starting each batch is constant. During a batch Monte Carlo tracks each particle through various random events, either collisions or boundary crossings, until the particle is absorbed or leaks from the system. If at a collision a fission event occurs, then Monte Carlo banks the location of that collision until the next batch. In Eq. (5.7) Q is the fission bank, A^{-1} is the inverted transport-collision operator (random walk plus collision analysis), M is the fission operator, which stores expected number of fission sites following a collision, and k_{eff} is the multiplication factor.

Define the index n to be a fission generation. The solution of Eq. (5.7) by fission source iterations can be expressed as follows:

$$Q^{(n+1)} = \frac{1}{k^{(n)}} MA^{-1}Q^{(n)},\ n = 1, 2, ... \tag{5.8}$$

where $k^{(0)} = 1$ and $Q^{(0)} = \text{the initial source guess}$. The actual Monte Carlo implementation of the operator $\frac{1}{k^{(n)}} MA^{-1}$ in Eq. (5.8) consists of a random walk plus collision analysis resulting in the following number of fission sites for the next cycle:

$$N = W \frac{\upsilon \Sigma_f}{fac.\Sigma_t} \frac{1}{k^{(n)}} + \xi \tag{5.9}$$

In Eq. (5.9), W is the weight of the neutron entering the collision, $\xi[0,1]$ is a random number, and $k^{(n)}$ is the estimate of k_{eff} from the previous cycle. Equation (5.9) is scaled by $k^{(n)}$ to maintain roughly the same number of fission neutrons in each batch. This will eliminate the computational difficulties arise from an increasing (in the case of supercritical system) or decaying (subcritical) neutron population. However, this normalization will introduce a bias. The bias is usually insignificant as it is inversely proportional to the number of histories per batch [14, 15]. Sometime the factor, fac, is used to increase the source sites to be sampled from the next generation. For example $fac = 0.8$ could be used to generate approximately 25 percent extra source sites for the next batch.

At the end of the current generation, the eigenvalue must be updated. Integrate Eq. (5.7) over all volume to obtain

$$k_{eff} = \frac{\int_V MA^{-1}Q(\underline{x})dV}{\int_V Q(\underline{x})dV} = \frac{\int MA^{-1}QdV}{\int QdV}. \tag{5.10}$$

The latest information available for Q is used to estimate k_{eff} at the end of the current generation:

$$k^{(n+1)} = \frac{\int MA^{-1}Q^{(n)}dV}{\int Q^{(n)}dV}. \tag{5.11}$$

Here $Q^{(n)}$ is used instead of $Q^{(n+1)}$ because $MA^{-1}Q^{(n+1)}$ is not known and the numerator and denominator must be evaluated at the same time. By substituting Eq. (5.8) into Eq. (5.11) the following expression is obtained for the updated eigenvalue at the end of generation n:

$$k^{(n+1)} = k^{(n)} \frac{\int Q^{(n+1)}dV}{\int Q^{(n)}dV}. \tag{5.12}$$

The ratio in Eq. (5.12) is simply the ratio of the new and old fission sources. Equivalently, one can use track length, collision or absorption estimators to estimate these quantities. The track length estimator of k_{eff} is given by

$$k_{path} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C_i} w_{i,c} d_{i,c} . \upsilon\Sigma_f , \tag{5.13}$$

where N is the number of particles per batch, C_i is the total number of events (collisions, boundary crossing etc) for i^{th} history, $w_{i,c}$ is the weight of the particle between event $c-1$ and c, and $d_{i,c}$ is the track length between event $c-1$ and c. Similarly the collision and absorption estimators of k_{eff} are given by

$$k_{coll} = \frac{1}{N}\sum_{\substack{all \\ collisions}} w_i . \frac{\upsilon\Sigma_f}{\Sigma_t}, \tag{5.14}$$

$$k_{abs} = \frac{1}{N}\sum_{\substack{all \\ absorptions}} w_i . \frac{\upsilon\Sigma_f}{\Sigma_a}. \tag{5.15}$$

The final k_{eff} is calculated by taking the average over all the active cycles.

5.3 Source Convergence Difficulties

In this section the convergence of the k_{eff} and the fission source iteration and their dependence on the dominance ratio will be discussed. This discussion is based on Brown's [16] notes on Monte Carlo particle transport. Introducing $MA^{-1} = F$ in Eq. (5.7) yields

$$Q = \frac{1}{k_{eff}} FQ. \tag{5.16}$$

Q can be expanded in terms of the eigenfunctions $v_i(\underline{x})$ of F to find

$$Q = \sum_{i=0}^{\infty} a_i v_i = a_0 v_0 + a_1 v_1 + a_2 v_2 + ... \tag{5.17}$$

Note the eigenfunctions are orthogonal:

$$\int v_i v_k = \delta_{ik}, \tag{5.18}$$

where δ_{ik} is the Kronecker delta function. Orthogonality of the eigenfunctions yields

$$a_i = \int Q v_i dV. \tag{5.19}$$

Since the v_i's are a complete set of eigenfunctions of the operator F with eigenvalues k_i's, the following can be written

$$v_i = \frac{1}{k_i} F v_i. \tag{5.20}$$

Now expand the initial source guess in terms of the eigenfunctions, yielding

$$Q^{(0)} = \sum_{i=0}^{\infty} a_i^{(0)} v_i. \tag{5.21}$$

After applying recursion to Eq. (5.16), substituting the expansion of $Q^{(0)}$ from Eq. (5.21), and using Eq. (5.20), an expression for the updated fission source can be obtained:

$$\begin{aligned} Q^{(n+1)} &= \frac{1}{k^{(n)}} F.Q^{(n)} \\ &= \frac{1}{k^{(n)}} \ldots\ldots \frac{1}{k^{(0)}} F^{n+1} . \left\{ a_0^{(0)} v_0 + a_1^{(0)} v_1 + a_2^{(0)} v_2 + \ldots\ldots \right\} \\ &= \left[\prod_{m=0}^{n} \frac{1}{k^{(m)}} \right] \left\{ a_0^{(0)} k_o^{n+1} v_0 + a_1^{(0)} k_1^{n+1} v_1 + a_2^{(0)} k_2^{n+1} v_2 + \ldots\ldots \right\} \\ &= \left[\prod_{m=0}^{n} \frac{k_o}{k^{(m)}} \right] a_o^{(0)} \left\{ v_0 + \frac{a_1^{(0)}}{a_o^{(0)}} \left(\frac{k_1}{k_0} \right)^{n+1} v_1 + \frac{a_2^{(0)}}{a_o^{(0)}} \left(\frac{k_2}{k_0} \right)^{n+1} v_2 + \ldots. \right\}. \end{aligned} \tag{5.22}$$

Combining Eq. (5.12) and (5.22), the following expression for $k^{(n+1)}$ in terms of the dominance ratio can be derived:

$$
\begin{aligned}
k^{(n+1)} &= k_0 \frac{\left[1+\frac{a_1^{(0)}}{a_o^{(0)}}\left(\frac{k_1}{k_0}\right)^{n+1}.G_1+\frac{a_2^{(0)}}{a_o^{(0)}}\left(\frac{k_2}{k_0}\right)^{n+1}.G_2+\ldots\right]}{\left[1+\frac{a_1^{(0)}}{a_o^{(0)}}\left(\frac{k_1}{k_0}\right)^{n}.G_1+\frac{a_2^{(0)}}{a_o^{(0)}}\left(\frac{k_2}{k_0}\right)^{n}.G_2+\ldots\right]} \\
&\approx k_0\left[1+\frac{a_1^{(0)}}{a_o^{(0)}}\left(\frac{k_1}{k_0}\right)^{n+1}.G_1+\ldots\right].\left[1-\frac{a_1^{(0)}}{a_o^{(0)}}\left(\frac{k_1}{k_0}\right)^{n}.G_1\right] \\
&\approx k_0\left[1+\frac{a_1^{(0)}}{a_o^{(0)}}\left(\frac{k_1}{k_0}\right)^{n}\left(\frac{k_1}{k_0}-1\right).G_1+\ldots\right],
\end{aligned}
\tag{5.23}
$$

where

$$
G_i = \frac{\int v_i\left(\underline{x}\right)dV}{\int v_0\left(\underline{x}\right)dV}. \tag{5.24}
$$

As $k_0 > k_1 > k_2 > \ldots..$, after the initial transients both the source Q and the k_{eff} will depend mainly on the dominance ratio of the system, which is given by k_1/k_0. If the dominance ratio is low ($k_1/k_0 << 1$) then from Eq. (5.22) and (5.23), it is evident that the source Q and the k_{eff} will converge to the fundamental eigenfunction v_0 and the fundamental eigenvalue k_0 respectively without much effort. For systems with dominance ratio close to unity the error in k_{eff} may be small, since the factor $\left(k_1/k_0 - 1\right)$ (in Eq. (5.23)) is small, but the same is not true for the source Q. Therefore, for problem with high dominance ratio k_{eff} may appear converged even if the fission source distribution is not converged.

5.4 Measurement of Fission Source Convergence

The relative information entropy is employed to diagnose the stationarity of the Monte Carlo fission source distribution [2]. Let $S^B\left(i\right)$ and $T^B\left(i\right)$ be two different binned sources that are normalized to unity. The relative entropy of S^B w.r.t. T^B is defined by

$$
D\left(S^B \,||\, T^B\right) = \sum_{i=1}^{B} S^B\left(i\right)\log_2\left(\frac{S^B\left(i\right)}{T^B\left(i\right)}\right). \tag{5.25}
$$

Relative information entropy can be viewed as the statistical distance between two distributions, S^B and T^B. If $S^B = T^B$, then the relative entropy is zero, otherwise it is always positive. The posterior diagnosis is performed as prescribed by Ueki and Brown [2] by checking whether or not the relative entropy crosses the maximum stationary level (msl) downward before active cycles begin. The msl is calculated following the method proposed by Ueki and Brown [2].

5.5 Conventional Fission Bank Sampling

The goal is to select N fission sites for the current generation from N' fission sites from the previous generation, which are in the fission bank. The following algorithm, which is taken from Knuth [17], is used for this purpose.

Step 1: Initialize $t = 0$ and $c = 0$. c represents the total number of fission sites from the fission bank we have dealt with and t represents the number of fission sites selected so far.

Step 2: Begin a loop with index i, where i runs from 1 to N', and select the i^{th} fission site from the fission bank.

Step 3: Calculate $p = \left\lfloor \frac{N-t}{N'-c} + \xi \right\rfloor$, where $\lfloor x \rfloor$ is the floor function (largest integer $< x$), and ξ is a random number uniformly distributed between zero and one.

Step 4: Increase c by one.

Step 5: Select the i^{th} fission site from the bank p times and also increase t by one for p times.

Step 6: If $t == N$, then exit the loop, otherwise continue.

This algorithm is based on the probability of all the possible ways of choosing N items from N' items such that t items occur in the first c. The probability is given by

$$\binom{N'-c-1}{N-t-1} \Big/ \binom{N'-c}{N-t} = \frac{N-t}{N'-c}, \tag{5.26}$$

where $0 < N \le N'$.

5.6 KDE Sampling of the Fission Bank

KDE is used to estimate the shape of the fission source distribution at the end of each batch by using the fission sites from the fission bank. Samples from the estimated source distribution are used as the starting particles for the next batch. It is expected that this method will free the particles from spatial inter-generational correlation, since samples from the estimated source distribution are used rather than the actual fission sites. The estimator $\hat{f}(x)$ in Eq. (2.8) is a pdf, normalized to 1, since the kernel function $\frac{1}{h}k\left(\frac{x-X_i}{h}\right)$ is normalized to 1. It is straightforward to sample from the estimator $\hat{f}(x)$.

Realizations Y from $\hat{f}(x)$ can be generated on-the-fly without constructing the whole density function as shown below [18]:

1. Choose I uniformly with replacement from $\{1,.....,N\}$ by $I = \xi N + 1$, where ξ is the random number between 0 and 1.
2. Generate sample ε from the kernel k (e.g., Box-Muller if k is Gaussian).
3. Set $Y = X_I + h\varepsilon$.

A multivariate version of the above sampling method is easily constructed and unequal bandwidths in the various coordinate directions can be accounted for using the corresponding transformation in step 3. As the product kernel estimator (see Eq. (2.47)) is used for the multivariate case, samples can be drawn individually in each dimension by using the above on-the-fly sampling scheme. The rescaled Epanechnikov kernel is used for all our simulations. Devroye and Gyorfi [19] proposed a very fast simulation from the rescaled Epanechnikov kernel

$$k(x) = \frac{3}{4}(1-x^2),\ |x| \le 1 \tag{5.27}$$

as follows:

1. Generate three uniform $[-1,1]$ random variates V_1, V_2, V_3.
2. If $|V_3| \ge |V_2|$ *and* $|V_3| \ge |V_1|$, *set* $\varepsilon = V_2$.
3. Otherwise set $\varepsilon = V_3$.

5.7 Boundary Correction

Generating realizations from the boundary-corrected KDE is not straight forward and may be expensive due to the need to employ a rejection sampling scheme. A density function for rejection sampling need to be constructed after each Monte Carlo generation and the procedure loses the advantage of on-the-fly sampling discussed earlier. This will be more serious in higher dimensions. It has been found that truncating the estimator at the boundaries and then renormalizing $\hat{f}(x)$ to unity is the simplest and the best boundary correction for KDE-based Monte Carlo fission source iteration. This can be easily achieved by repeating the on-the-fly scheme, discussed earlier, until the sampled point is within the user domain. This scheme, called "renormalization" sampling, is used to sample source locations from the previous generation, which are used as the starting source locations in the current generation.

5.8 Numerical Example 1 – Criticality of a Large 1-D Slab

Monte Carlo criticality calculations with KDE and the conventional fission source method are carried out for a 100 mfp (mean free path) wide multiplying slab. The slab contains homogeneous material with the following properties.

$$\Sigma_t = 1.000\, cm^{-1}$$
$$\Sigma_s = 0.500\, cm^{-1}$$
$$\Sigma_f = 0.1852\, cm^{-1}$$
$$\Sigma_c = 0.3148\, cm^{-1}$$
$$\nu = 2.70.$$

Isotropic scattering is assumed. With KDE, the starting neutrons for each cycle are sampled from the estimated source distribution of the previous cycle using renormalization sampling and the bandwidth is calculated by Eq. (2.21). Both simulations used $10,000$ particles per batch and 1500 batches, and the first 500 batches were discarded.

Figures 5.1 and 5.2 present the comparison between the conventional and KDE methods. The binned sources T^b, are calculated with 100 bins by averaging the source

distributions over the final 500 cycles. This averaged source is used for the relative entropy calculation [2]. It is clear from Figure 5.2 that the KDE method has improved source convergence.

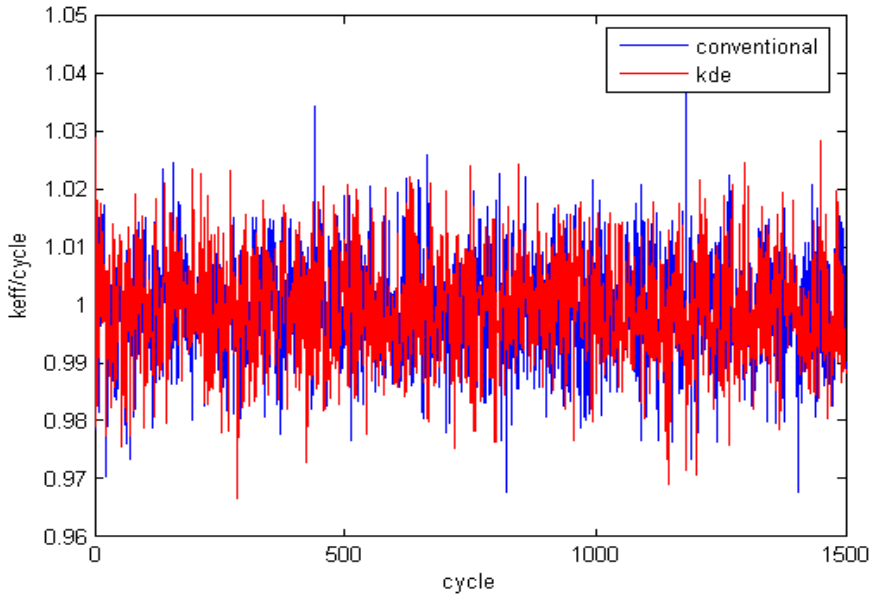

Figure 5.1 Cyclewise k for conventional and KDE method for numerical example 1.

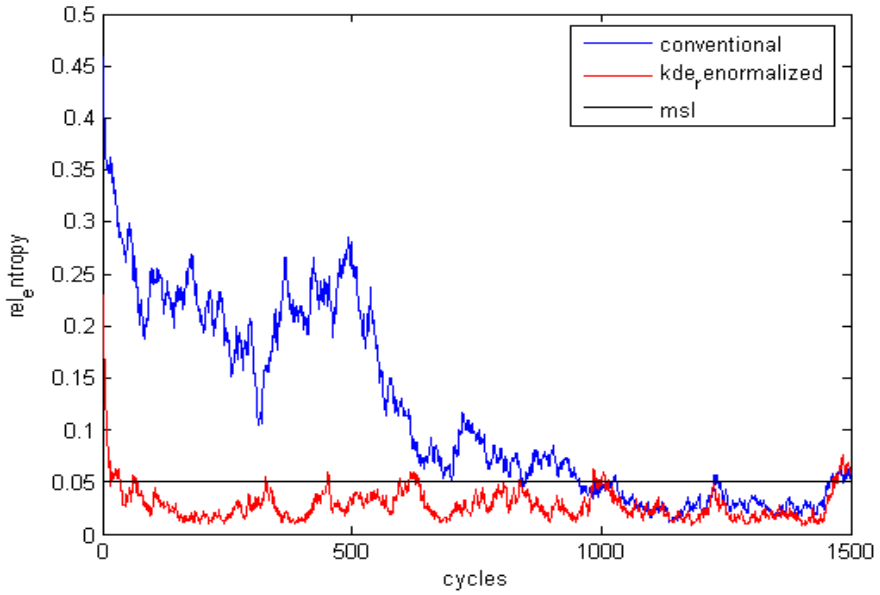

Figure 5.2 Posterior relative entropy comparison between conventional and KDE for numerical example 1.

5.9 Numerical Example 2 – Checkerboard

Figures 5.3 and 5.4 (not to scale) present a one group isotropic checkerboard problem which is a 3D version of the checkerboard problem used by Ueki and Nease [20]. Monte Carlo criticality calculations with KDE and the conventional fission source method are implemented for this case. With KDE, the bandwidth is calculated by Eq. (2.21). Both simulations used $20,000$ particles per batch and 1500 batches, and the first 500 batches were discarded. Figures 5.5 and 5.6 present the comparison between the conventional and KDE methods.

The binned sources (T^b) are calculated by averaging the source distributions over the final 500 cycles. 200 mesh tally regions (1 in the x-direction, 1 in the y-direction and 200 in z-direction) are used to obtain the binned source, which is used to calculate the source entropy. Only the z coordinates of the starting neutrons for each cycle are sampled from the estimated source distribution (z coordinates only) of the previous cycle using renormalization sampling scheme. It is clear from Figure 5.6 that the KDE method has improved source convergence.

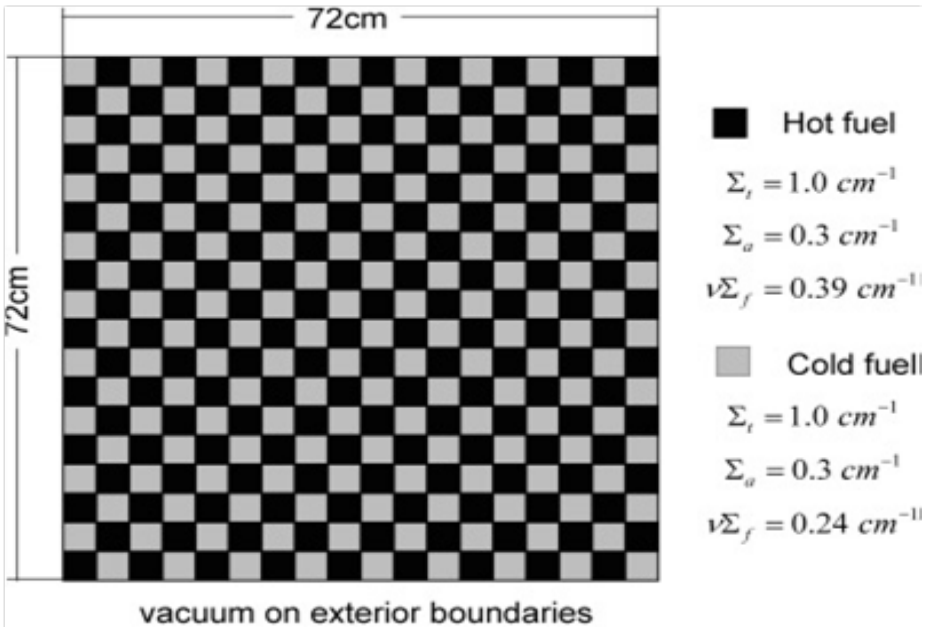

Figure 5.3 Top view of the one group 3-D isotropic checkerboard problem.

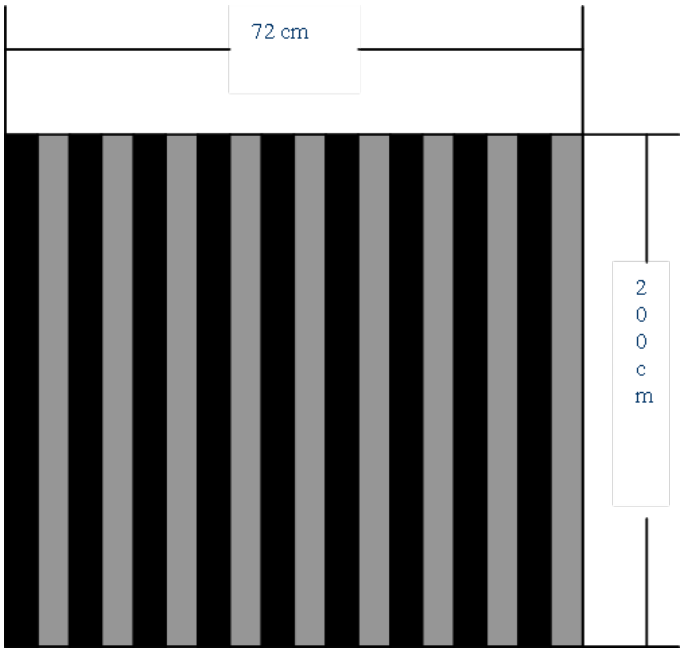

Figure 5.4 Side view of the one group 3-D isotropic checkerboard problem.

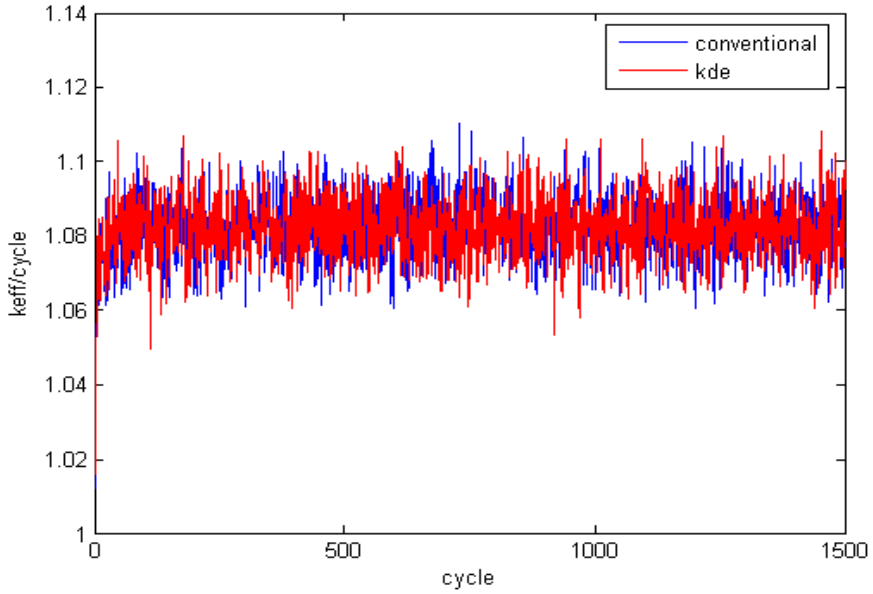

Figure 5.5 Cyclewise k for conventional and KDE method for numerical example 2.

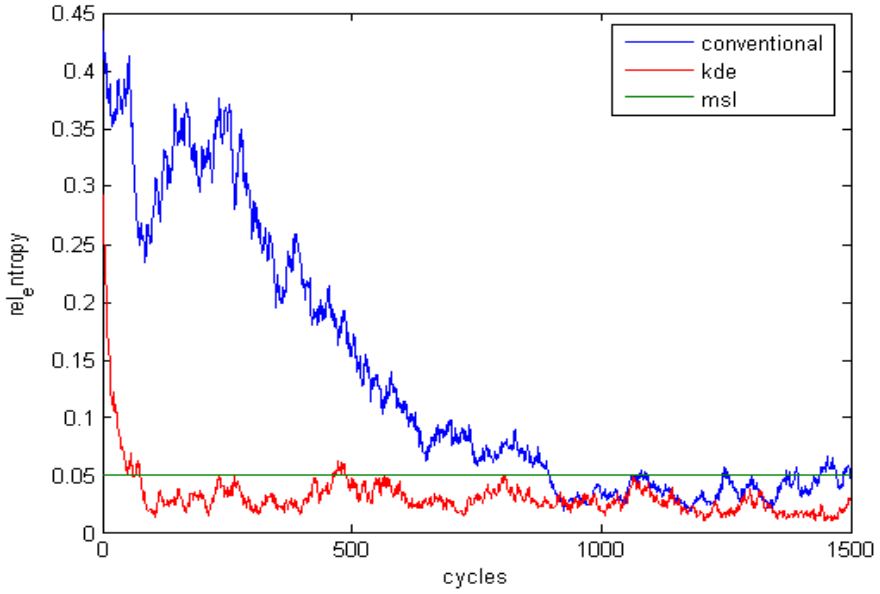

Figure 5.6 Posterior relative entropy comparison between conventional and KDE for numerical example 2.

5.10 Numerical Example 3 – Checkerboard

Monte Carlo criticality calculations with KDE and the conventional fission source method are implemented for the checkerboard case with renormalized KDE sampling in all 3 coordinate directions. With KDE, the bandwidth in each coordinate direction is

calculated by Eq. (2.21). Both simulations used $135,000$ particles per batch and 2000 batches, and the first 1000 batches were discarded. The binned sources (T^b) are calculated by averaging the source distributions over the final 500 cycles. 3200 mesh tally regions (4 in the *x*-direction, 4 in the *y*-direction and 200 in *z*-direction) are used to obtain the binned source for the source entropy calculation. The scaling factor σ in Eq. (2.21) (more specifically σ_x, σ_y, and σ_z for 3 coordinate directions) is calculated for each material cell (there are $18\times18\times1=324$ material cells) and therefore the bandwidths are also calculated for each material cell and in each coordinate direction, and N is the total number of samples (fission sites from previous batch) in Eq. (2.21). In this regard the bandwidths are local in this case instead of the global bandwidth used in example 2.

Only the exterior vacuum boundaries are considered for the renormalized boundary correction. Figures 5.7 and 5.8 present the comparison between the conventional and KDE methods. It is clear from Figure 5.8 that the KDE method has improved source convergence.

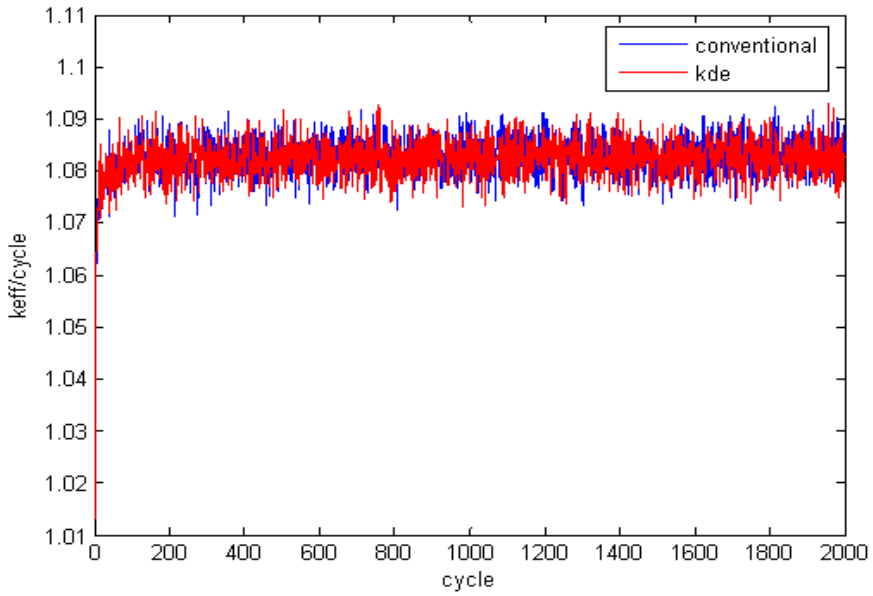

Figure 5.7 Cyclewise k for conventional and KDE method for numerical example 3.

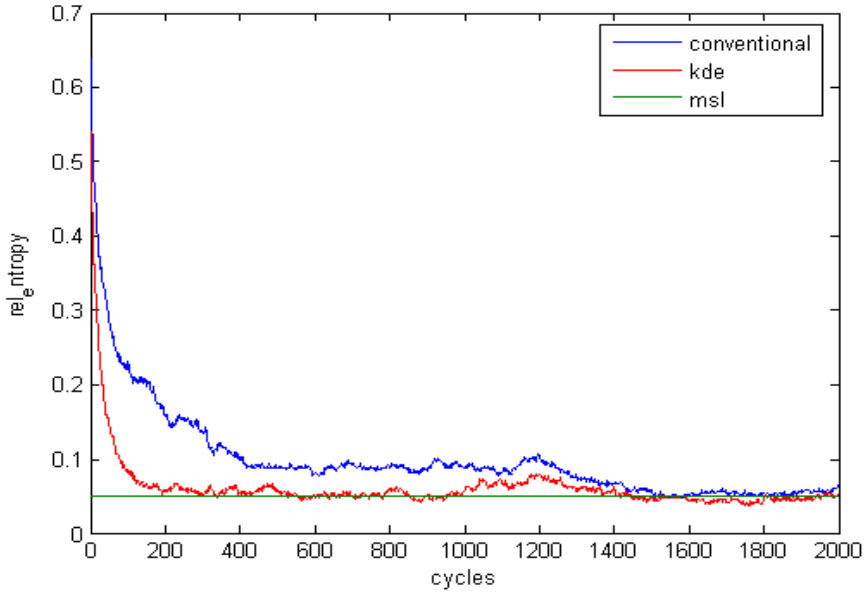

Figure 5.8 Posterior relative entropy comparison between conventional and KDE for numerical example 3.

5.10 Numerical Example 4 – Checkerboard

Monte Carlo criticality calculations with KDE and the conventional fission source method are carried out for the same checkerboard case with renormalized KDE sampling in all *3* coordinate directions. All the simulation parameters are same as those in example 3 except with KDE, the bandwidth in each coordinate direction is calculated by Eq. (2.59) (with $d = 3$), which is the case for multivariate KDE. Figures 5.9 and 5.10 present the comparison between the conventional and KDE methods. It is clear from Figure 5.10 that the KDE method has immensely improved source convergence.

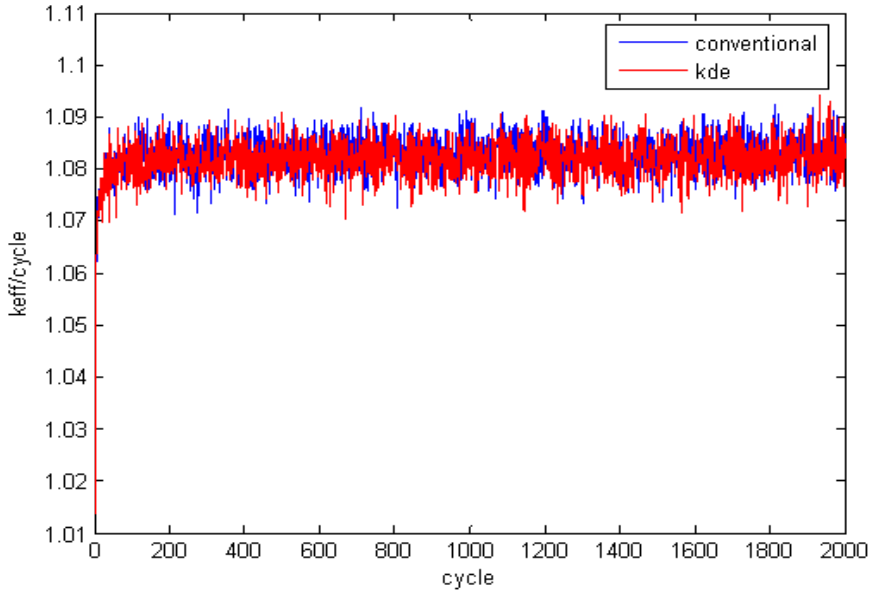

Figure 5.9 Cyclewise k for conventional and KDE method for numerical example 4.

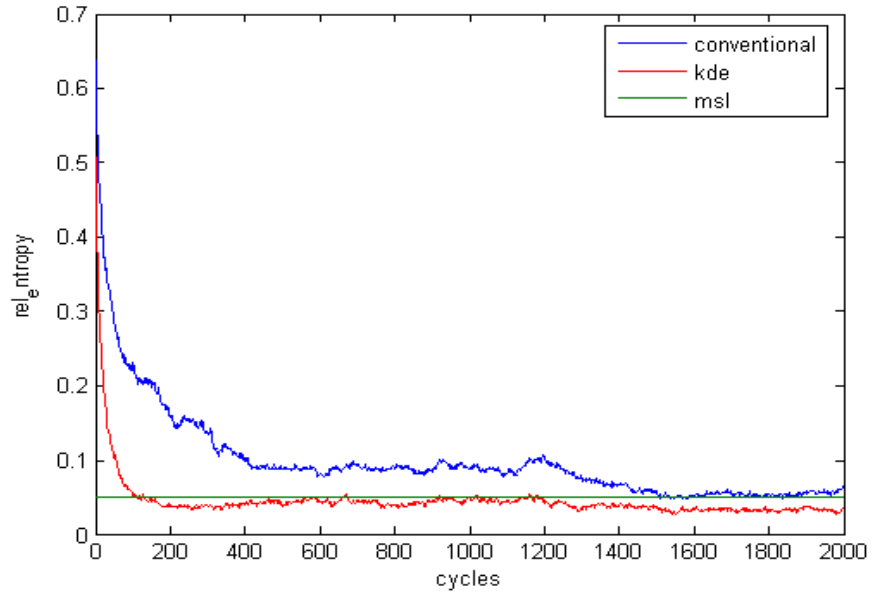

Figure 5.10 Posterior relative entropy comparison between conventional and KDE for numerical example 4.

5.11 Conclusions

The use of KDE to implement fission source iteration in criticality problems has been shown to be successful in simple 1D slab geometry and also in more complicated 3D checkerboard geometry. KDE results in substantially faster source convergence than conventional Monte Carlo fission source iteration. Faster source convergence occurs mainly in large homogeneous regions (axial direction in most nuclear power reactors). This is due to the comparatively large bandwidth in the large homogeneous region, which increases communication between neutrons in a large loosely coupled system. The value of the bandwidth is extremely important for the proposed KDE sampling of the fission source distribution. With increasing number of histories per batch, the relative advantage of faster convergence with KDE reduces compared to conventional fission source iteration. This is because for loosely coupled systems, with the increasing number of histories per batch, the neutron communication is improving for the conventional fission source iteration while the bandwidth is decreasing for KDE sampling. This is also evident from examples 3 and 4. In example 3, the 1D bandwidth formula (Eq. (2.21)) was used, and this is slightly smaller than the multi-D bandwidth given by Eq. (2.59). Due to the slightly larger bandwidth in example 4, the KDE source convergence is faster than that of example 3. As faster convergence mainly depends on the KDE bandwidth, which is not a physics-based parameter, this remains a topic for future research.

It is conjectured that KDE sampling of the fission source helps to minimize the intergenerational correlation in Monte Carlo k-eigenvalue calculation because KDE sampling uses fission sites sampled in the neighborhood of the previous cycle fission sites rather than the actual fission sites.

5.12 References

[1] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*, John Wiley & Sons (1976).

[2] T. Ueki and F. B. Brown, "Informatics Approach to Stationarity Diagnostics of the Monte Carlo Fission Source Distribution," *Trans. Am. Nucl. Soc.*, New Orleans, LA, Vol. **89**, pp. 458 (November 2003).

[3] G. E. Whitesides, "A Difficulty in Computing the k-effective of the World," *Trans. Am. Nucl. Soc*., **14**, pp. 680 (1968).

[4] R. E. Swaja, *Application of Stochastic Optimal Estimation Principles to Monte Carlo Eigenfunction Convergence*, Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, PA, October, 1972, recorded as 1973 in the reprint by University Microfilms International, Ann Arbor, MI (1993).

[5] T. J. Urbatsch, *Iterative Acceleration Methods for Monte Carlo and Deterministic Criticality Calculations*, Ph.D. Thesis, Nuclear Engineering and Radiological Sciences, University of Michigan (1995).

[6] T. Kuroishi, and Y. Nomura, "Development of Fission Source Acceleration Method for Slow Convergence in Criticality Analyses by Using Matrix Eigenvector Applicable to Spent Fuel Transport Cask with Axial Burnup Profile," *Journal of Nuclear Science and Technology*, **40**, pp. 433-440 (2003).

[7] T. Yamamoto, and Y. Miyoshi, "Reliable Method for Fission Source Convergence of Monte Carlo Criticality Calculation with Wielandt,s Method," *Journal of Nuclear Science and Technology*, **41**, pp. 99-107 (2004).

[8] T. Yamamoto, "Acceleration of Monte Carlo Solution by Conjugate Gradient Method," *Proc. Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications,* American Nuclear Society, Avignon, France (September 2005).

[9] J. P. Finch, J. S. Hendricks, and C. K. Choi, "Vacation Matrix Method for Correct Source Distribution in Monte Carlo Criticality Calculations," *The Monte Carlo Method: Versatility Unbounded in a Dynamic Computing World*, American Nuclear Society , Chattanooga, Tennessee, April 17-21, on CD ROM (2005).

[10] E. D. Larsen and J. Yang, "A Functional Monte Carlo Method for k-Eigenvalue Problems," *Nuclear Science and Engineering*, **159**, pp. 107-126 (2008).

[11] D. P. Griesheimer and B. E. Toth, "A Novel Source Convergence Acceleration Scheme for Monte Carlo Criticality Calculations, Part I: Theory, *Joint International Topical meeting on Mathematics and Computation and*

Supercomputing in Nuclear Applications, Monterey, American Nuclear Society, CA, April 15-19, on CD ROM (2007).

[12] D. P. Griesheimer, W. R. Martin, and J. P. Holloway, "A Functional Expansion Method for Monte Carlo Eigenvalue Calculations," *The Monte Carlo Method: Versatility Unbounded in a Dynamic Computing World*, American Nuclear Society, Chattanooga, Tennessee, April 17-21, on CD ROM (2005).

[13] K. Banerjee, W. R. Martin, "A Proposed Kernel Density Estimator Method for Monte Carlo Eigenvalue Calculations," *Proc. PHYSOR-'08*, Sept 14-19 Interlaken, Switzerland (2008).

[14] R. J. Brissenden and A. R. Garlick, "Biases in the Estimator of keff and its Error by Monte Carlo Methods," *Annals of Nuclear Energy*, 13, pp. 63-83 (1986).

[15] T. M. Sutton, "Analysis of Monte Carlo Eigenvalue Bias," (unpublished) (1991).

[16] F. B. Brown, "Fundamentals of Monte Carlo Particle Transport," LA-UR-05-4983, Los Alamos National Laboratory, http://mcnp.lanl.gov/publication/mcnp_publications.html, (2005).

[17] D. E. Knuth, *The Art of Computer Programming*, Vol. 2: Semi-numerical Algorithms, 3rd Edition, Addison-Wesley, Reading, MA (1998).

[18] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London (1986).

[19] L. Devroye and L. Györfi, *Nonparametric Density Estimation – The L_1 View*, John Wiley & Sons, New York (1985).

[20] T. Ueki and B. R. Nease, "Time Series Analysis of Monte Carlo Fission Source-II: Confidence Interval Estimation, *Nuclear Science and engineering*, **153**, pp. 184-191 (2006).

CHAPTER 6

Summary and Future Work

6.1 Global KDE Tally

Monte Carlo radiation transport method is very efficient for estimating integral quantities, such as the volume averaged flux inside a region, and the corresponding uncertainties. However, if the shapes or the distributions of some quantities are of interest, then the traditional method is to employ histogram estimators. Histograms can be viewed as probability density estimators. Other sophisticated statistical density estimators can also be used to estimate the distributions of the important nuclear engineering quantities. In this thesis, the Kernel Density Estimator is used to represent Monte Carlo tallies. Two new neutron flux estimators and their variances are developed, namely the KDE-collision and KDE-track-length estimators. These new estimators are capable of estimating the flux at any point within a given domain without any bin structure.

In Chapter 3, the advantages of the KDE flux tallies over the FET and histogram flux tallies have been shown with simple 1D geometries and also with more involved 2D geometries. KDE can be also used to depict other kind of Monte Carlo tallies with higher order shape information. The primary disadvantage of KDE for a criticality calculation is the requirement of storing all the points (e.g., collision points from each neutron trajectory) during a cycle for calculating the bandwidth (h) at the end of the cycle. That disadvantage has been successfully overcome by using the bandwidth from the previous batch for the criticality calculations. The bandwidth can be calculated online since it mainly depends on the standard deviation. So the remedy is to start calculating the bandwidth just before the active cycles begin as all the tallies are averaged only over the active cycles. This methodology works as the fission source is converged within statistical fluctuations before the active cycles begin.

A major advantage of the KDE tally is the elimination of internal bin structure to resolve the shape of the tally within a given region. KDE tallies have functional forms; therefore it is straightforward to estimate the flux at any point within the system without any binning. Also, we conjecture that the KDE path-length tally can be used with Woodcock tracking or delta tracking [1], which is a kind of rejection method for sampling distance to collision. Woodcock tracking allows one to ignore where the material boundaries are located while the neutron is being tracked through the geometry. It is very useful for problems where the total cross-section varies rapidly within the geometry. Woodcock tracking is also frequently used for tracking through reactor fuel assemblies, where the geometry is a regular lattice. As KDE tallies are mesh free, these tallies could be used with delta tracking to entirely eliminate the requirements of binning, both material and tally bins.

Though in the convergence analysis part of the KDE tally we assume that the underlying true density is such that its second derivative is continuous and square integrable, KDE tallies can estimate quantities within the domain with material discontinuities. The KDE tallies are truly global and not piecewise approximations, and excellent results were obtained for a 2-D fuel assembly using one global bandwidth and no boundary corrections for the internal material boundaries.

A convergence analysis indicates that KDE tallies compare favorably to the conventional histogram tallies as well as FET tallies. We have shown, both analytically and numerically, that KDE always converges faster than the conventional histogram tally and usually converges faster than the FET tally.

In summary, the KDE global flux tally is a substantial improvement over the conventional histogram tally, which is a step approximation of the true distribution. Not only does it provide a higher order approximation to the underlying tally, KDE is a mesh-free tally and can be implemented without a prescribed bin structure. Implementation of KDE tallies is very straightforward, and the computational cost is comparable with the traditional histogram tally, although storage costs may increase depending on the ultimate use of the KDE tally.

6.2 Monte Carlo Tallies with Unbounded Variance

Flux at a point and average flux over a surface are important nuclear engineering quantities, especially in shielding calculations. Monte Carlo point detector and surface crossing flux tallies estimate these quantities, but these tallies have singularities due to the divergence in space for the point detector tally and the divergence in angle with the surface flux tally. Due to these singularities, their variances are unbounded and cannot be used reliably to measure the convergence of these tallies to their true means. If the point detector is in a scattering or source region, the convergence of this tally is very slow or non-existent. Similarly, the surface crossing flux tally can be unreliable when particles cross the surface in grazing angles. The KDE methodology provides an alternative approach to both of these tallies that greatly improves over current techniques to handle the singularities. Based on the observation that KDE estimates the underlying PDFs for the particle interactions (i.e., collisions or surface crossings) that are scored for both tallies, variance reduction methods have been developed for both tallies that yield 1/N convergence for their variances, while maintaining unbiased (at least asymptotically) estimators.

The implementation of the variance reduction schemes for both estimators involves using the approximate (KDE-estimated) PDF for the domain near the singularity and allowing the width of this domain to vary. In essence, this parameter becomes a variance reduction tool.

6.3 KDE Fission Source Iteration

Monte Carlo criticality calculations sometimes suffer from slow fission source convergence. The slow fission source convergence occurs mainly in large loosely coupled systems with high dominance ratio. Dominance ratio is the ratio between the first and the fundamental eigenvalue of the system. A large number of inactive cycles may be required for this type of problem to converge the fission source distribution, which is particularly important if local information like pin power is required. KDE has been shown to speed up the fission source convergence. KDE is applied to sample fission

locations from the fission bank which are then used as the starting source locations for the next generation.

The use of KDE to implement fission source iteration in criticality problems has been shown to be successful in simple 1D slab geometry and also in more complicated 3D checkerboard geometry in Chapter 5. KDE results in substantially faster source convergence than conventional Monte Carlo fission source iteration. Faster source convergence occurs mainly in large homogeneous regions (axial direction in most nuclear power reactors). This is due to the comparatively large bandwidth in the large homogeneous region, which increases communication between neutrons in a large loosely coupled system. We found that the value of the bandwidth is extremely important for our proposed KDE sampling of the fission source distribution. We have seen with increasing number of histories per batch the advantage of faster convergence with KDE is decreasing compared to conventional fission source iteration. This is because for loosely coupled systems, with the increasing number of histories per batch, the neutron communication is improving for the conventional fission source iteration, while the bandwidth is decreasing for KDE sampling. As faster convergence mainly depends on the KDE bandwidth, we need to study this effect in the future.

We believe that KDE sampling of the fission source also helps to minimize the intergenerational correlation in Monte Carlo k-eigenvalue calculation, because KDE sampling uses neighboring source locations instead of the same source locations from the previous generation. This may reduce the bias in the variance introduced by the cycle to cycle correlation.

6.4 Future Work

We conclude this chapter with a brief summary of future work with KDE-based estimation methods for radiation transport Monte Carlo.

KDE-track-length tally is implemented by dividing each track into some arbitrary number (n) of sub-tracks. We used different n for different problems. It is quite obvious that large n yields a smoother flux distribution. However, excessively large n is

computationally burdensome. An investigation should be performed to determine the optimum number of sub-tracks. This should take into account the underlying material regions because the typical use of path-length estimators is to tally reaction rates, and the accuracy and uncertainty of these reaction rates may depend on the number of sub-tracks.

In two-dimensional fuel assembly problems, higher resolution could be obtained by using region-based bandwidths instead of one global bandwidth. We note here that the global bandwidth used for those problems yields a well-resolved flux distribution. However, it is expected that local bandwidths could perform better for the fuel assembly problems.

The practical use of KDE tallies for production Monte Carlo codes will depend on the accuracy and efficiency of the methodology compared to conventional tallies. A promising approach may be to implement KDE with mesh tallies rather than with conventional tallies that are scored during the random walk. Since mesh tallies are performed with a mesh that overlays the geometry of interest, and KDE can be implemented with no knowledge of the eventual tally mesh, this may be a natural path to incorporate KDE tallies into a production Monte Carlo code.

Conventional pathlength tallies cannot be implemented if Woodcock or delta tracking [1] is employed. For example, the production Monte Carlo code Serpent [2] uses Woodcock tracking but does not allow pathlength tallies. This is due to the fact that material boundaries are not known during the random walk when Woodcock tracking is used, so pathlength scores cannot be assigned to regions. However, since the KDE pathlength tally is a mesh-free tally, it can probably be implemented with Woodcock tracking.

A *m*th-order kernel k is the kernel whose first $(m-1)$ moments are zero and the *m*th moment is nonzero. The *m*th moment of kernel k is defined as

$$\mu_m(k) = \int x^m k(x)\,dx. \tag{6.1}$$

In this thesis, we employ a second order kernel, which itself is a probability density function. If k is constrained to be a probability density function, then $\mu_2 > 0$ is necessarily true. However, if we could relax this constraint, it is possible to construct k

with several vanishing moments. This will reduce the bias and improve the optimal convergence rate of MISE. We note here that for higher order kernels the resulting kernel density estimator will not be a normalized density function itself. We did not explore higher order kernels in this dissertation, but they should be investigated in future work.

In this dissertation, we numerically demonstrate that the KDE modified point detector and surface crossing flux tallies are unbiased. For a simple setting with a constant scoring function, it is straightforward to show that this KDE modification is unbiased. It is also obvious that the KDE modified tallies are asymptotically unbiased as KDE itself is asymptotically unbiased. However, it is an open question whether a finite sample is unbiased. That is, is the expectation of a finite sample of KDE-based observations equal to the true mean?

The conventional point detector flux tally was improved by employing a KDE-collision estimator. It may be possible to apply the KDE-track-length estimator and reduce the variance even further.

The interpolation scheme, which is successfully implemented for the point detector estimator, should be extended to the surface crossing flux tally also. The interpolation scheme is applied to estimate the KDE collision flux at any sampled points, from the uniform distribution, by using previously estimated KDE flux at few equispaced points. This method is faster than the exact implementation of the KDE modified point detector tally.

The renormalizing boundary correction is used for KDE fission source iteration, but this is not an order h^2 boundary correction. There remains room for improvement in this regard.

We conjecture that KDE sampling scheme may reduce the cycle to cycle correlation in the criticality calculations. This should also be investigated in future. Another future work could be the development of a mathematical analysis to support fission source acceleration with KDE.

Bandwidth determination and boundary correction methods are very important for all KDE-based Monte Carlo methods. This could be another potential area for further research. In this thesis, bandwidths are calculated from the observed data. Perhaps bandwidth calculation based on the physics and intrinsic statistics of the problem could be a challenging research area.

To handle more realistic problems, the KDE based Monte Carlo methods should be applied to continuous energy problems, including in the energy domain. There is no conceptual reason why this would not be successful.

6.5 References

[1] E.R. Woodcock, T. Murphy, P.J. Hemmings, T.C. Longworth, "Techniques Used in the GEM Code for Monte Carlo Neutronics Calculations in Reactors and Other Systems of Complex Geometry," *Proc. Conf. Applications of Computing Methods to Reactor Problems*, ANL-7050, pp. 557, Argonne National Laboratory (1965).

[2] J. Leppänen, "PSG2/Serpent – A Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code," *User's Manual*, VTT Technical Research Center Finland (2009).