1997

Stochastic Techniques for the Solution of Electrostatic Problems With Applications to Electron Optics.

Elias Kougianos
Louisiana State University and Agricultural & Mechanical College

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

STOCHASTIC TECHNIQUES FOR THE SOLUTION OF ELECTROSTATIC PROBLEMS WITH APPLICATIONS TO ELECTRON OPTICS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Electrical and Computer Engineering

by
Elias Kougianos
B.S., University of Patras, 1985
M.S., Louisiana State University, 1987
M.S., Louisiana State University, 1988
December 1997

Acknowledgments

I wish to thank Drs. A. H. Marshak and P. K. Ajmera for their moral and material support; Dr. W. Hamilton for being one of the very best teachers I ever had the opportunity to associate with, and Dr. Aravena not only for his support but also for his substantial help with the material presented in the appendices. I also wish to express my deep gratitude to Dr. M. Feldman. Without his advising, assistance and friendship, this work would have never materialized.

Table of Contents

List of Tables

List of Figures

Abstract

We apply stochastic techniques towards the solution of the two-dimensional Laplace's equation in boundary value problems encountered in the calculation of electrostatic potentials in electron lenses and deflectors. The justification of these techniques arises from an astonishingly simple but far-reaching principle, which has been known for a long time but has been rarely used: the potential at any point in the interior of a charge-free region can be calculated by performing random walks starting at this point and terminating at the boundary of the region – the potential is then the average of the potential boundary values (assumed known) over the random walks. By an optimal combination of the stochastic Monte-Carlo and deterministic Relaxation methods, we show the advantages and competitiveness of our hybrid Monte-Carlo-Relaxation (MCR) technique compared to the conventional numerical techniques used in the previously mentioned problem. In order to enhance the performance of our method, we investigate the convergence, speed and accuracy of MCR versus traditional techniques. We also develop optimized computational techniques that we believe increase MCR's appeal to problems not previously considered amenable to Monte-Carlo type simulations as well as demonstrate its applicability in problems that are intractable by traditional relaxation or analytical techniques. We use MCR to simulate electrostatic lenses and deflectors previously presented in the literature. Finally, we demonstrate the application of MCR towards the numerical solution of general elliptic problems in arbitrary domains and we present the generalization of the stochastic method to solve problems with space charge, namely Poisson's equation.

1 Introduction and Survey

We begin with a brief survey of electron optics, a major field of application of the techniques developed in this work, potential theory, and the apparently unrelated area of Brownian motion and random walks. Although we do not examine the underlying theory here in great detail, a task relegated to chapter 2, we do provide the main motivation for the particular approach we have chosen for our simulations.

1.1 Electron Optics

Unlike many other fields of applied science, electron optics has a rather precise moment of birth: in 1926, H. Busch [4] demonstrated for the first time the use of electron beams in image formation. The main object of study of the field is the properties of beams formed by free electrons which have been released from a source and which propagate in vacuum, under the influence of external static or time varying electromagnetic fields.

Why the study of such beams is important comes as no surprise when one considers some of the more common applications encountered today: cathode ray tubes, microwave tubes, particle accelerators, energy and mass spectrometers, scanning analytical instruments, high-energy beam technology, x-ray sources, scanning electron microscopes, and image converters. This partial list should provide a glimpse of the diverse areas where electron optics principles and technologies apply.

Furthermore, the application of electron beams to the fabrication of microcircuits guarantees a continued interest in the area that is expected to provide additional momentum to the field throughout the next decade. Traditional optical methods for lithography are hindered by diffraction limitations while the promise of x-ray processes has not fully materialized due to technological difficulties. Focused electron beams provide paths for maskless, computer controlled lithography processes, as well as conventional systems developed with masks.

For the types of applications we will study, an accurate description of the electron motions through electrostatic optical systems is required. This description typically proceeds in two stages: the value of the static electric field through which the electron beam propagates is first determined, and, subsequently, the trajectory of the electrons through this field is calculated. Usually, the beam is restricted to within a narrow range close to the optical axis of the system in order to prevent performance degradation due to lens aberrations. It is clear, therefore, that it is sufficient to calculate the field only near the optical axis (which is typically a curve of high symmetry of the device). Due to the nature of the problem, however, it is impossible to isolate the solution domain on or near the axis using traditional boundary value solution techniques: since the field on the axis is established by the imposition of external voltages on boundaries far from the axis, a solution must be obtained for the entire domain, regardless of whether such information is necessary or not. In this respect, as we will meticulously point out throughout this work, Monte Carlo simulations are superior in situations where the geometry does not allow an analytical evaluation of the Green's function (see chapter 2) for the problem.

The essence of the previous discussion is that the establishment of the properties of an electron optical device requires the knowledge of electrostatic fields through at least parts of the device, a task which generally falls in the field of potential theory.

1.2 Potential Theory

The origin of potential theory can be placed in the 18th century. Lagrange observed in 1773 that gravitational forces, as predicted by Newton's theory, can be derived instead in an elegant manner from a function which was named a "potential" by Gauss in 1840. It soon became evident that potential theory may also be considered as the study of the solutions of Laplace's equation:

$$\nabla^2 \phi = 0. \tag{1.1}$$

Equation (1.1) belongs to the class of partial differential equations known as *elliptic.* In a broad sense, it can be considered as describing the mathematics of equilibrium. As mentioned in the previous section, the determination of the fields along the optical axis of an electrostatic optical device is of paramount importance. Since this determination is accomplished most easily when the electrostatic potential in the device, satisfying (1.1) in the absence of space charge effects, is known, the connection of potential theory to electron optics becomes apparent.

Mathematicians call the study of solutions of (1.1) harmonic analysis and an enormous literature exists with potential theory and harmonic analysis as its subject. Surveys of the subject from the mathematician's point of view are provided by Constantinescu and Cornea [5] and Axler *et al.* [6], while Kellogg's classic [7] is unsurpassed in its clarity even today and is more accessible to physicists and engineers. More modern treatments are given by Morse and Feshbach [8], Moon and Spencer [9], Dautray and Lions [10] and, of course, Jackson [11].

From a solution point of view, (1.1) alone is not sufficient to specify uniquely a potential ϕ. Additional boundary conditions are required. Depending on the form of these conditions, the resulting problem is named as a *Dirichlet, Neumann* or *mixed* boundary value problem. If the value of the potential is known on the boundary, the problem is called a Dirichlet problem. If, on the other hand, the value of the derivative of the potential along the normal to the boundary is given, the problem is called a Neumann problem. Finally, if a combination of these boundary conditions is given, we have a mixed problem. Also depending on whether the solution is required in the interior or the exterior of a given domain, the problem is further classified as an *interior* or *exterior* boundary value problem. The traditional mathematical methods of solution can be very different for each type of problem and a more detailed presentation will be given in chapter 2.

1.3 Brownian Motion and Random Walks

Brownian Motion connects phenomenological macroscopic quantities to atomistic microscopic ones and appears not only in physics but also plays a prominent role in the biological, social and financial sciences [12]. The first report on the phenomenon was given in 1785 by the Dutch physician Jan Ingenhausz who was studying the random movement of powdered charcoal on an alcohol surface. It was not, however, until 1828 when Robert Brown published his observations on the highly irregular movement of pollen on a water surface that the phenomenon was provided with a solid experimental basis. Subsequently, it was named after Brown. Following his observations, many scientists attempted to explain this phenomenon. It was found that the motion of the particles was affected by heat, that it depends on the viscosity of the surrounding liquid medium and that lighter and finer particles move more rapidly. Perhaps the most interesting observation was that the motion of the particles was "memoryless", i.e. the past of the motion did not appear to have a bearing on its future. By the end of the 19th century, it was concluded that the irregular nature of Brownian Motion arises from the random collisions between the particles in the suspension and the molecules of the liquid medium.

Brownian motion was provided with a solid physical theoretical foundation with the work of Einstein [13, 14], Smoluchowski [15] and Langevin [16] in the early 1900's. Although Einstein's and Langevin's analysis focuses on opposing properties of the particles (the *position* for the former, the *velocity* for the latter), their result was identical: the mean-square displacement of the Brownian particle evolves linearly with time.

The rigorous mathematical theory of Brownian Motion was developed by P. Lévy [17, 18] and N. Wiener [19]. Wiener confirmed the results of Einstein and Langevin and proved that the path of the Brownian particle, although everywhere continuous, is nowhere differentiable. Intuitively, this result is appealing: since the particle cannot appear and disappear spontaneously, the path must be continuous; its direction changes erratically all the time, so its path must be all corners, hence it is nowhere differentiable.

Brownian Motion may be approximated by a *Random Walk* [20, 21, 22] or, as is sometimes referred as, a "drunkard's walk". We will give a mathematical definition of the random walk in chapter 2. For the present we will mention that random walks appear in conjunction with many varied physical phenomena such as diffusion [23], the configurations of a polymer chain [24, 25, 26, 27], critical phenomena [28, 29], problems arising in solid-state physics [30, 31], field theory [32, 33], electric networks [34, 35, 36], fluctuation of the macroscopic state of proteins [37], the dynamical properties of complex systems [38] and the study of vibrating membranes [39, 40], just to name a few. Last but not least, random walks can be applied to solve potential problems (see section 1.4), a subject whose study is the main object of this work.

Before concluding this section, we should mention that although we will mainly concentrate on *simple* random walks (with one exception, in the solution of the general elliptic problem), in *random* media, other types of random walks [41, 22, 42, 43] are possible but beyond the scope of this work.

1.4 Random Walks and Potential Theory

In a 1928 paper Courant, Friedrichs and Lewy [3] made a connection between random walks and potential theory for the first time. This connection at first appeared to be a rather arbitrary consequence of the similarity in functional form of the discretized Laplace's equation on one hand and the mathematical formulation of a random walk on the other, and provides an algorithm for the solution of Laplace's equation (more specifically the interior Dirichlet problem): the potential at any point in the interior of a charge-free region can be calculated by performing random walks starting at this point and terminating at the boundary of the region – the potential is then the average of the potential boundary values (assumed known) over the random walks. This similarity, however, goes much deeper than this, as potential theorists were to observe with, to use Doob's expression [44], "jaundiced eyes". In a very real sense, the mathematics of potential theory and random

walks, or more generally Brownian motion, are not only interrelated but manifestations of the same structure.

One only needs to consult Doob's [44] massive treatise to appreciate the extent to which the two disciplines complement each other. An interesting and important consequence is that results which are extremely difficult to prove in one theory, are almost trivial in the other and vise versa [45]. Even more importantly, however, probabilistic formulations of potential problems are of a nature that is easy to calculate via digital computer. It is this aspect of the theory, more fully presented in chapter 2, that we will make extensive use of.

1.5 The Current Status of Monte-Carlo Simulations

Although, as noted, the idea of using random walk techniques in connection with potential problems is well known, practical applications of the method are astonishingly scarce. Certainly we are not aware of *any* use of these techniques to solve electrostatic field problems arising in the analysis of electron optical devices. Even when applied, there has not been, to the best of our knowledge, any consistent effort to optimize the algorithms, a principal result of the present work.

Algorithmically, random walks are well suited to the Monte Carlo method. Originally proposed by Metropolis *et al.* in 1953 [46], the Monte Carlo method has been successfully applied to a large variety of problems in physics and engineering [47, 48, 49, 50]. In general, the method is applicable to the class of problems where a formal equivalence between the behavior of the system under study and the expected behavior of a stochastic system, can be established. In our problem, the equivalence is established, as we will see, via the correspondence of a path integral formulation of Laplace's equation and the Monte Carlo evaluation of this integral.

After the appearance of [3], sporadic applications of the method were presented in the literature. Kac [51] examined its applicability to a wide range of integral and differential equations, while Curtiss [52] followed the same route as [3] towards the solution of

difference equations. Explicit methods for the Dirichlet problem are given by Yowell [53], Muller [54] and Wasow [55].

An examination of von Plato's otherwise comprehensive history of modern probability theory [56] reveals a complete absence of probabilistic potential theory, much less any mention of any specific applications, a state of affairs indicative of the overall situation. The same observation applies to van Kampen's classic reference [57]. A number of modern textbooks on the numerical solution of electrostatic problems [58, 59, 60, 61] do not even mention the Monte Carlo method. On the other hand, the 1963 edition of Binns' and Lawrenson's text [62] devotes an entire chapter to the method, which, however, in the 1973 edition [63] was relegated to an appendix. Finally, in the revised and expanded 1993 edition [64], the subject was entirely dropped! More recently, we note a hardware-oriented paper by Sadeh and Franklin [65], and applications of the method to problems in fluid dynamics [66] and heat conduction [67, 68].

As far as the field of electron optics is concerned, a literature search failed to produce any applications of the Monte Carlo method to the solution of Laplace's equation in electron focusing devices. Furthermore, the standard textbooks on the field, [69] and [70, 71, 72] do not mention this technique.

It is clear, in our opinion, that the applicability of the Monte Carlo method to these problems should be evaluated and optimization techniques be developed for its application. It is the main purpose of this work to do so.

1.6 Organization of Presentation

A basic presentation of the theory, both deterministic and stochastic, required for enumerating and classifying the methods of solution of Laplace's equation, is given in Chapter 2. We will also review the most common numerical methods of solution. Additionally, we will describe in detail the main theorem upon which this work is based. This theorem provides the framework on which our stochastic algorithms rely. The necessary results

from classical electron optics theory will also be given. The general elliptic problem, of which equation (1.1) is but one special case, is defined and work on its stochastic solution is reviewed.

Chapter 3 analyzes, in a detailed manner, the challenges presented by the implementation via digital computer of the stochastic algorithms on which this work relies. These algorithms present inherent problems, not the least of which is the reliable generation of long sequences of random numbers. We therefore present the results of our evaluation of several commonly used random number generators. The other major obstacle of stochastic simulations, long execution times, is then addressed. We present detailed, intelligent algorithms that will allow us to significantly reduce execution times by taking advantage of symmetry as well as particular features of harmonic functions and which will minimize the execution time required to obtain a given accuracy. We examine both one- and two-dimensional cases. Although the practical applications of our work are expected to be on two-dimensional geometries, we will show that a wealth of information can be extracted from one-dimensional model calculations which have the added advantage of quick execution time and ease of interpretation. At this point, we will present our principal algorithm and justify it theoretically.

Chapters 4 and 5 comprise the culmination of our work by presenting results of numerical experiments in which our algorithms are tested. We present an exhaustive analysis of one-dimensional prototype simulations and use the results as guides for more intensive (and more interesting) two-dimensional simulations. We demonstrate the viability of our method by solving Laplace's equation stochastically in non-trivial geometries and compare our results and execution times with traditional approaches. Where applicable, we compare with analytically known solutions. Finally, we synthesize our work by applying it to the simulation of actual electrostatic deflectors and lenses which have been presented in the literature.

Chapter 6 summarizes our work, clearly indicates points of departure from traditional techniques as well as the novelty of our approach and presents our estimation of the advantages and disadvantages of our algorithms.

A straightforward proof of the Main Theorem is given in appendix A while the theorem is extended to general linear elliptic problems in appendix B.

2 Basic Theory

A basic presentation of the theory, both deterministic and stochastic, required for enumerating and classifying the methods of solution of Laplace's equation, is given in this chapter. Additionally, we will discuss the main theorem on which this work is based. This theorem provides the framework on which our stochastic algorithms rely. The general elliptic problem, of which equation (1.1) is but one special case, is defined and work on its stochastic solution is reviewed.

2.1 The Theory of Brownian Motion and Random Walks

Deterministically, many physical systems of interest are described by an evolution equation of the form:

$$\frac{dX(t)}{dt} = A(X(t), t), \tag{2.1}$$

where $X(t)$ is a relevant physical quantity of the system and A is a smooth function describing the dynamic behavior of the system. Equation (2.1) is also expressed in the suggestive form:

$$X(t + dt) = X(t) + A(X(t), t)dt. \tag{2.2}$$

Evolution equations of this type describe physical processes that are "deterministic", due to the unequivocal nature of (2.2), "continuous", due to the fact that $X(t + dt) \to X(t)$ as $dt \to 0$ and "memoryless", due to the fact that values of X *before* t are not required to calculate X at $t + dt$. Hence X is a *continuous memoryless deterministic* process.

Consider now the case where X is not a deterministic process, but rather of a stochastic nature, implying that we cannot predict what the value $X(t + dt)$ is, but, we can instead assign definite probabilities for every admissible value of $X(t + dt)$. We still require X to be continuous and memoryless, or, as commonly called, a continuous *Markov* process.

The question is now, what is the appropriate evolution equation, corresponding to (2.2) when X is stochastic rather than deterministic? It can be shown [73, 74] that the answer is provided by the *Langevin equation* for X:

$$X(t+dt) = X(t) + A(X(t),t)dt + D^{1/2}(X(t),t)N(t)(dt)^{1/2}, \tag{2.3}$$

where $X(t)$ is a continuous Markov process, $A(X(t),t)$ and $D(X(t),t)$ are smooth functions of their arguments and are called, respectively, *drift* and *diffusion* functions of the process, and $N(t)$ is a temporally uncorrelated unit normal random variable. An equivalent representation of (2.3) is:

$$\frac{dX(t)}{dt} = A(X(t),t) + D^{1/2}(X(t),t)\Gamma(t), \tag{2.4}$$

where $\Gamma(t)$ is a *Gaussian white noise* process. Processes described by (2.4) are of quite general nature. One particular case occurs when the drift and diffusion functions are given as $A(X(t),t) = -(1/\tau)X(t)$ and $D(X(t),t) = c^{1/2}$, with τ and c positive constants:

$$\frac{dX(t)}{dt} = -\frac{1}{\tau}X(t) + c^{1/2}\Gamma(t). \tag{2.5}$$

Equation (2.5) describes the continuous Markov process known as *Brownian Motion*. The simplest mathematical model of linear (one-dimensional) Brownian motion is the *Simple Symmetric Random Walk*[22, 20]: a particle starts at $x = 0$, at $t = 0$ and moves one unit to the left with probability $1/2$ or one unit to the right, again with probability $1/2$ during one time unit. In the next time step it moves again one step to the left or to the right with equal probabilities, *independently of its previous position*. The generalization to more than one dimensions is straightforward: if the number of dimensions is d, the particle has $2d$ points to which it can move at any time step. The probability of moving to any particular point in this case is $1/2d$.

2.2 Random Walks and Potential Theory

In view of the previous discussion, an interesting quantity that arises in the context of random walks is $P_n(x, y)$, the probability that the particle starting at x will be at y after exactly n steps. Going a step further, we inquire what is the probability $P(x, y)$ that the particle will pass y during the walk if it starts at x. Because of the statistical independence of each step (Markovian property), we can write:

$$P(x, y) = \sum_{n=0}^{\infty} P_n(x, y). \tag{2.6}$$

Quite often, $P(x, y)$ is called the *transition function* of the random walk and is sufficient to completely specify the walk.

We now turn to potential theory. As mentioned in chapter 1, our goal is to determine the potential distribution in certain areas of electron optical devices. This potential distribution, say $\phi(\mathbf{p})$, where $\mathbf{p}$ is a position vector, in the cases that we will consider, is the solution of the *interior Dirichlet problem*:

$$\nabla^2 \phi = 0. \tag{2.7}$$

Here it is assumed that the solution is desired in an interior domain Ω, figure 2.1, bounded by a continuous curve $\partial\Omega$, called the *boundary* of Ω. On the boundary, ϕ takes specified, known values, say $g(\mathbf{q})$, $\mathbf{q} \in \partial\Omega$.

There are many ways in which the interior Dirichlet problem can be attacked [9, 11, 75, 7, 76]. The most appropriate, for our purposes, is the integral *formula* representation given in terms of the Green's function:

$$\phi(\mathbf{p}) = -\int_{\partial\Omega} \phi(\mathbf{q}) \frac{\partial G(\mathbf{p}, \mathbf{q})}{\partial \mathbf{q}} \cdot d\mathbf{q}, \tag{2.8}$$

where $G(\mathbf{p}, \mathbf{q})$ is the Green's function for the problem. G is independent of the boundary

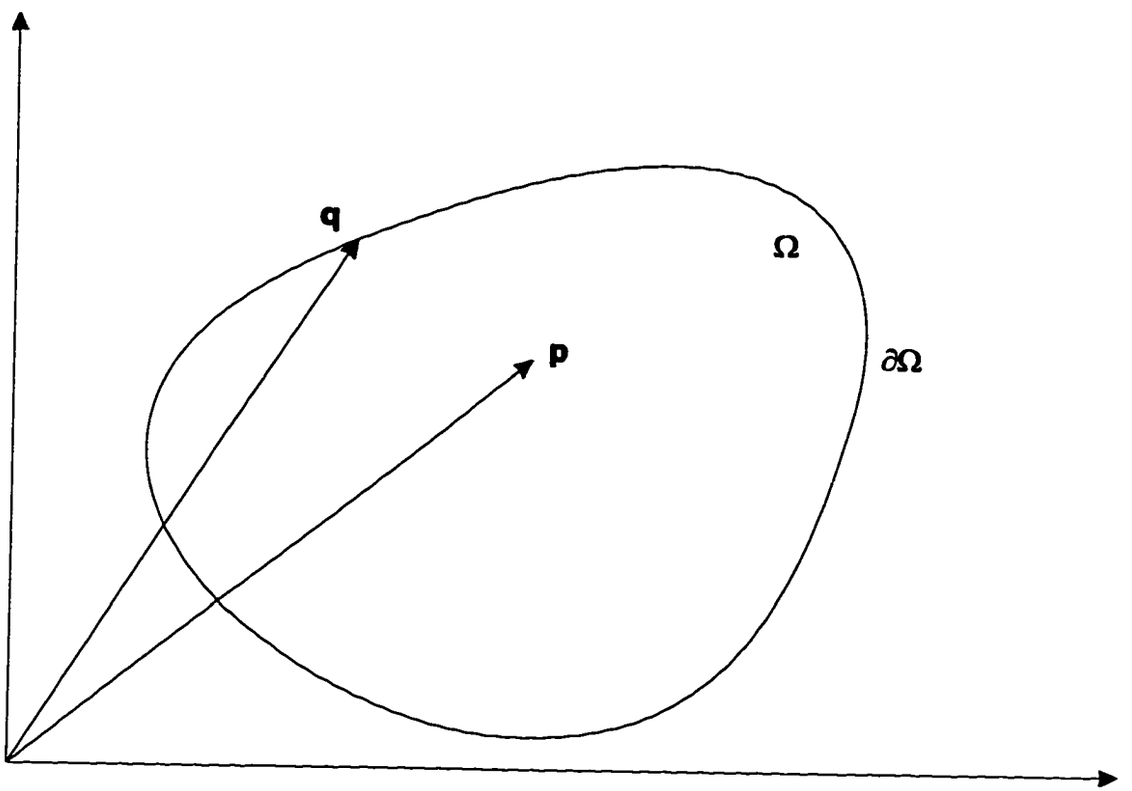

Figure 2.1: The solution domain for the interior Dirichlet problem.

conditions and depends entirely upon the geometry of the domain. G can be calculated from first principles or can be found in tabulated form [77].

The main result upon which this work relies may now be formulated:

Main Theorem. *The solution to the interior Dirichlet problem* $\nabla^2\phi(\mathbf{p}) = 0$ *for* $\mathbf{p} \in \Omega$, *with* $\phi(\mathbf{q}) = g(\mathbf{q})$ *for* $\mathbf{q} \in \partial\Omega$, *is given by:*

$$\phi(\mathbf{p}) = \mathbf{E}_{\mathbf{p}}\left[g(\mathbf{q}_\tau)\right], \tag{2.9}$$

where $\mathbf{E}_{\mathbf{p}}\left[g(\mathbf{q}_\tau)\right]$ *is the expectation value for the function* g *of a random walk that starts at* $\mathbf{p}$ *and terminates when it reaches the boundary* $\partial\Omega$ *at time* τ *and at point* $\mathbf{q}$.

We provide a proof of the Main Theorem in appendix A along with some important extensions. An alternative proof is also given by Durrett [78].

In practical terms, this theorem provides us with a simple algorithm to determine the potential at any given point $\mathbf{p} \in \Omega$:

- Start a random walk from $\mathbf{p}$.
- When the random walk reaches the boundary $\partial\Omega$ at point $\mathbf{q}$, it terminates and a new random walk is started from $\mathbf{p}$. Keep track of the value $g(\mathbf{q})$.
- After a large number of random walks, the potential at $\mathbf{p}$ is the average of $g(\mathbf{q})$.

This algorithm forms the basis of the Monte Carlo solution of Laplace's equation, or, more specifically, the interior Dirichlet problem. We also show in this work how to extend the algorithm to solve a general class of partial differential equations, including Poisson's equation.

2.3 The General Linear Elliptic Problem

The Laplacian (∇^2) that appears in Laplace's and Poisson's equations is the simplest case of a *general linear elliptic operator of order 2*, $L[\cdot]$, defined as:

$$L[\phi] = \sum_{i,j=0}^{2} a_{ij} \frac{\partial^2 \phi}{\partial x_i \partial x_j}, \tag{2.10}$$

where the coefficients a_{ij} are functions of the coordinates $(x_1, x_2) = (x, y)$ and satisfy the *ellipticity condition*:

$$\mathbf{x}^T A \mathbf{x} \geq 0 \qquad \text{for} \qquad \mathbf{x} \neq 0, \tag{2.11}$$

with A being the matrix (a_{ij}) and $\mathbf{x}$ the vector $[x, y]^T$, with the superscript T denoting matrix transposition.

The corresponding general linear elliptic *equation* of order 2 is given by:

$$L[\phi(x, y)] = -f(x, y), \tag{2.12}$$

where, as usual, $\phi(x, y)$ is an unknown function defined in a domain Ω, satisfying Dirichlet, Neumann or mixed boundary conditions on the boundary $\partial\Omega$ of the domain, and $f(x, y)$ is a known function in $\bar{\Omega} = \Omega \cup \partial\Omega$.

Numerical solution methods of (2.12) include the traditional deterministic techniques used in the solution of Laplace's equation (Pironneau [79] provides the necessary Finite Difference, Finite Element and Boundary Element formulations) as well as Monte Carlo methods. Booth [80, 81] has considered Monte Carlo solutions of the Dirichlet problem with constant coefficients for the restricted equation of the form:

$$\nabla^2 \phi = -\alpha^2 \phi, \qquad \alpha^2 = \text{const.}, \tag{2.13}$$

while Vrbik [82] considered the same problem from the point of view of a Fokker-Planck equation [57] and obtained solutions of only linear or, at most, quadratic accuracy in the domain step. Furthermore his method is not sufficiently generalizable to higher order accuracy, both theoretically and numerically due to the awkward matrix approach used. Booth's work, on the other hand, is not directly applicable to discretized domains and requires special transformations in order to handle cases more general than (2.13). These transformations are usually far from trivial and could destroy any symmetry that the domain possesses, a property of which we make extensive use in this work. It is far more preferable, from a computational point of view, to work on an original discretization grid of the domain than to transform the grid and move back and forth between the original and transformed domain. Not only does this avoid unwanted numerical errors, but it also maintains the Monte Carlo method's inherent simplicity. Lastly, it is not clear how Vrbik's work can be extended to Neumann or mixed problems.

We will show that it is possible to associate a *non-simple* random walk, where the probabilities of each outcome are no longer all equal, with the discrete approximation of the linear differential operator (2.10), to any desired accuracy with respect to the domain step. The transition function of this non-simple random walk can be readily derived once the discrete approximation has been obtained. Using this approach, it is then possible to solve the Dirichlet, Neumann or mixed problem (2.12) via the MCR technique.

2.4 Deterministic Numerical Methods

When solving Laplace's equation numerically, via digital computer, several approximations and therefore sources of error are introduced. Because of the manner in which computers operate, it is necessary to convert the physical problem to an equivalent discrete idealization. Some of the errors are due to:

- The incorporation of approximations by the differential equations describing the problem. We will not be concerned with this type of error in this work.

- The conversion of the continuous differential equation to a discrete approximation suitable for solution by computer. The error of this type is known as a *discretization* error and is unavoidable. It is, however, possible to reduce this error to any desired accuracy, provided sufficient terms in the discretization process are maintained.

- The discretization process converts the original problem to an algebraic one, where iterative methods are typically used to obtain the final result. Since the iterations must eventually be terminated, a *truncation* error is introduced.

In the Monte Carlo method, the discretization error is the same as in the deterministic methods, arising from the approximation of the Laplacian with a finite-differences expression, and the truncation error originates from the fact that only a finite number of random walks can be performed.

Perhaps the most fundamental component of any type of numerical solution of Laplace's equation is the *grid* which divides the problem domain into a finite number of subdomains, consisting of nodes, finite elements, control volumes etc.

Grids are broadly categorized as *structured* and *unstructured*. In a structured grid, the defining lines are curves on which one of the coordinates is constant. Therefore each grid point is uniquely defined by a set of grid indices. In an unstructured grid the layout of the nodes is essentially arbitrary and the grid lines, where identifiable, do not form coordinate surfaces. The choice of the grid points affects the speed and convergence of the solution very strongly. In this work we will use exclusively structured grids.

Once the grid has been identified, the Laplacian must be approximated on this grid via a *discretization* process. Depending on this process and the resulting set of algebraic equations and their method of solution, many different techniques arise, such as the finite difference method (FDM), the finite element method (FEM) [61, 83, 84, 60] and the boundary element method (BEM)[85, 58, 86]. The FDM was introduced by H. Liebmann as early as 1918 and its mathematical theory has been studied extensively, for instance by Forsythe and Wasow [87], Varga [88], Ames [89], Jacobs [90] and Vesely [91]. This is the discretization

method of choice for direct methods of solution of Laplace's equation and is the method that we will use exclusively in this work.

2.5 Summary

The fundamental connection between random walks and Brownian motion on one hand and potential theory on the other, was established in this chapter via the main theorem. This approach forms the framework on which the remainder of this work is based.

3 Computational and Theoretical Issues

In this chapter we analyze, in a systematic manner, the challenges presented by the implementation via digital computer of the stochastic algorithms on which this work relies. These algorithms present inherent problems, not the least of which is the reliable generation of long sequences of random numbers. We therefore evaluate several different random number generators, using standard statistical methods. The other major obstacle of stochastic simulations, long execution times, is then addressed. We develop detailed, intelligent algorithms that will allow us to significantly reduce execution times by taking advantage of symmetry, as well as particular features of harmonic functions. Although the practical applications of our work are on two-dimensional geometries, we show that a wealth of information can be extracted from one-dimensional model calculations which have the added advantage of quick execution time and ease of interpretation. The same results also apply in three-dimensional geometries. At this point, we are ready to present our proposed algorithms and justify them theoretically.

3.1 General Characteristics of Monte-Carlo Simulations

As mentioned in chapter 1, the key feature of Monte Carlo methods is the identification of the system under consideration with an equivalent, statistically sampled stochastic system. It is of paramount importance to obtain a large, statistically independent set of data points that are free of undesirable correlations. The minimization of statistical fluctuations dictates the need for very large numbers of samples. In the course of this work, it is not unusual to execute as many as 10^7 random walks per point. These considerations impose a heavy toll on CPU time.

In order to reduce execution times, we implement our algorithms in an intelligent manner so as to reduce, as much as possible, "house-keeping" type of computations. This is in addition to problem-specific optimizations which we present later.

In Monte Carlo simulations, an inordinate amount of time is spent on the generation of random numbers via computer, a process which is as close as we can come to implementing the stochastic nature of the sample system. In general, the calls to Random Number Generators (RNG's) are very expensive from a computational point of view. If we keep in mind that all we really need is a random *bit* for the one-dimensional case, or, at most, two random bits for a simple two-dimensional random walk, it is obvious that significant savings can be achieved if we break down the returned number from a RNG to its constituent bits, and use these instead. The savings are significant; in the CPU used in the course of the simulations in this work (INTEL Pentium $x86$ at 166 MHz), a long integer, which is what the RNG returns, is represented by 32 bits. Hence, with one call we obtain 32 or 16 (for the two-dimensional case) random steps. Of course this assumes that our RNG is reliable, an issue addressed in the next section.

Furthermore, a significant cost is usually spent on conditional statement evaluations. An example arises when a determination is made whether the random walk should proceed left or right, up or down, for example. In our algorithms we replace this conditional procedure with simple indexing: the current position of the random walk is increased or decreased, via *bitwise* operations, using the value of the random bit(s) directly. Since these operations are executed much faster than conditional evaluations, non-trivial savings in time arise.

3.2 Random Number Generators

From the preceding discussion it is apparent that the generation of random numbers that are *truly* random is of paramount importance for the success of any Monte Carlo simulation. Indeed, we obtained results (chapter 4) that were inconsistent with theoretical expectations precisely because of the presence of unexpected correlations in the RNG that was used.

3.2.1 Linear Congruence RNG's

The usual method preferred by most RNG's is the so-called "linear congruence method" [92]. Unfortunately, this method suffers from unwanted correlations in the low-order bits of the numbers produced [93, 94]. A random integer U_{i+1} is produced from another random integer U_i through the operation:

$$U_{i+1} = (aU_i + c) \mod m, \tag{3.1}$$

where the multiplier a and increment c are positive integers, smaller than m which is also an integer. Subsequently, the integer U_{i+1} is converted into a real random number R_{i+1} in the interval [0, 1] via:

$$R_{i+1} = \frac{U_{i+1}}{m}. \tag{3.2}$$

The random sequence is started by inputting a random integer U_0, known as the *seed* and subsequent integers are obtained from (3.1). Since there is nothing stochastic about (3.1), it is clear that the same seed will produce the same sequence of "random" numbers.

The most common variant of (3.1) found on 32-bit CPU's, such as the one used in this work, is:

$$U_{i+1} = \frac{aU_i + c}{d} \mod m, \tag{3.3}$$

with

$$m = 32678 \qquad a = 1103515245 \qquad c = 12345 \qquad d = 65536. \tag{3.4}$$

One shortcoming of the linear congruence method is the sequential nature of the random integers generated, a fact that introduces strong correlations between successive

random numbers. A straightforward way of alleviating this tendency is to simply shuffle the random sequence, after generating a large number of integers. The shuffling is achieved by generating a second sequence of discrete random integers which is used as the index variable for the original sequence.

3.2.2 Subtractive or Fibonacci RNG's

An improvement to the method was given by Knuth [93], in which a random number is generated from integers far removed from it in the sequence, reminiscent of the way Fibonacci sequences are generated:

$$U_{i+1} = (U_{i-55} - U_{i-24}) \mod m \qquad \text{for} \quad m \geq 55. \tag{3.5}$$

The large separation between the numbers used to generate successive terms removes any correlations due to proximity of terms and provides random numbers different than the linear congruence method, not just shuffled random numbers.

3.2.3 Tests of Randomness

It is very difficult to design comprehensive tests for randomness that are universally applicable. There are, however, standard statistical tests that can be used to isolate the most efficient RNG for a given situation.

One such test is the frequency or χ^2 test. The interval $[0, 1]$ is divided in a number of "bins", N_{bin}. If the random sequence is distributed randomly, the number of random samples in each bin should be approximately equal. The width of every bin is then $w = 1/N_{bin}$. If we then generate n random numbers, we expect nw samples per bin, on the average. A standard χ^2-test [95] can then be applied; if the actual number of samples in the j-th bin is M_j, the statistical χ^2 coefficient for this distribution is given by the following

expression:

$$\chi^2 = \sum_{j=1}^{N_{bin}} \frac{(M_j - nw)^2}{nw}. \tag{3.6}$$

The expected value for χ^2 should be equal to the number of degrees of freedom, $\nu = N_{bin} - 1$.

Another test is to simply look for *serial* correlations in a sequence of random integers $U_1, U_2, \ldots, U_n$, i.e. the consistent succession of large and small (or large) random numbers by defining a correlation coefficient [96]:

$$c = \frac{n(\sum_{i=1}^{n} U_{i-1}U_i) - (\sum_{i=1}^{n} U_i)^2}{n\sum_{i=1}^{n} U_i^2 - (\sum_{i=1}^{n} U_i)^2}. \tag{3.7}$$

The value of c for an uncorrelated sequence is expected to be with a 95% probability in the range $[\mu_n - 2\sigma_n, \mu_n + 2\sigma_n]$, where:

$$\mu_n = -\frac{1}{n-1}, \qquad \sigma_n = \frac{1}{n-1}\sqrt{\frac{n+1}{n(n-3)}}. \tag{3.8}$$

In order to determine the most appropriate RNG for our Monte Carlo simulations, we generated $n = 5000$ random numbers. We then applied the χ^2, with $N_{bin} = 50$ bins, and the serial correlation test for the linear congruence with and without shuffling and the Fibonacci RNG's. The results are shown in table 1.

The performance of the Fibonacci RNG is clearly superior to that of the linear congruence methods. Although the shuffled linear congruence exhibits comparable correlation to that of the Fibonacci, its χ^2 is much larger than the expected value, indicating that there are large numbers of bins with a statistically significant number of samples above or below the expected value.

In the following discussion, unless otherwise stated, it is implied that the RNG used is of the Fibonacci type.

Table 1: Statistical tests of different RNG's.

Test	χ^2	Correlation c
Expected value	$\nu = 49$	$[-0.028, 0.028]$
Linear Congruence	173	-0.023
L.C. with shuffling	170	0.005
Fibonacci	54	-0.002

3.3 The Relaxation Method

In chapter 2 the relaxation method (RM) was briefly mentioned. Because of its ease of implementation and straightforward physical interpretation, we have chosen RM as the standard with which to compare Monte Carlo methods to deterministic approaches. Convergence information for RM is obtained in this section as the most important criterion of evaluation of RM.

RM is perhaps the first numerical method to be applied to electrostatic problems, even before the advent of high speed digital computers, and the relevant literature can be traced back to Southwell's definitive work [97, 98, 99]. More recent applications to electrostatic problems are given by DiStasio and McHarris [2], Crow [100], Gash [1], MacDonald [101], Henderson [102] and Nachman [103].

The basis for the method is the *mean value theorem* [11]: for charge-free space the value of the electrostatic potential at any point is equal to the average of the potential over the surface of *any* sphere (or circle in the plane) centered on that point.

This fact provides the basis for an iterative solution: initial, often arbitrary, values for the potential are assigned to every point in a given domain. Subsequently, the potential at every point is readjusted to be the average of a selected set of neighboring points. This process continues iteratively until subsequent adjustments produce a change that is below

a prescribed threshold. In one dimension, this algorithm translates to:

$$\phi_m^{(j)} = \frac{\phi_{m-1}^{(j-1)} + \phi_{m+1}^{(j-1)}}{2}, \tag{3.9}$$

where $\phi_m^{(j)}$ is the potential at point m during the j-th iteration and $\phi_{m-1}^{(j-1)}, \phi_{m+1}^{(j-1)}$ are the potentials of its immediate neighboring points during the previous, $(j-1)$-th iteration.

The two-dimensional analogue of (3.9), in obvious notation, is:

$$\phi_{m,n}^{(j)} = \frac{\phi_{m,n-1}^{(j-1)} + \phi_{m,n+1}^{(j-1)} + \phi_{m-1,n}^{(j-1)} + \phi_{m+1,n}^{(j-1)}}{4}. \tag{3.10}$$

What is of interest to us in this section, is the following question: given a specified geometry, one- or two-dimensional, how fast do the algorithms described by (3.9) and (3.10) converge? The answer to this question is of extreme practical interest for our subsequent work. To answer it, consider (3.9) recast in the following form:

$$\frac{\phi_m^{(j)}}{(h^2/2)} = \frac{\phi_{m-1}^{(j-1)} + \phi_{m+1}^{(j-1)}}{h^2}, \tag{3.11}$$

with h being the spacing of the spatial grid on which the solution to Laplace's equation is sought. We rewrite (3.11) in a more suggestive form:

$$\frac{\phi_m^{(j)} - \phi_m^{(j-1)}}{(h^2/2)} = \frac{\phi_{m-1}^{(j-1)} + \phi_{m+1}^{(j-1)} - 2\phi_m^{(j-1)}}{h^2}. \tag{3.12}$$

Correspondingly, (3.10) is cast as:

$$\frac{\phi_{m,n}^{(j)} - \phi_{m,n}^{(j-1)}}{(h^2/4)} = \frac{\phi_{m-1,n}^{(j-1)} + \phi_{m+1,n}^{(j-1)} - 2\phi_{m,n}^{(j-1)}}{h^2} + \frac{\phi_{m,n-1}^{(j-1)} + \phi_{m,n+1}^{(j-1)} - 2\phi_{m,n}^{(j-1)}}{h^2}. \tag{3.13}$$

Unlike (3.9) and (3.10) which connect the potential at the point of interest with the potential at *neighboring* points during the previous iteration, (3.12) and (3.13) connect the potential at the point of interest during consecutive iterations, thus introducing an element

of "time". Indeed, examination of (3.12) and (3.13) reveals their similarity with the discretized form of the one- and two-dimensional *heat equation*, respectively:

$$\frac{\partial \phi}{\partial t} = \frac{\partial^2 \phi}{\partial x^2} \qquad \text{and} \qquad \frac{\partial \phi}{\partial t} = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}, \tag{3.14}$$

with the stepping in time being equal to $(h^2/2)$ for the one-dimensional case and $(h^2/4)$ for the two-dimensional. With equations (3.14) as starting point, Garabedian [104] has shown that the error in the j-th iterative approximation satisfies an inequality of the form:

$$|\phi_m^{(j)} - \phi_m| \leq R_1 e^{-j\lambda_1 h^2/2} \qquad \text{and} \qquad |\phi_{m,n}^{(j)} - \phi_{m,n}| \leq R_2 e^{-j\lambda_2 h^2/4}, \tag{3.15}$$

where ϕ_m ($\phi_{m,n}$) is the true potential for the one- (two-) dimensional case and R_1 (R_2) and λ_1 (λ_2) are fixed positive numbers. These equations show that the convergence of the relaxation method improves exponentially with the number of iterations.

3.4 Optimization of the Monte Carlo Method

In order to obtain some estimates about the time requirements, accuracy and convergence of the Monte Carlo method, we will consider the simplest possible solution of Laplace's equation, namely the solution to the one-dimensional problem:

$$\frac{d^2\phi(x)}{dx^2} = 0, \qquad \phi(0) = 0, \quad \phi(1) = 1. \tag{3.16}$$

We selected particularly simple boundary conditions and limited the interval of interest to $[0, 1]$ so that the solution has a very simple form:

$$\phi_e(x) = x, \qquad x \in [0, 1]. \tag{3.17}$$

The subscript e indicates that this is the *exact* solution. Although this particular case is almost trivial, we will extract a large amount of very useful information which will guide the development of our algorithms in two dimensions.

In this section we will mainly concentrate on trying to minimize the execution time of a Monte Carlo solution of (3.16).

3.4.1 Full Monte Carlo (FMC) Method

In order to obtain an estimate of the computational requirements of the Monte Carlo method, we will calculate analytically the expected number of steps undertaken during the execution of the random walk. In this section we consider the most straightforward type of Monte Carlo simulation, namely the Full Monte Carlo (FMC) method, in which the same number of random walks is started from *each* of the domain points.

Consider the interval $[0, 1]$ with a mesh of $N_P + 1$ points superimposed on it, figure 3.1. The spacing is uniform and equal to $h = 1/N_P$ and each grid point is located at $x_i = i/N_P$, with the index i varying from 0 to N_P.

As we prove in appendix A, the expected number of steps $U(x)$ for a random walk that starts at point x to reach either boundary point, in our case either $x = 0$ or $x = 1$, is given by the solution to the equation (see equation A.28):

$$\frac{d^2U(x)}{dx^2} = -\frac{2}{h^2} = -2N_P^2, \qquad U(0) = U(1) = 0. \tag{3.18}$$

The solution is:

$$U(x) = N_P^2 \left(x - x^2\right). \tag{3.19}$$

We are now interested in calculating the expected total number of steps $< U_{tot} >$ if we

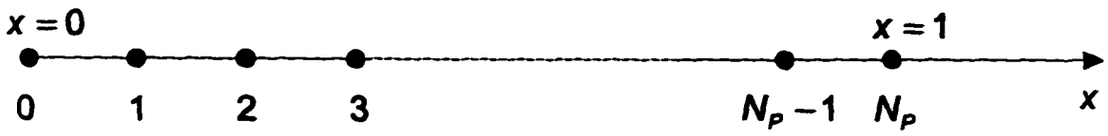

Figure 3.1: One-dimensional geometry for model simulations.

start one random walk from each of the $N_P - 1$ interior points. This is given by:

$$< U_{tot} >= \sum_{i=0}^{N_P-1} U(x_i) = N_P^2 \sum_{i=0}^{N_P-1} \left[\frac{i}{N_P} - \frac{i^2}{N_P^2} \right]. \tag{3.20}$$

Making use of the identities:

$$\sum_{i=0}^{N} i = \frac{N(N+1)}{2} \quad \text{and} \quad \sum_{i=0}^{N} i^2 = \frac{N(N+1)(2N+1)}{6}, \tag{3.21}$$

we find:

$$< U_{tot} >= \frac{N_P^3 - N_P}{6} \simeq \frac{N_P^3}{6}. \tag{3.22}$$

We note that the error in calculating $< U_{tot} >$ by direct integration of (3.19) is only $N_P/6$, which, for large N_P, is small compared to $N_P^3/6$, hence (3.22) follows. The salient point

of this argument is that the total expected number of steps per random walk is proportional to N_P^3. Hence, if we reduce N_P by a factor of 2, the overall execution time is reduced by a factor of 8. The savings in computational time are impressive but there will be a penalty in accuracy.

3.4.2 The Partitioned Monte Carlo (PMC) Method

We now consider the question: is there a more effective way to reduce the execution time by manipulating N_P,without adversely affecting the accuracy? The answer is affirmative and relies on the fact that, for Laplaces's equation, we need to know the value of the potential at the boundaries only.

Consider figure 3.2 which displays the expected number of steps as a function of position for $N_P = 10$. The total number of steps to execute a single random walk from all points is approximately $10^3/6 \simeq 167$. This is the average that would be obtained if a large number of random walks were performed from that point.

Now, suppose the value of the potential at $x = 0.5$ was known. Then, effectively, the problem has been decoupled into two independent problems: perform random walks in the interval $[0, 1/2]$ using as boundary values the known potential at $x = 0$ and $x = 0.5$ and repeat the same in the interval $[1/2, 1]$. The advantage is obvious: it takes only a total of $5^3/6 \simeq 21$ steps per random walk in the reduced intervals. Since, from (3.19), it takes $10^2(1/2 - 1/4) = 25$ steps to calculate the potential at $x = 0.5$, the overall cost of this approach is $25 + 2(21) = 67$ steps per random walk, as opposed to 167, a state of affairs illustrated in figure 3.3. The gain is very significant, a reduction in execution time of the order of 60%. Furthermore, it should be recognized that what we calculate is, effectively, the Green's function for the problem, as we pointed out in chapter 2. Now, the Green's function does not depend on the boundary conditions *per se*, but, rather, on the underlying geometry of the domain. If the domain possesses any kind of symmetry, this fact can be used to our advantage, unlike conventional techniques which can utilize symmetry only if

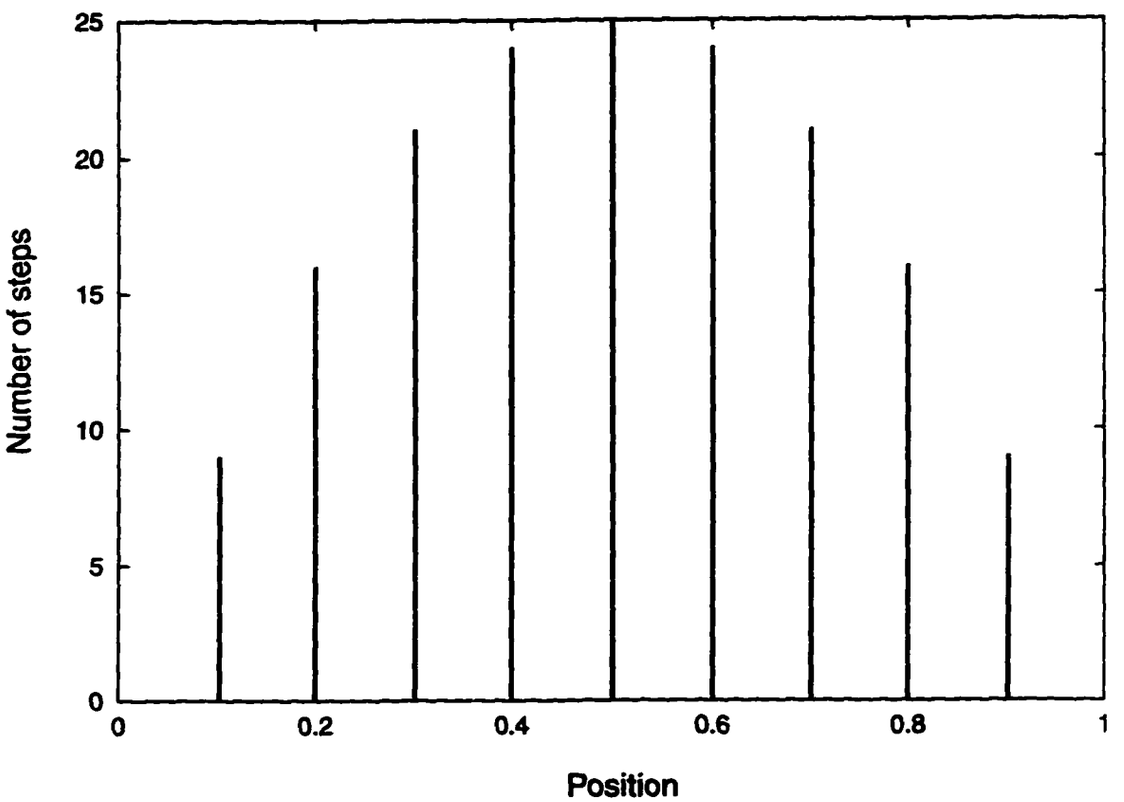

Figure 3.2: Expected number of steps per random walk for $N_P = 10$.

it is possessed by the domain *and* the boundary conditions. In the simple case studied, it is obvious that the Green's function is identical for each subdivision. Hence we need only gather statistics for random walks executed in one subdivision and use the same statistics for the other subdivision(s). So, in reality, we only require $25 + 21 = 46$ steps, a reduction of 73% in execution time.

Before formalizing this procedure, we should pause and address an important issue: since now the boundary value at $x = 0.5$ is not given but *calculated*, it would appear that we are introducing a systematic error and the overall result, for a given number of random walks, would not be as accurate as if the domain was not subdivided. The Markovian property (lack of "memory") of the random walks, however, ensures that this is not the case and the result in the subdivided procedure is at least as accurate as the original.

To prove this fact, consider a domain Ω with boundary $\partial\Omega$ and a subdomain Ω' with boundary $\partial\Omega'$, entirely contained within Ω (figure 3.4).

The potential of any point $\mathbf{p}$ inside Ω' can be calculated by starting random walks at $\mathbf{p}$. Assume that the random walk terminates at $\mathbf{q} \in \partial\Omega$. The *calculated* potential at $\mathbf{p}$ is then given as an integral over all boundary points $\mathbf{q}$:

$$\phi_c(\mathbf{p}) = \int_{\partial\Omega} P(\mathbf{p}, \mathbf{q})\phi(\mathbf{q})d\mathbf{q}, \tag{3.23}$$

where $P(\mathbf{p}, \mathbf{q})$ is the probability (Green's function) that a random walk starting at $\mathbf{p}$ terminates at $\mathbf{q}$, and $\phi(\mathbf{q})$ is the known boundary condition. Note that, since Ω' is entirely contained within Ω, any random walk starting at $\mathbf{p}$ and terminating at $\mathbf{q}$, must, by necessity, pass through some point on the boundary $\partial\Omega'$. What this means is that we can consider the random walk to be a two-step process: first the random walk reaches $\partial\Omega'$ and terminates, say at $\mathbf{r}'$, and then a *new* random walk starts from that point and terminates when it reaches $\partial\Omega$. We now appeal to the Markovian ("memoryless") property of the random walk to

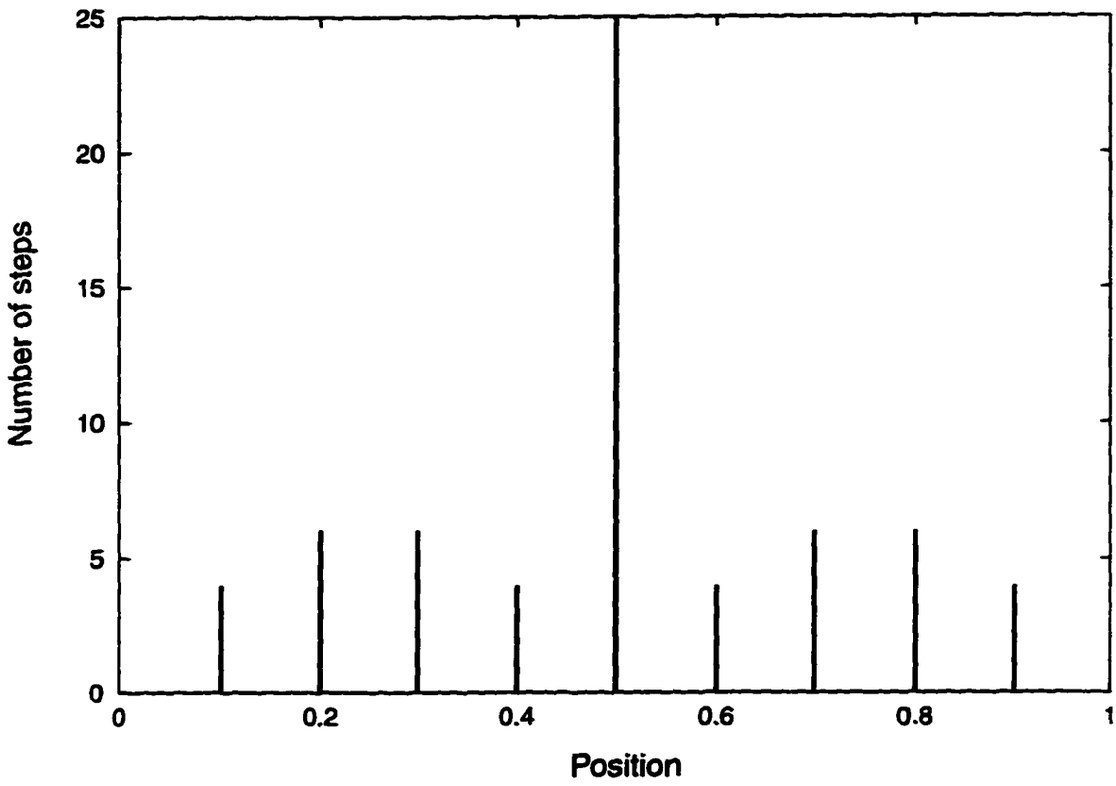

Figure 3.3: Expected number of steps per random walk for $N_P = 10$ and a two-fold subdivision of the original domain.

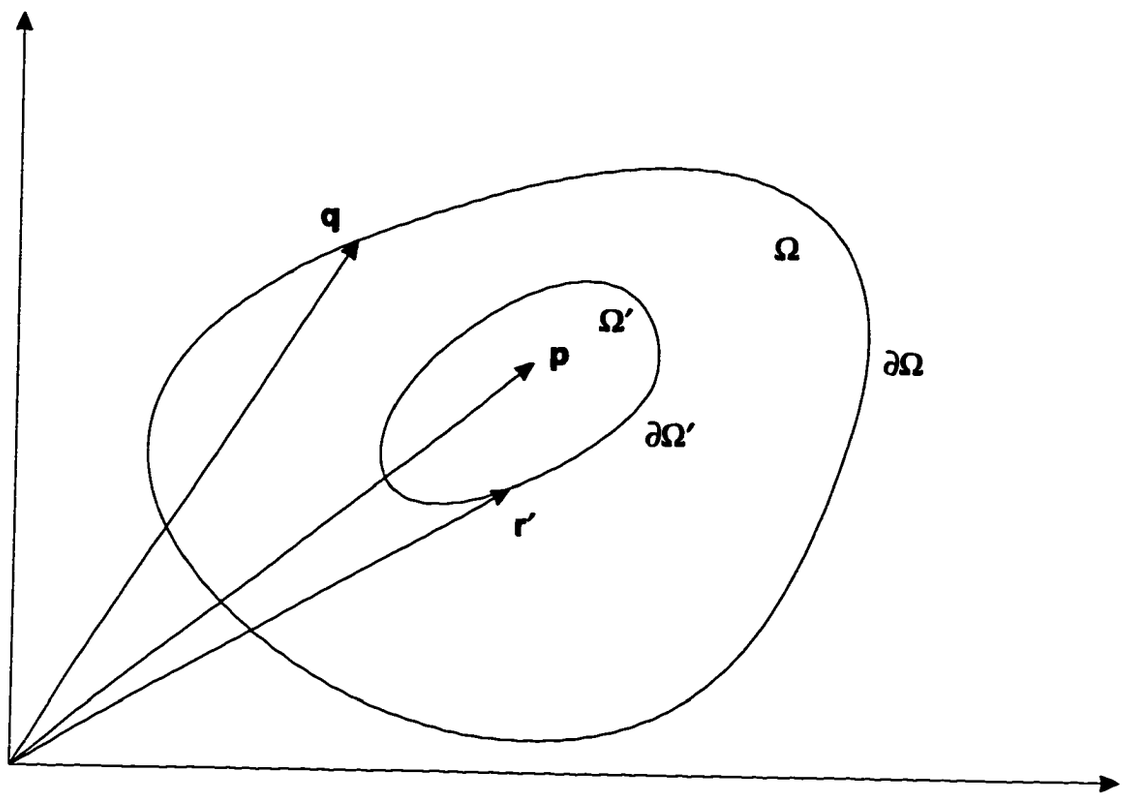

Figure 3.4: A subdivision Ω' of Ω.

note that these two random walks are statistically independent and hence we can write:

$$P(\mathbf{p},\mathbf{q}) = \int_{\partial\Omega'} P(\mathbf{p},\mathbf{r}')P(\mathbf{r}',\mathbf{q})d\mathbf{r}'. \tag{3.24}$$

Substituting (3.24) in (3.23), we have:

$$\phi_c(\mathbf{p}) = \int_{\partial\Omega} \left[\int_{\partial\Omega'} P(\mathbf{p},\mathbf{r}')P(\mathbf{r}',\mathbf{q})d\mathbf{r}'\right] \phi(\mathbf{q})d\mathbf{q}, \tag{3.25}$$

and, upon rearrangement,

$$\phi_c(\mathbf{p}) = \int_{\partial\Omega'} P(\mathbf{p},\mathbf{r}') \left[\int_{\partial\Omega} P(\mathbf{r}',\mathbf{q})\phi(\mathbf{q})d\mathbf{q}\right] d\mathbf{r}'. \tag{3.26}$$

But the term inside brackets is precisely $\phi(\mathbf{r}')$. Hence:

$$\phi_c(\mathbf{p}) = \int_{\partial\Omega'} P(\mathbf{p},\mathbf{r}')\phi(\mathbf{r}')d\mathbf{r}'. \tag{3.27}$$

In other words, whether we start with (3.23) or (3.27), we obtain the same result, provided (3.24) holds.

We now return to the original issue of Monte Carlo optimization. It is possible that if we subdivide the interval $[0, 1]$ in even more partitions than 2, we might reduce the computational time involved even more. However, it should be kept in mind that the more partitions we make, the more boundary values we need to calculate. We will investigate therefore if there is an optimal number of partitions that can be performed, beyond which the added overhead of calculating boundary values overcomes the computational savings.

Consider again the linear segment shown in figure 3.1, with $N_P + 1$ mesh points. We assume that we subdivide the segment into L partitions, each with N_P/L points (we are not concerned whether L is a proper divisor of N_P; our results will be slightly inaccurate but still highly useful). The i-th boundary is then located at $x_i = i/L$, where i runs

from 1 to $L-1$. The number of steps to calculate the value of the potential at each boundary is then $U(x_i)$, with U given by (3.19). The number of steps per random walk for each subdivision is given by (3.22), only now instead of N_P, we use N_P/L since each subdivision contains a reduced number of points. The total cost is then:

$$< U_{tot} >= \sum_{i=1}^{L-1} U(x_i) + \frac{N_P^3}{6L^3} = \sum_{i=1}^{L-1} N_P^2 \left(\frac{i}{L} - \frac{i^2}{L^2} \right) + \frac{N_P^3}{6L^3}. \tag{3.28}$$

With the help of the computer algebra system MATHEMATICA [105], this can be cast as:

$$< U_{tot} >= \frac{N_P^3}{6L^3} + N_P^2 \left(\frac{L-1}{2} - \frac{(2L-1)(L-1)}{6L} \right), \tag{3.29}$$

and differentiating with respect to L:

$$\frac{\partial < U_{tot} >}{\partial L} = \left(\frac{1}{2} - \frac{L-1}{3L} + \frac{(2L-1)(L-1)}{6L^2} - \frac{2L-1}{6L} \right) N_P^2 - \frac{N_P^3}{2L^4}. \tag{3.30}$$

Finally, solving the algebraic equation $\partial < U_{tot} > /\partial L = 0$ for various values of N_P, we obtain figure 3.5. Not only is there an optimal subdivision of the linear segment, but the number of subdivisions is comparatively small and increases slowly with N_P. This result, as far as we can ascertain, is novel and it will allow us to reduce execution times of the Monte Carlo method by a significant factor.

3.4.3 Theoretical Error Estimates

Before we proceed to present more methods of minimizing the computational time involved in various Monte Carlo schemes, it will be advantageous to develop certain error estimates of the method. This will facilitate the comparison of different schemes.

Regardless of the particular methodology used, the generation of data is similar in all Monte Carlo solutions of Laplace's equation: after a number of random walks, N_W, has

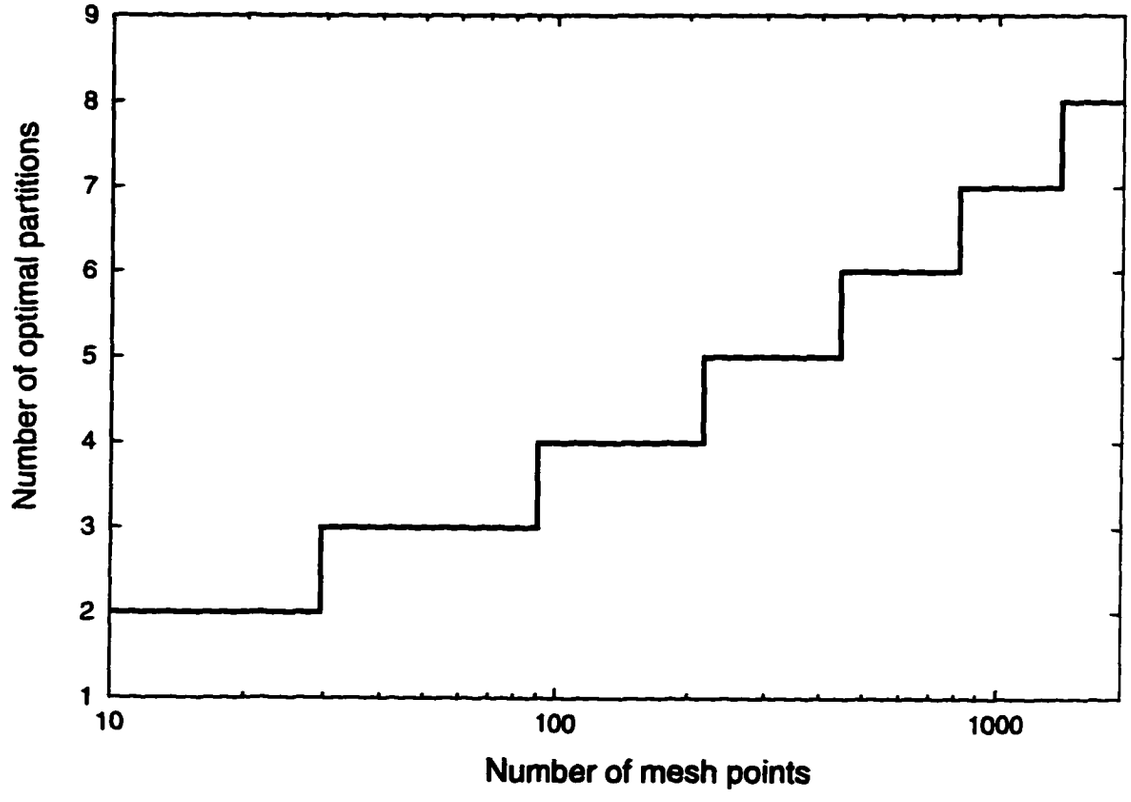

Figure 3.5: Optimal subdivision of a linear segment.

been executed, the collected statistics for each point x are used to compute the potential at that point, $\phi_c(x)$. The subscript c indicates that this is the calculated potential, in contrast to the exact potential $\phi_e(x)$.

In the Full Monte Carlo method and for the model problem (3.16), at every node i, $(i = 1, \ldots, N_P - 1)$, corresponding to point x, N_W random walks are started. Of them, a number $n_0(x)$ terminate at the left boundary $x = 0$ and $n_1(x)$ at the right boundary $x = 1$, with $n_0(x) + n_1(x) = N_W$. According to the main theorem, the calculated potential at i is:

$$\phi_c(x) = \frac{n_0(x)}{N_W}\phi(0) + \frac{n_1(x)}{N_W}\phi(1). \tag{3.31}$$

Since in our case $\phi(0) = 0$ and $\phi(1) = 1$, this reduces to:

$$\phi_c(i) = \frac{n_1(i)}{N_W}. \tag{3.32}$$

We now define the *error* at point x as $\epsilon(x) = \phi_e(x) - \phi_c(x)$ and the *absolute error* as $|\epsilon(x)| = |\phi_e(x) - \phi_c(x)|$. It is clear that $\epsilon(0) = \epsilon(1) = 0$ since the boundary values are given. We now ask a very important question, what is the standard deviation, $\sigma_\phi(x)$, in the value of $\phi_c(x)$ calculated from (3.32)? From basic probabilistic characteristics of the Monte Carlo method [47], we expect this to vary as $1/\sqrt{N_W}$. We give a more exact value here.

From (3.32), we see that:

$$\sigma_\phi^2(x) = \frac{1}{N_W^2}\sigma_{n_1}^2(x), \tag{3.33}$$

where $\sigma_{n_1}^2(x)$ is the uncertainty in the number of random walks that start at point x and terminate at $x = 1$. Since $n_1(x)$ is clearly binomially distributed, $\sigma_{n_1}^2(x) = \sqrt{N_W p(1-p)}$ [106, 95], where p is the probability that a random walk starting at x terminates at $x = 1$.

Fortunately, for this problem we know the answer beforehand: $p = x$ and hence we have the very important result:

$$\sigma_\phi(x) = \sigma_\epsilon(x) = \sqrt{\frac{x(1-x)}{N_W}}. \tag{3.34}$$

The expected $1/\sqrt{N_W}$ behavior is seen but there is also a positional dependence on the uncertainty, with the result at the midpoint ($x = 0.5$) being the most uncertain. Intuitively, this result is expected since the midpoint is the most distant from both boundaries.

We also denote the mean value of the error as $\mu_\epsilon(x)$. In terms of practical significance, we expect the error $\epsilon(x)$ to lie in the interval $[\mu_\epsilon(x) - \sigma_\epsilon(x), \mu_\epsilon(x) + \sigma_\epsilon(x)]$ with a 68% probability.

Finally, we present the statistical parameters of the absolute error, $|\epsilon(x)|$. Since the error itself, $\epsilon(x)$, is normally distributed, with mean $\mu_\epsilon(x)$ and standard deviation $\sigma_\epsilon(x)$, the absolute error follows a *folded normal* distribution [107, 108] with mean:

$$\mu_{|\epsilon|}(x) = \sigma_\epsilon(x)\sqrt{\frac{2}{\pi}}e^{-\theta^2(x)/2} - \mu_\epsilon(x)\left[1 - 2\Phi(\theta(x))\right], \tag{3.35}$$

and variance:

$$\sigma^2_{|\epsilon|}(x) = \mu^2_\epsilon(x) + \sigma^2_\epsilon(x) - \mu^2_{|\epsilon|}(x), \tag{3.36}$$

where $\theta(x) = \mu_\epsilon(x)/\sigma_\epsilon(x)$ and

$$\Phi(z) = \frac{1}{\sqrt{2\pi}}e^{-z^2/2}. \tag{3.37}$$

When $\mu_\epsilon(x) \simeq 0$, i.e. $|\epsilon(x)|$ follows a *one-sided* normal distribution, (3.35) becomes:

$$\mu_{|\epsilon|}(x) = \sigma_\epsilon(x)\sqrt{\frac{2}{\pi}}. \tag{3.38}$$

3.4.4 Single-Point Monte Carlo (SPMC) Method

In another attempt to minimize execution time, we take advantage of the Markovian property of random walks. Consider a random walk that originates at the midpoint of the interval $[0, 1]$. After a certain number of steps and before the walk terminates, the walk passes through some point, say x_a. Subsequently, the random walk terminates by reaching one of the boundaries. Now, for all practical purposes and because of the Markovian property, we can consider this random walk as actually *two* walks: one that started at the midpoint and one that started at x_a, due to the fact that the walk, when it passed x_a, has no "memory" of its previous state and can be considered as independent of its past history. The implications are significant: instead of starting N_W walks from *each* point, we start N_W walks from the midpoint *only* and keep track of which points this walk passes through and where it terminates. Using this procedure we gather statistics for many points with one walk.

One question we are able to answer with a combination of probabilistic and potential theory arguments, is the following: consider the midpoint x_m and an arbitrary point x_a (for convenience assume $x_m < x_a$); if N_W random walks are started at x_m, how many actually pass through x_a? Call this number N_a. There are three possibilities:

- A number of random walks, $N_{a,l}$ that originate at x_m, pass through x_a and terminate at the left boundary, $x = 0$.
- A number of random walks, $N_{a,r}$ that originate at x_m, pass through x_a and terminate at the right boundary, $x = 1$.
- A number of random walks, N_l that originate at x_m, do not pass at all through x_a and, by necessity, terminate at the left boundary, $x = 0$

It is clear that $N_{a,l} + N_{a,r} + N_l = N_W$ and $N_{a,l} + N_{a,r} = N_a$. The potentials at the points

x_a and x_m, ϕ_a and ϕ_m, respectively, are given by:

$$\phi_m = \frac{N_{a,r}}{N_W} \qquad \text{and} \qquad \phi_a = \frac{N_{a,r}}{N_a}, \tag{3.39}$$

from which the following elegant result follows:

$$N_a = N_W \frac{\phi_m}{\phi_a} \qquad \text{for} \qquad \phi_m < \phi_a, \tag{3.40}$$

and, by symmetry arguments,

$$N_a = N_W \frac{\phi_m}{1 - \phi_a} \qquad \text{for} \qquad \phi_m > \phi_a. \tag{3.41}$$

Combining (3.34) with (3.40) and (3.41) we are able to compare theoretically the standard deviation of the Full Monte Carlo (FMC) versus the Single-Point Monte Carlo (SPMC) method. The result is shown in figure 3.6.

It is clear that the SPMC method is less accurate than FMC, as expected, but the loss of accuracy is outweighed by the significant savings in execution time. By increasing the number of random walks at the single center point, the accuracy for *all* other points is also improved to the point where the SPMC result is as accurate or better as the FMC result.

3.4.5 The Monte-Carlo-Relaxation Method

Based on the previous discussion, we see that it is also possible, in principle, to combine the relaxation method with the Monte Carlo method in what we call the Monte-Carlo-Relaxation (MCR) method. In this synthesis, we are combining the best features of each individual method and suppress their weakest points. As far as we can ascertain, this is the first time Monte Carlo and Relaxation methods have been combined into one unified algorithm.

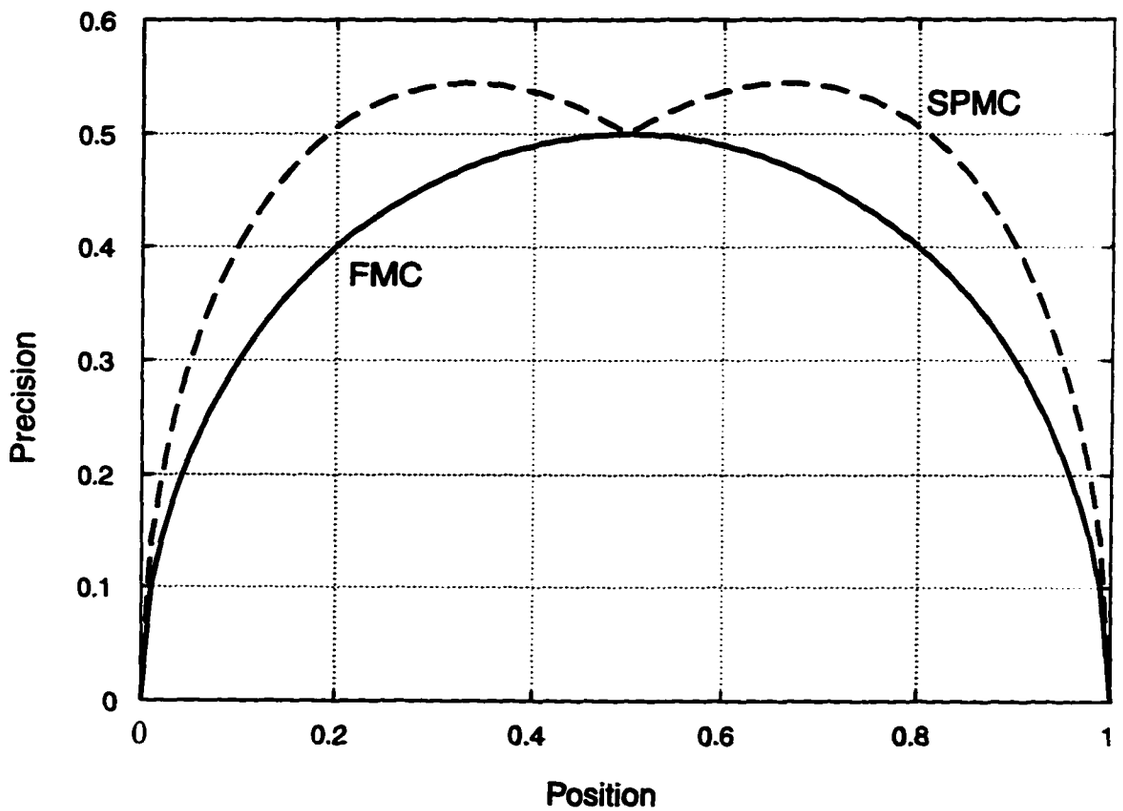

Figure 3.6: Comparison of the standard deviation of the Full Monte Carlo (FMC) vs. the Single-Point Monte Carlo (SPMC) methods. The standard deviation is normalized to $1/\sqrt{N_W}$.

There are several approaches that can be implemented:

- One could calculate the potential via Monte Carlo at a number of points and then use these values for relaxation in the in-between points.
- Alternatively, one could start with Monte Carlo for a certain number of random walks and then switch to relaxation. In this manner, the fast initial convergence of Monte Carlo is utilized and, subsequently, the switch to relaxation accelerates the final convergence. We will show in chapter 4 that this is indeed how these two methods behave.

3.5 Summary

We addressed in this chapter several important issues that have a significant effect on the convergence and precision of Monte-Carlo-type simulation methods. Based on a statistical investigation of commonly used RNG's we determined that the best available generator for our simulations is of the Fibonacci type.

An initial theoretical investigation of error estimates for the results of the FMC method on a prototype problem has allowed us to establish a number of optimized, derivative Monte Carlo methods, namely the SPMC, PMC and MCR methods. As far as we can ascertain these methods are presented here for the first time, at least within the context of the solution of elliptic problems. Although this discussion has focused entirely on one-dimensional cases, the extension of these alternative algorithms to two or more dimensions should present no significant conceptual or implementation problems.

In the next chapter we will perform actual simulations based on these techniques and compare their performance with the theoretical expectations, where applicable.

4 Numerical Experiments, Results and Applications

We present results of numerical experiments in which our algorithms are tested. We perform an exhaustive analysis of one-dimensional prototype simulations and use the results as guides for more intensive (and more interesting) two-dimensional simulations. We also demonstrate the viability of our method by solving Laplace's equation stochastically in non-trivial geometries and compare our results and execution times with traditional approaches. Where applicable, we compare with analytically known solutions. Finally, we perform practical evaluations of our work by applying it to the simulation of actual electrostatic deflectors and lenses which have been presented in the literature.

4.1 One-Dimensional Simulations

4.1.1 Relaxation Solution of a Prototype Problem

We consider equation (3.16):

$$\frac{d^2\phi(x)}{dx^2} = 0, \qquad \phi(0) = 0, \quad \phi(1) = 1, \tag{3.16}$$

with solution (3.17):

$$\phi_e(x) = x, \qquad x \in [0, 1]. \tag{3.17}$$

as our prototype one-dimensional problems. Apart from all the other reasons for which this particular problems is ideally suited to study the performance of various Monte Carlo schemes, as detailed in chapter 3, there is an added advantage intrinsic only to one-dimensional solutions of Laplace's equation, but, in general, not present in two or more dimensions: the approximation of the Laplacian via finite differences:

$$\nabla^2\phi = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2}, \tag{4.1}$$

is *exact*. In other words, there is no discretization error but only a truncation error, dependent on the method of solution used. This feature allows the exclusive concentration of the error analysis on the method used alone and not the discretization method, thus allowing meaningful comparison and classification of the results. Parenthetically, this also eliminates any dependence of the resulting calculated potential on the number of grid points, N_P, used in the simulations.

As a benchmark of the performance of the various methods we implemented, we performed simulations of (3.16) on a grid with $N_P = 20$ via the relaxation method. The results are shown in figure 4.1, for several iterations of the relaxation process. It is apparent from this figure that the relaxation method converges rather slowly initially, but as the number of iterations increases, convergence improves significantly (see also figure 4.3).

We also plot the execution time for the relaxation method as a function of the maximum absolute percentage error, $\mu_{|\epsilon|}$ for a grid with N_P =20 points, figure 4.2. The almost perfect exponential relationship predicted by (3.15) is immediately apparent.

Finally, in figure 4.3 we replot the execution time required for the relaxation method to reach a given level of accuracy, on a logarithmic scale. The "saturation" of the method is clearly visible.

4.1.2 Full Monte Carlo Simulations

We implemented the Full Monte Carlo (FMC) method, as outlined in section 3.4.1. From *each* of the $N_P - 1$ interior points, N_W random walks were started. Of those N_W walks, $n_0(x)$ terminated at $x = 0$ and $n_1(x)$ terminated at $x = 1$. With the help of the known boundary values $\phi(0)$ and $\phi(1)$, the potential at each point was calculated via (3.31):

$$\phi_c(x) = \frac{n_0(x)}{N_W}\phi(0) + \frac{n_1(x)}{N_W}\phi(1). \tag{3.31}$$

The simulation results are shown in the sequence of figures 4.4-4.7, for numbers of random walks ranging from $N_W = 10$ to $N_W = 10^4$. The underlying grid has $N_P = 20$

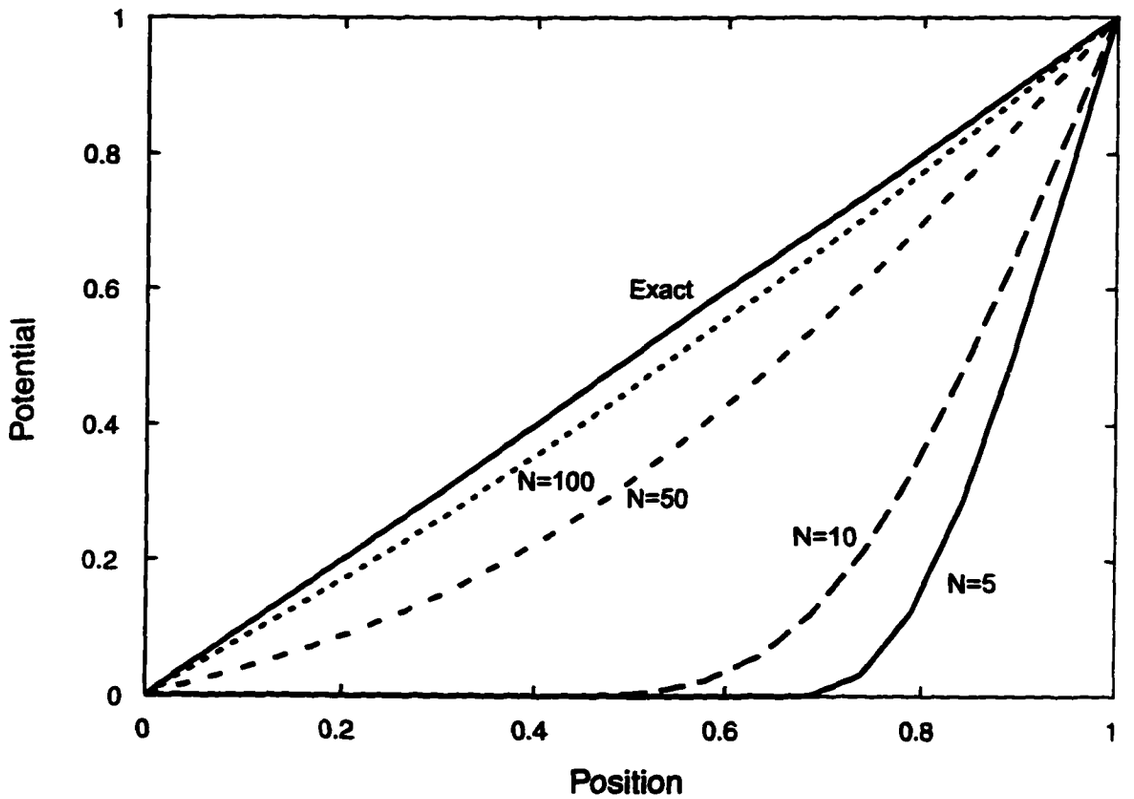

Figure 4.1: Relaxation method solution of the prototype one dimensional problem. Results after several iterations ($N = 5, 10, 50$ and 100) for a grid with $N_P = 20$ points are shown.

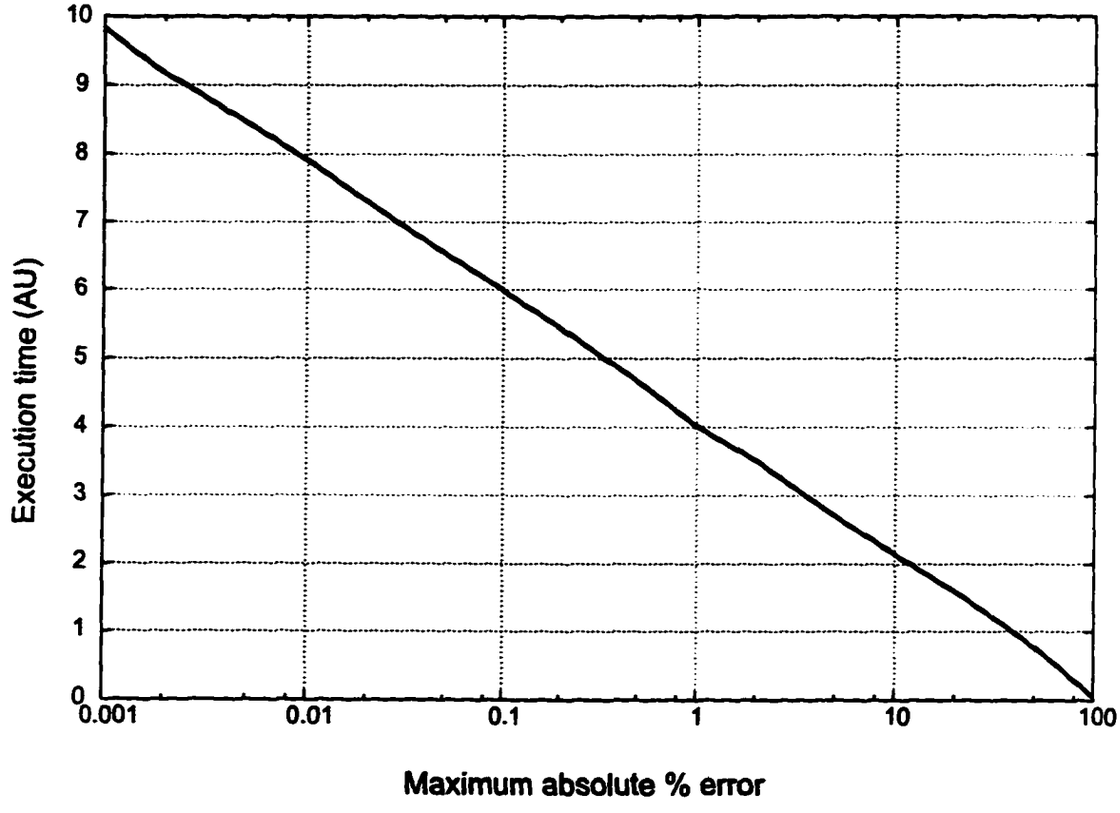

Figure 4.2: Execution time (in arbitrary units) of the Relaxation method, as a function of the average absolute percentage error. Results are shown for a grid of N_P =20 points.

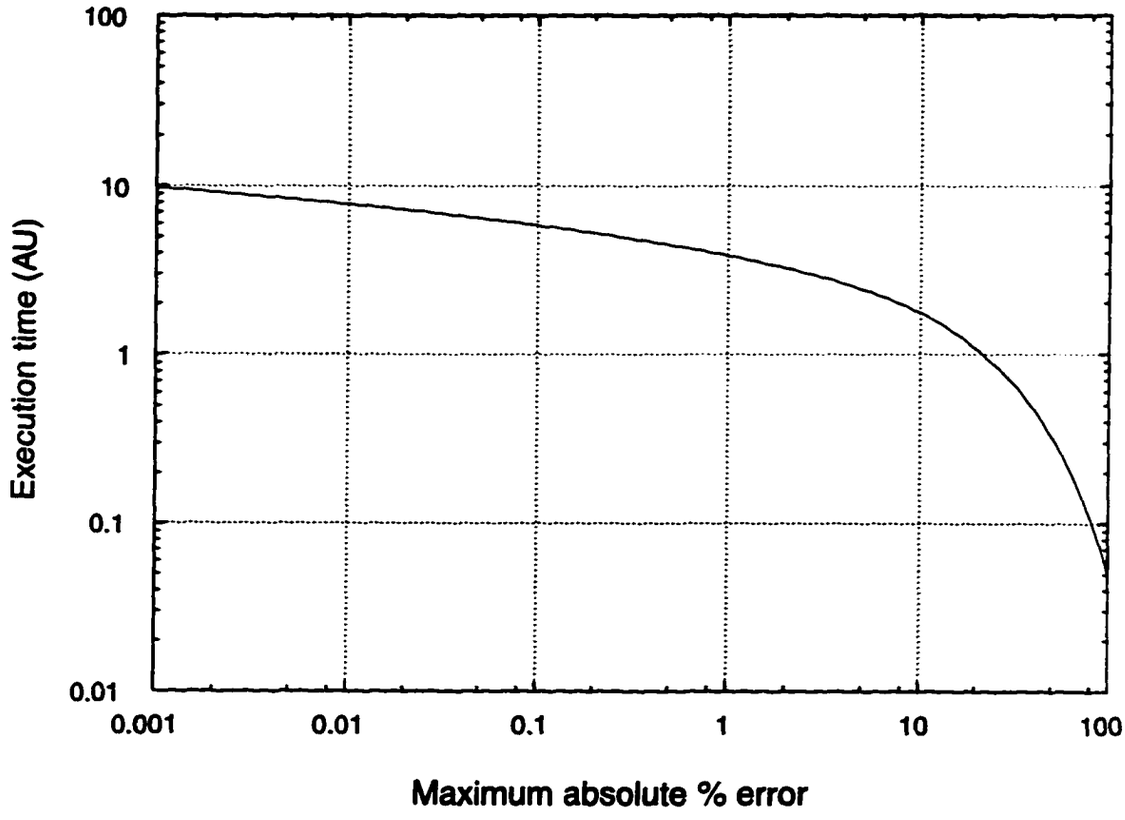

Figure 4.3: Execution time (in arbitrary units) of the Relaxation method, on a logarithmic scale, as a function of the maximum absolute percentage error. Results are shown for a grid of N_P =20 points.

points, a density that, without being overly coarse, limits the number of operations and execution times to manageable levels. Of course, as mentioned previously, the results presented in the one-dimensional simulations are entirely independent of grid spacing due to the absence of any discretization error. This situation, however, is not applicable to two-dimensional simulations.

Before we present simulation results for the error, we should clarify how these results were obtained. Examination of the FMC method and (3.31) reveals that during a simulation run, the potential at any given point x is calculated by averaging a number, $n_0(x)$, of "zeroes" and a number, $n_1(x)$, of "ones". As far as averages are concerned, this *bimodal* distribution is quite adequate. For the calculation of the standard deviation, however, this approach is unsuitable. Instead, we performed a number, N_E, of "experiments", where each experiment consisted of a FMC run at every point. We obtained thus N_E values of the potential at each point and the average of these values is what is plotted in figures 4.4–4.7. This set of N_E potentials at every point has a standard deviation $\sigma_E(x)$. From basic statistical theory [106], the standard deviation in the error of a single experiment, $\sigma_\epsilon(x)$ is related to $\sigma_E(x)$ via:

$$\sigma_E(x) = \frac{\sigma_\epsilon(x)}{\sqrt{N_E}}. \tag{4.2}$$

Thus determining $\sigma_E(x)$ experimentally, we can obtain $\sigma_\epsilon(x)$ from (4.2). The results for the simulations corresponding to figures 4.4–4.7 are shown in the sequence of figures 4.8–4.9. Comparing with the theoretically expected results, (3.34), which are also plotted on these figures, we see an excellent agreement. The $\sqrt{x(1-x)}$ dependence is clear. It is also seen that the maximum standard deviation occurs at the midpoint $x = 0.5$, again as expected theoretically.

Although these graphs indicate clearly the spatial behavior of the precision (standard deviation of the error), the dependence on the number of random walks, N_W, is not clear.

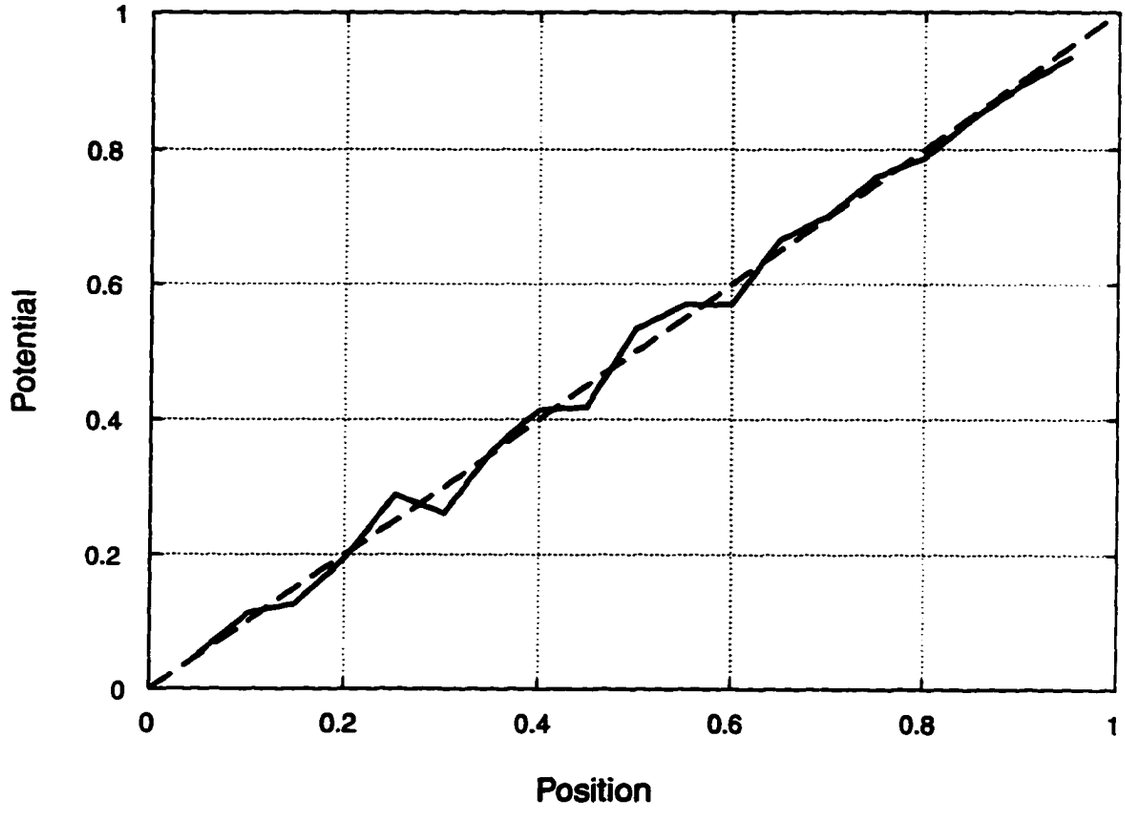

Figure 4.4: Results of Monte Carlo simulations for a prototype potential. The dashed line represents the exact solution while the solid line represents the calculated potential for $N_W = 10$ random walks per point. The underlying grid contains $N_P = 20$ points.

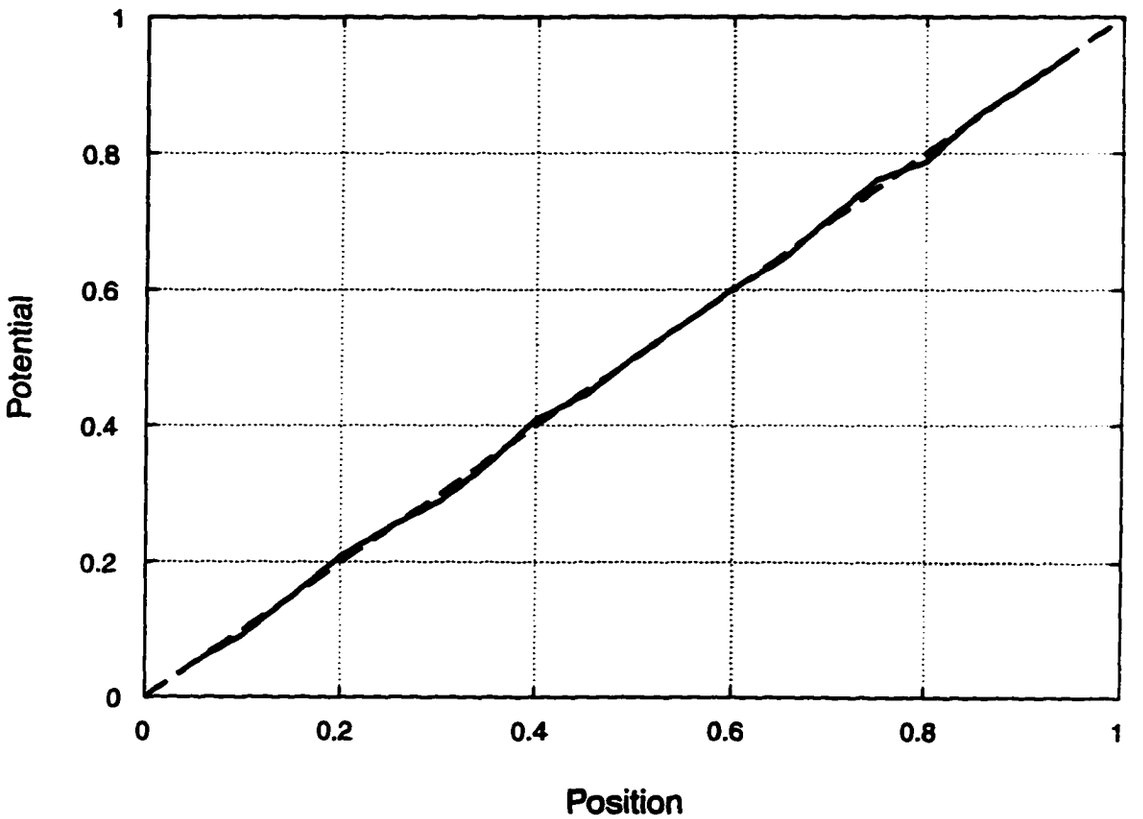

Figure 4.5: Results of Monte Carlo simulations for a prototype potential. The dashed line represents the exact solution while the solid line represents the calculated potential for $N_W = 100$ random walks per point. The underlying grid contains $N_P = 20$ points.

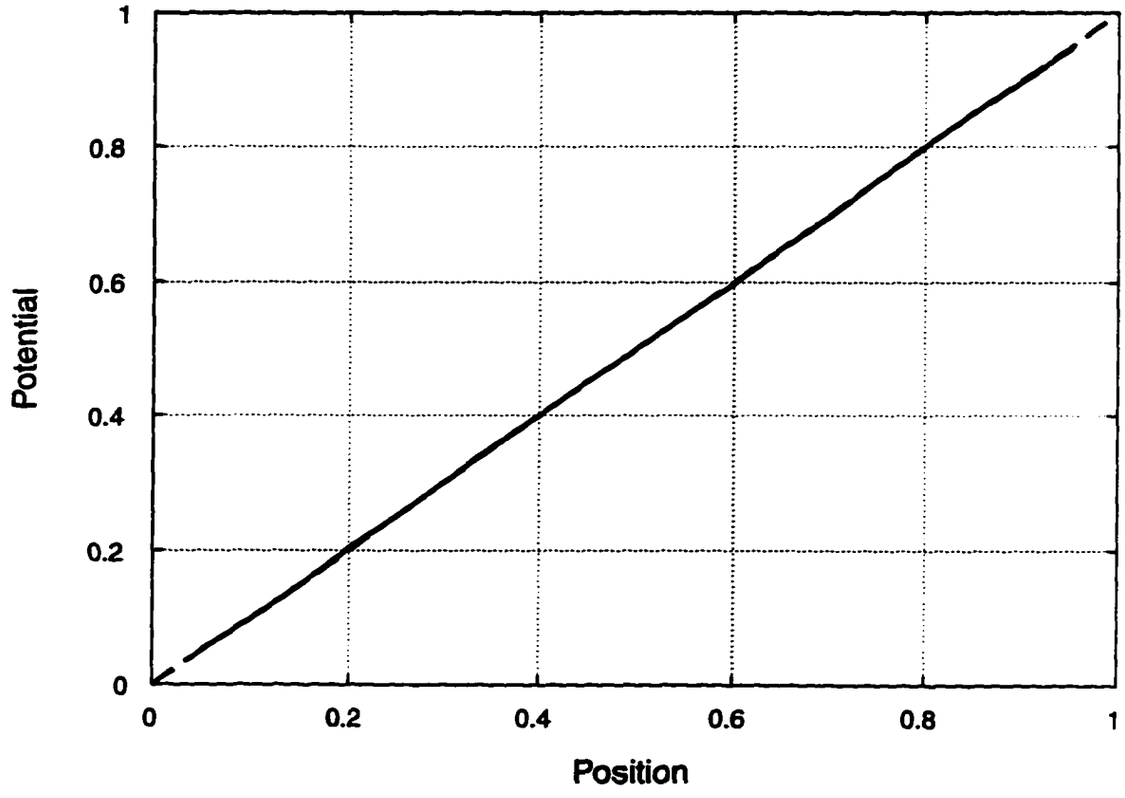

Figure 4.6: Results of Monte Carlo simulations for a prototype potential. The dashed line represents the exact solution while the solid line represents the calculated potential for $N_W = 10^3$ random walks per point. The underlying grid contains $N_P = 20$ points.

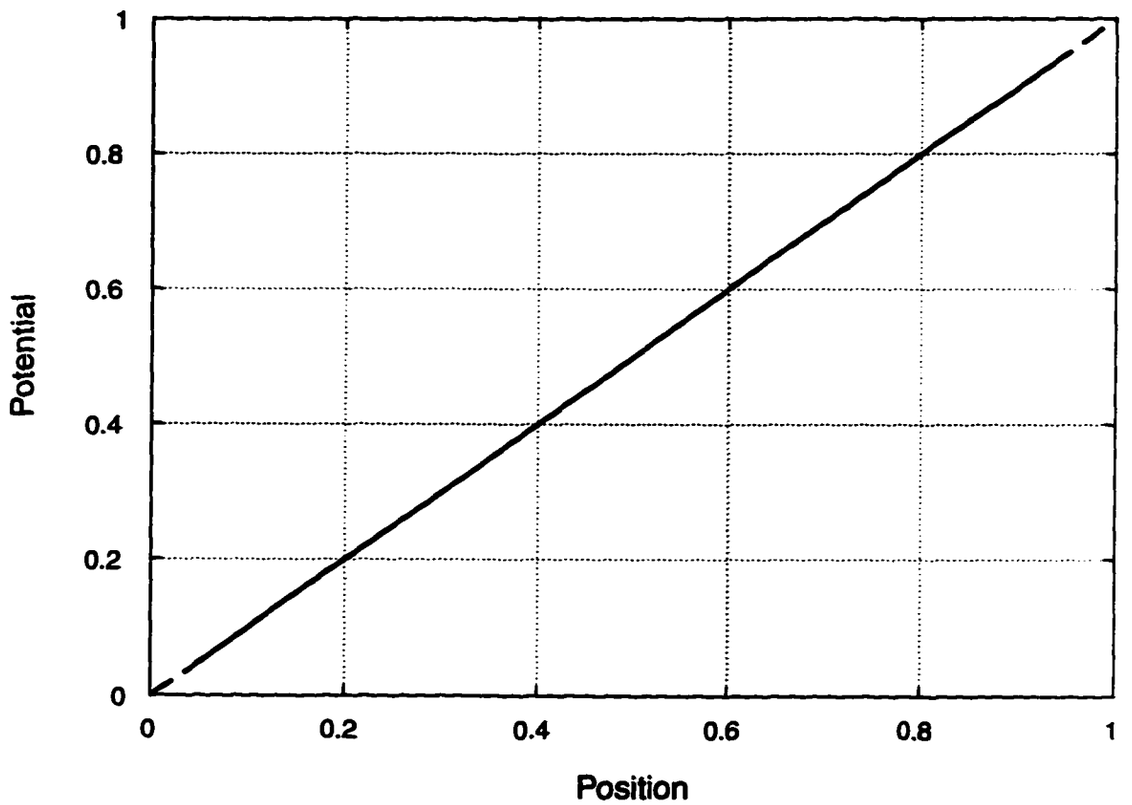

Figure 4.7: Results of Monte Carlo simulations for a prototype potential. The dashed line represents the exact solution while the solid line represents the calculated potential for $N_W = 10^4$ random walks per point. The underlying grid contains $N_P = 20$ points.

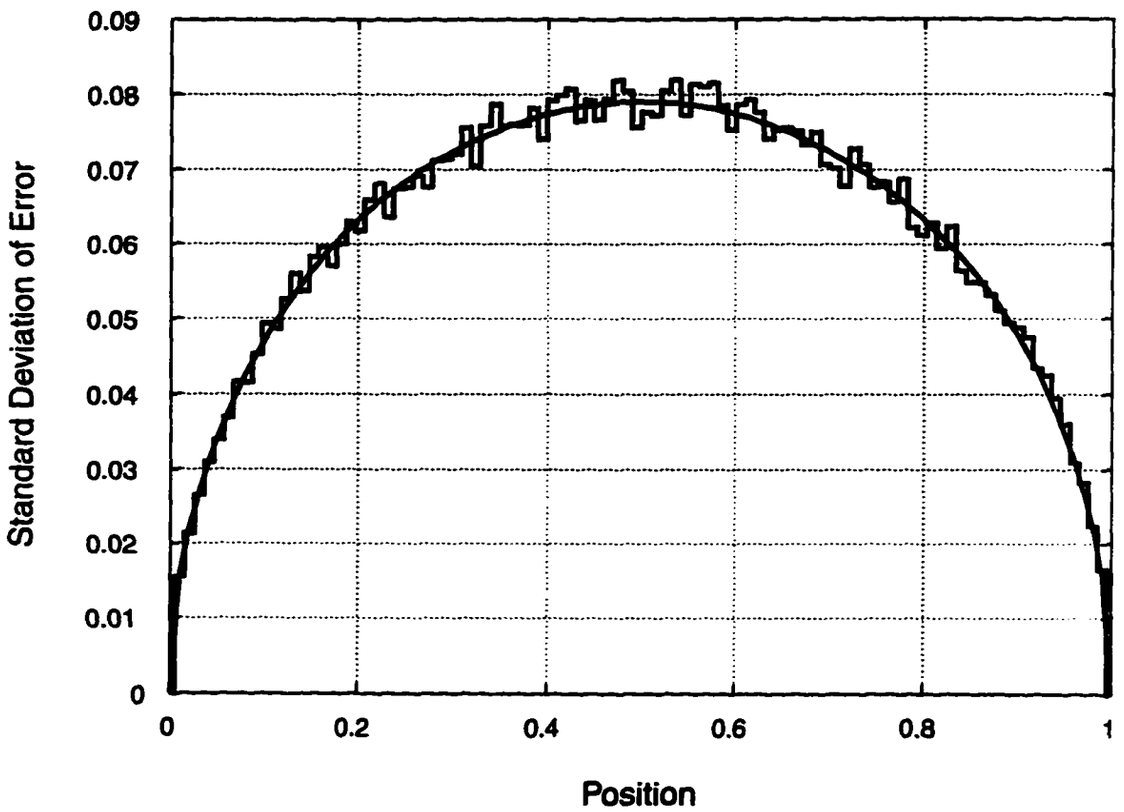

Figure 4.8: Results of Monte Carlo simulations for a prototype potential. The continuous curve represents the theoretically expected value of the standard deviation of the error, $\sigma_\epsilon(x)$, while the stepwise curve represents the calculated values, for $N_W = 10$ random walks per point. The underlying grid contains $N_P = 20$ points and the results were obtained from $N_E = 50$ experiments.

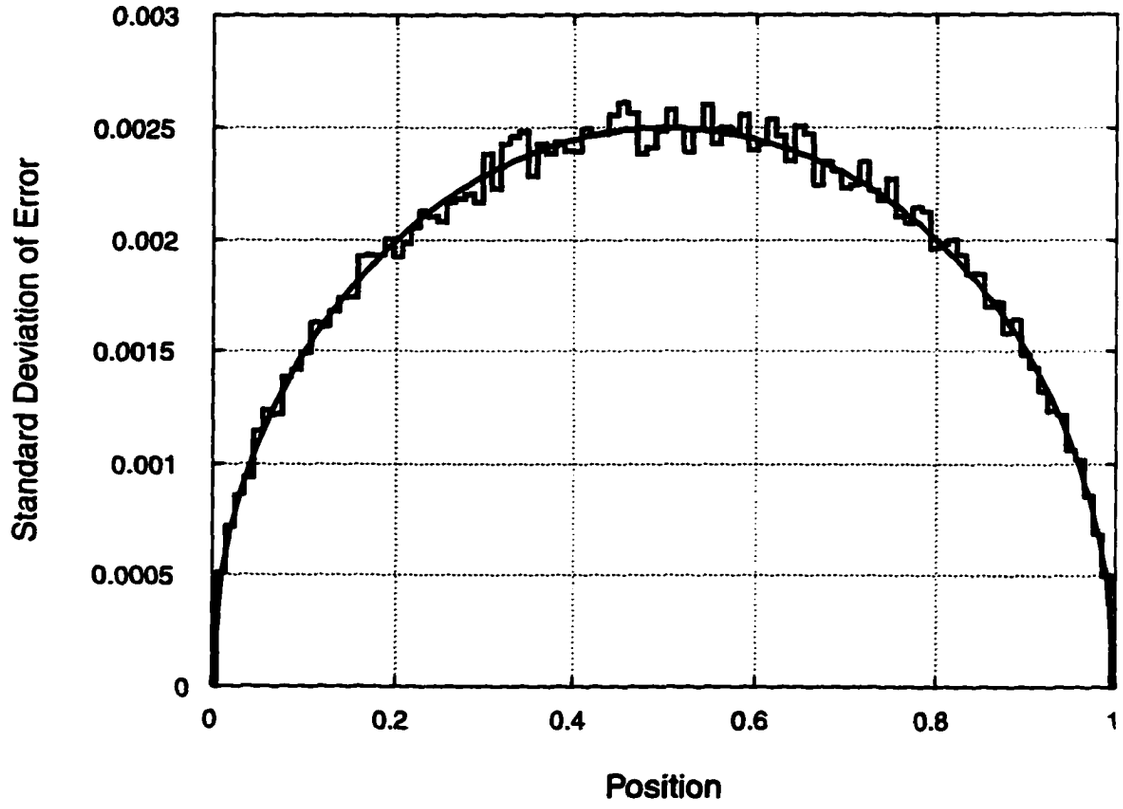

Figure 4.9: Results of Monte Carlo simulations for a prototype potential. The continuous curve represents the theoretically expected value of the standard deviation of the error, $\sigma_\epsilon(x)$, while the stepwise curve represents the calculated values, for $N_W = 10^4$ random walks per point. The underlying grid contains $N_P = 20$ points and the results were obtained from $N_E = 50$ experiments.

Therefore we concentrated on the midpoint where σ_ϵ is the highest and evaluated it as a function of N_W. The results are shown in figure 4.10 along with the theoretically expected results, (3.34). These results do not depend on the coarseness of the grid, N_P. Indeed, we repeated the simulations for values of $N_P = 10, 50$ and 100 with identical results to those shown in figure 4.10. We did, however, encounter some unexpected results when we used the simple linear congruence random number generator instead of the Fibonacci RNG. This situation is depicted in figure 4.11. It is clear from this figure that there is a correlation among the walks for large numbers of walks. This correlation arises from the highly correlated nature of "random" numbers in the sequence generated by the linear congruence RNG. This result is known [92] and arises because of the *quasi*-randomness of the sequence. Although, at first glance, it would appear that this correlation influences the simulations favorably, by providing a standard deviation that is better than we could theoretically expect, the accuracy itself is not improved. Furthermore, there are several problems associated with such sequences of "random" numbers, the most severe of which are the unpredictability of the results and the difficulty of optimizing a Monte Carlo method based on quasi-random generators [92]. In the remainder of this work we will use the Fibonacci RNG exclusively.

In figure 4.12 we show the mean absolute error, $\mu_{|\epsilon|}$ for the midpoint along with the maximum value attained by the error, ϵ_{max}. This plot provides an indication of the extent of the error. Since it is a logarithmic plot, the minimum value of the error, ϵ_{min}, which can be negative, is not shown. For comparison, lines for 1-, 2-, and 3-σ predictions based on (3.34) are also plotted. Only one data point lies outside the 3-σ region.

The validity of (3.38):

$$\mu_{|\epsilon|}(x) = \sigma_\epsilon(x)\sqrt{\frac{2}{\pi}}, \tag{3.38}$$

is demonstrated in figure 4.13 where the mean absolute value of the error, $\mu_{|\epsilon|}$, is plotted

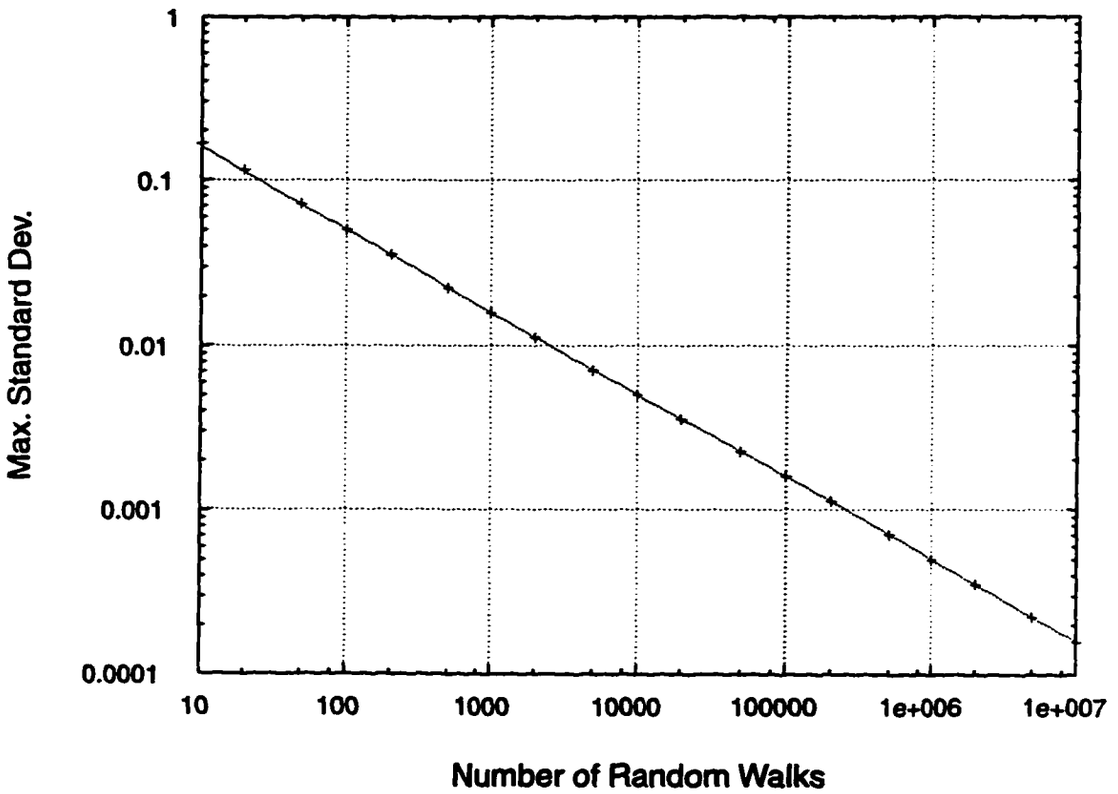

Figure 4.10: Results of Monte Carlo simulations for a prototype potential. The standard deviation of the error at the midpoint, σ_ϵ, is shown as a function of the number of random walks, N_W. The points indicated by + correspond to simulations while the dotted line represent a least squares fit. The theoretical prediction is indistinguishable from the least squares line. The results were obtained from $N_E = 50$ experiments on a grid of $N_P = 20$ points.

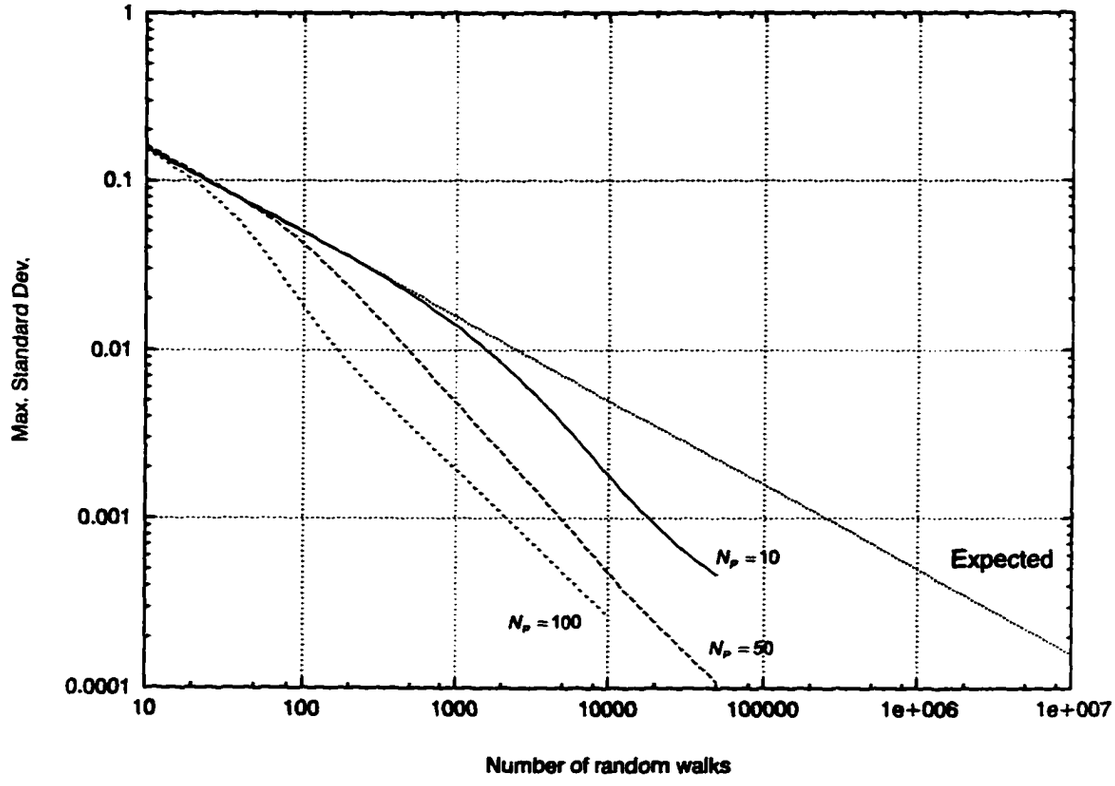

Figure 4.11: Results of Monte Carlo simulations for a prototype potential. The standard deviation of the error at the midpoint, σ_ϵ, is shown as a function of the number of "random walks", N_W, for various grid spacings. The linear congruence RNG was used. The straight line represents the theoretically expected result. The significant deviation from the expected behavior is due to the use of quasi-random number sequences.

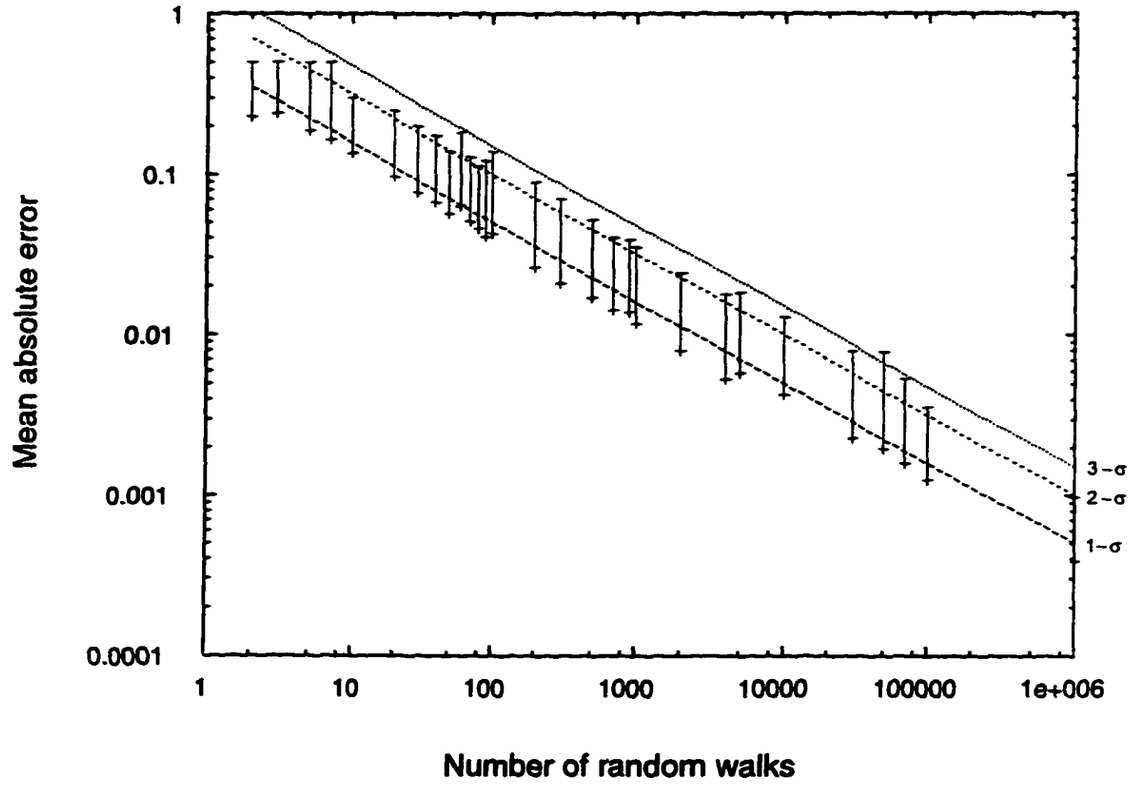

Figure 4.12: Results of Monte Carlo simulations for a prototype potential. The mean absolute value of the error at the midpoint, $\mu_{|\epsilon|}$, is shown as a function of the number of random walks, N_W, for a grid spacing of $N_P = 20$. The straight lines indicate the theoretically predicted 1-, 2- and 3-σ regions and the top horizontal bar at each data point corresponds to the maximum error. The results were obtained from $N_E = 50$ experiments.

along with the theoretical prediction of (3.38). We are also interested in the number of steps required on the average to reach either boundary. The theoretical solution was given in (3.19). A plot of this quantity is shown in figure 4.14, where the strong dependence on the number of grid points, N_P, is evident.

In figures 4.15-4.17 we plot the actual number of steps to reach a boundary, as determined by numerical experiments, for various grid sizes. To obtain statistically meaningful results, 10^4 random walks from every point were executed in each case. These results clearly verify the correctness of (3.19). The theoretical prediction and a parabolic fit based on the data were indistinguishable.

Of prime interest is also the amount of CPU time required for these simulations. The result is shown in figure 4.18.

We see from figure 4.18 that the execution time (T_e) depends on N_W and N_P as $T_e \sim N_W N_P^3$. The simple linear dependence on N_W is expected and the N_P^3 dependence is predicted by (3.22).

Combining the information conveyed in figures 4.13 and 4.18, we obtain the very important graph shown in figure 4.19 which indicates the CPU execution time requirements of the FMC method for a given performance level.

Finally, a comparison between the relaxation and FMC methods is shown in figure 4.20 where the information from figures 4.2 and 4.19 has been superimposed. The initial fast convergence of the FMC method is clearly seen, as is also the superiority of the relaxation method when progressively higher precision is required. This observation forms the basis of the Monte-Carlo-Relaxation (MCR) method.

4.1.3 The Partitioned Monte Carlo Method

We implemented the Partitioned Monte Carlo (PMC) method following the procedure outlined in section 3.4.2. From figure 3.5 we see that the optimal number of subdivisions for $N_P = 20$, is $L = 2$. The results, along with the timing for the FMC and relaxation

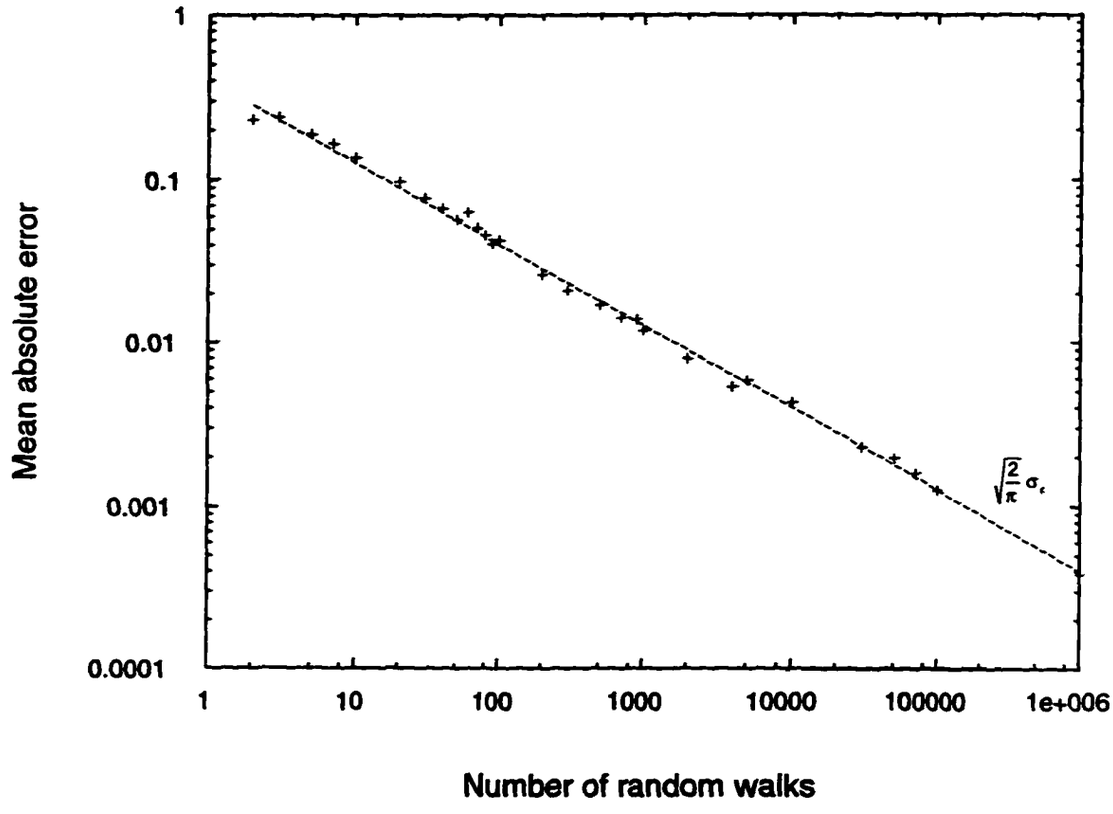

Figure 4.13: Results of Monte Carlo simulations for a prototype potential. The mean absolute value of the error at the midpoint, $\mu_{|\epsilon|}$, is shown as a function of the number of random walks, N_W, for a grid spacing of $N_P = 20$. The straight line indicates the theoretically predicted value, $\mu_{|\epsilon|} = \sqrt{2/\pi}\sigma_\epsilon$. The results were obtained from $N_E = 50$ experiments.

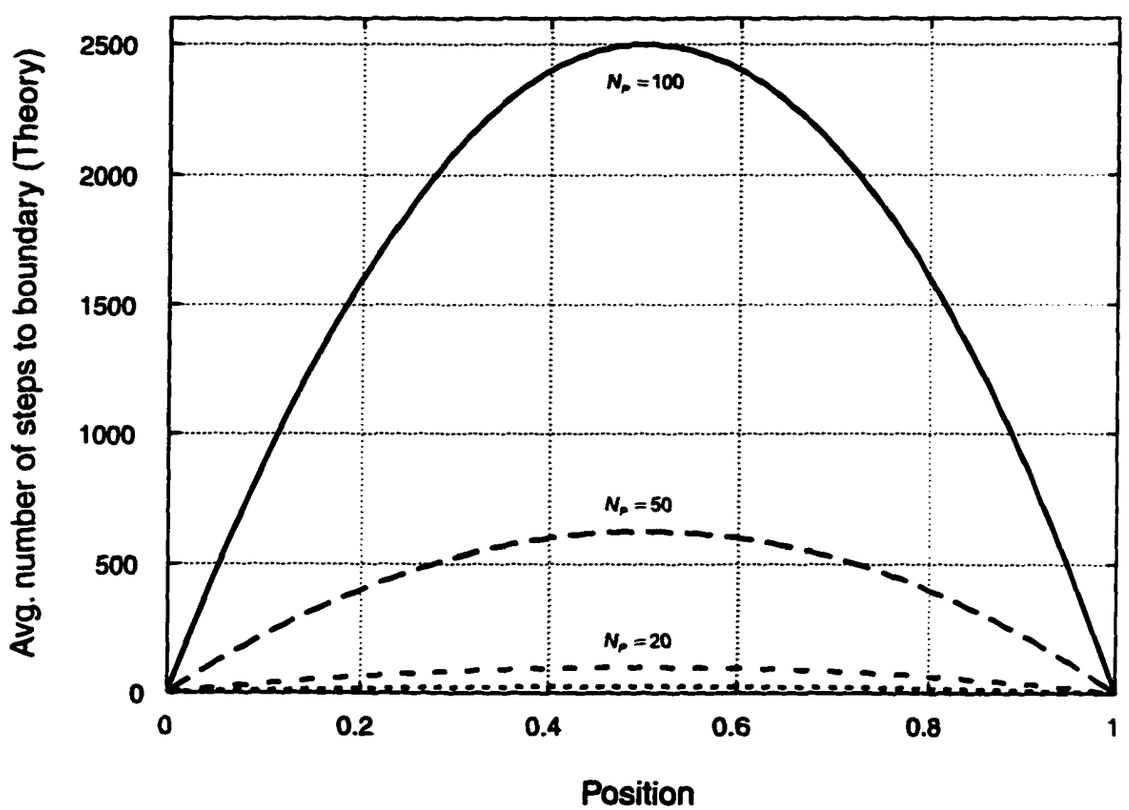

Figure 4.14: Theoretical prediction for the average number of steps required for a random walk to reach either boundary, as a function of the position of the starting point of the random walk. Results are shown for grids of N_P =100, 50, 20 and 10 points. The $N_P = 10$ result corresponds to the dotted line very close to the horizontal axis.

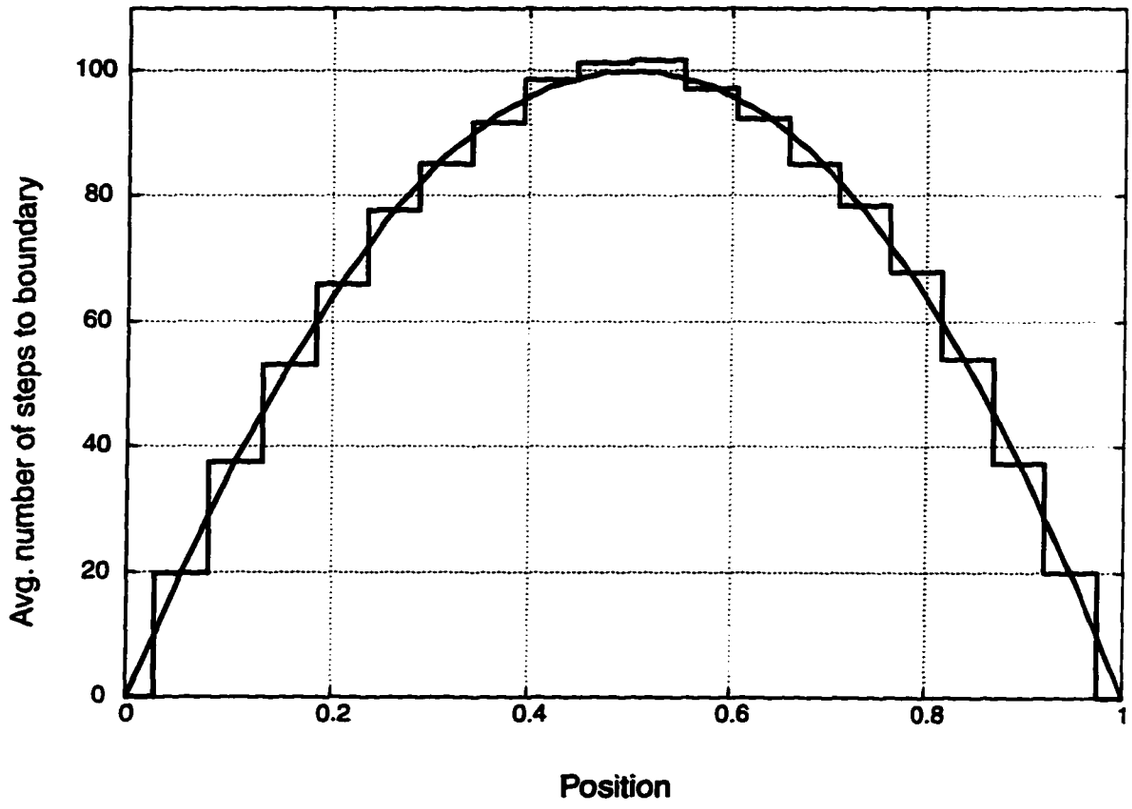

Figure 4.15: Theoretical prediction for the average number of steps required for a random walk to reach either boundary, as a function of the position of the starting point of the random walk (continuous line). The solid steps represent numerical simulations. Results are shown for a grid of N_P =20 points. There were $N_W = 10^4$ random walks started per point.

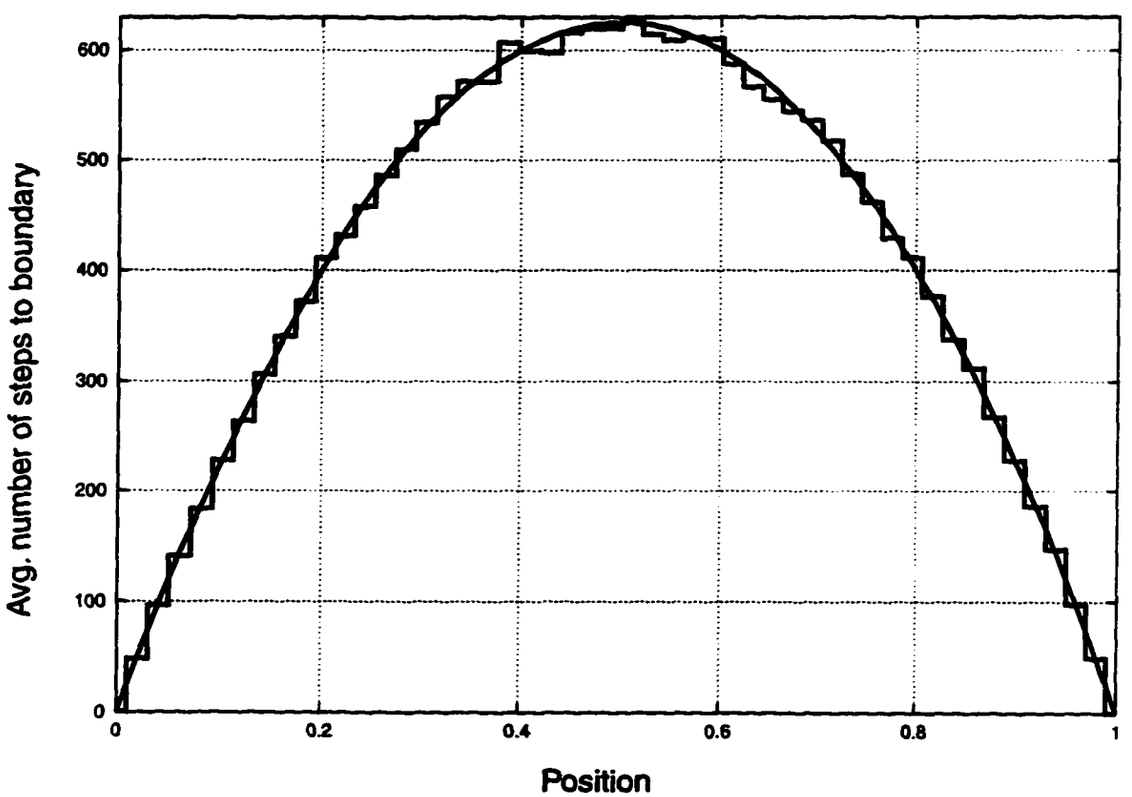

Figure 4.16: Theoretical prediction for the average number of steps required for a random walk to reach either boundary, as a function of the position of the starting point of the random walk (continuous line). The solid steps represent numerical simulations. Results are shown for a grid of N_P =50 points. There were $N_W = 10^4$ random walks started per point.

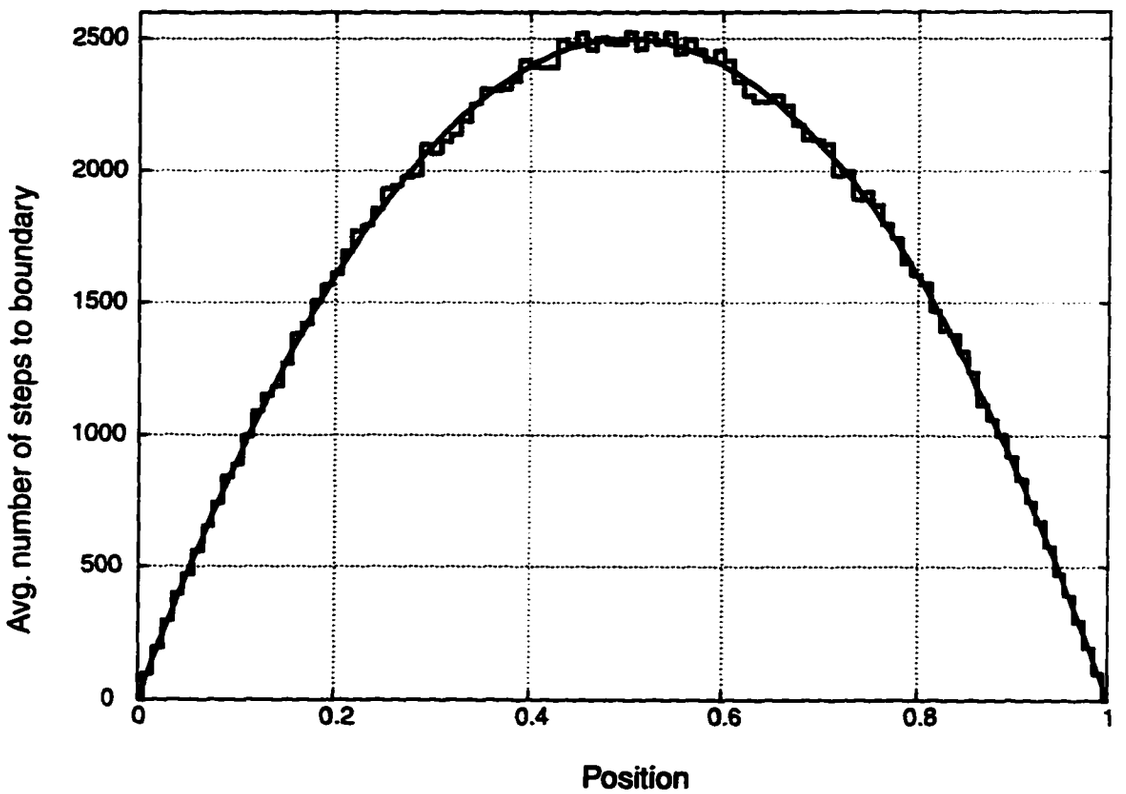

Figure 4.17: Theoretical prediction for the average number of steps required for a random walk to reach either boundary, as a function of the position of the starting point of the random walk (continuous line). The solid steps represent numerical simulations. Results are shown for a grid of N_P =100 points. There were $N_W = 10^4$ random walks started per point.

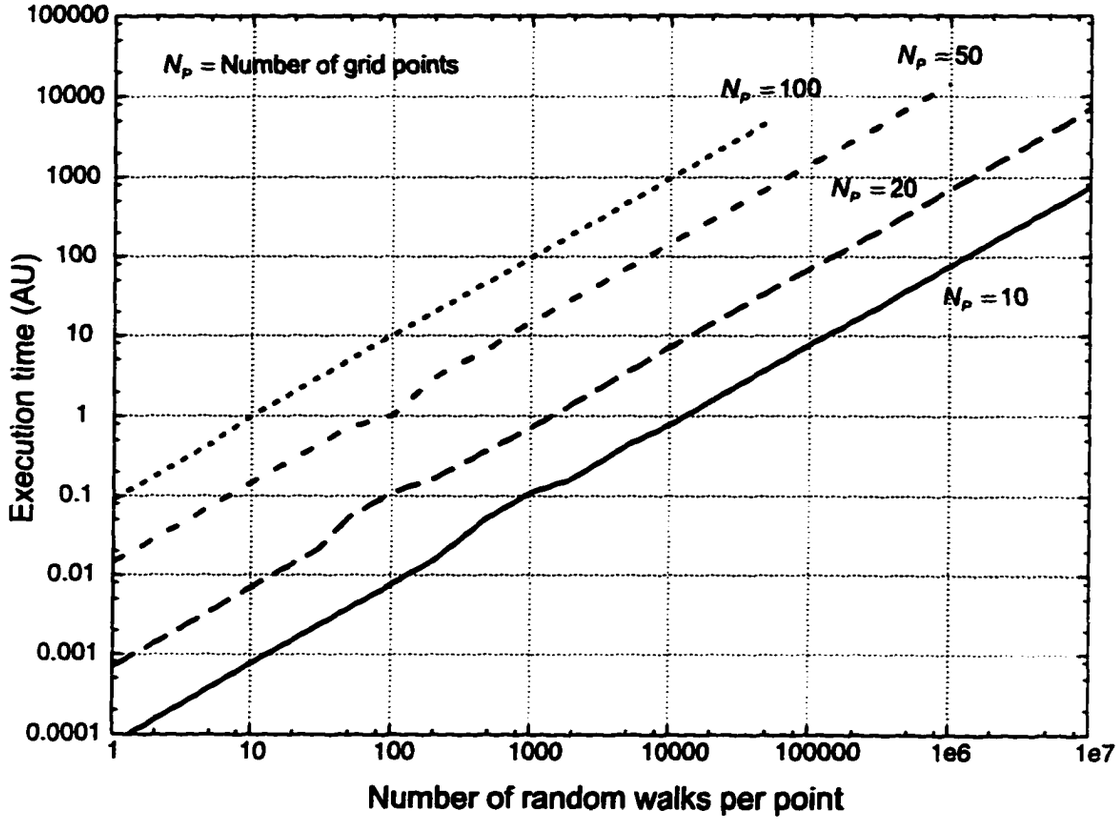

Figure 4.18: Execution time (in arbitrary units) of the Full Monte Carlo method, as a function of the number of random walks per point for various grid spacings.

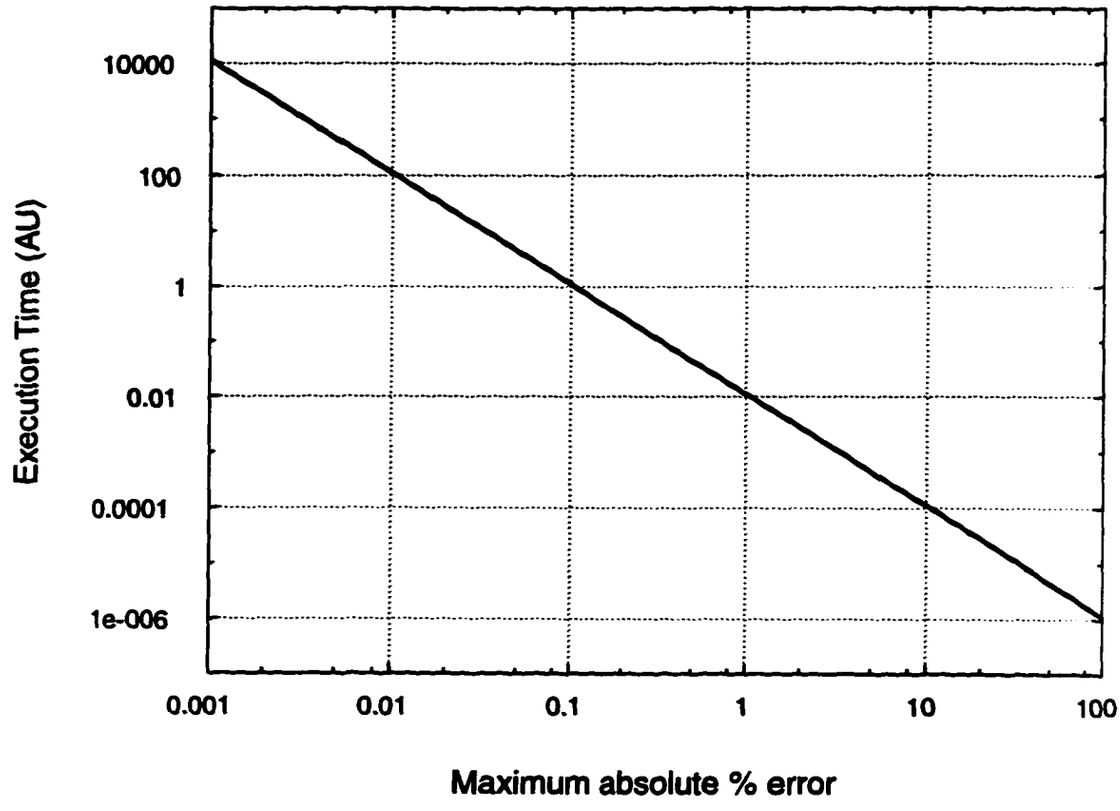

Figure 4.19: Execution time (in arbitrary units) of the Full Monte Carlo method, as a function of the maximum absolute % error. The grid spacing is N_P=20.

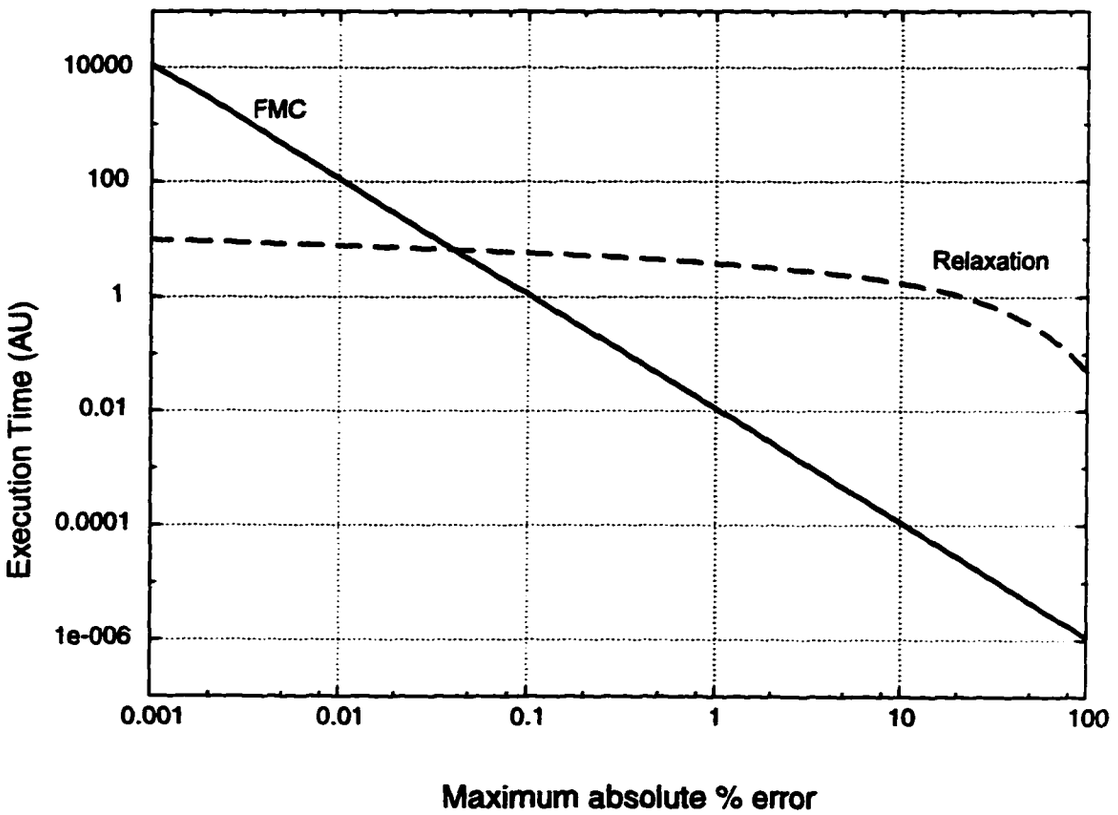

Figure 4.20: Execution time (in arbitrary units) of the Full Monte Carlo and relaxation methods, as a function of the maximum absolute % error. The grid spacing is N_P=20.

methods are shown in figure 4.21. From (3.29) we expect the number of operations to be slightly more than $1/L^3 = 1/8$ that of the FMC. This is indeed verified from figure 4.21.

4.1.4 The Single-Point Monte Carlo Method

We implemented the SPMC method following the procedure outlined in section 3.4.4. The results, along with the timing for the PMC, FMC and relaxation methods are shown in figure 4.22. It is seen that SPMC is about two times faster than PMC. A more dramatic improvement in speed might be expected; after all we perform random walks from only one point in the SPMC method as opposed to many points in the PMC. The reason behind the relatively small speed increase is twofold:

- As seen from figure 3.6 the maximum error occurs no longer at the mid-point but the quarter points ($x = 1/4$ and $x = 3/4$) and is larger than the error at midpoint. Hence a larger number of walks is needed in order to obtain the same level of accuracy in SPMC as that of PMC.

- The grid we selected, $N_P = 20$ is rather coarse. Finer grids, e.g. $N_P = 100$ or more, will result in a much larger speed improvement of SPMC versus PMC.

Before concluding our presentation of the SPMC method, we address another issue: since all the random walks originate from the same point, is there any correlation between potentials calculated from the statistics of these random walks? To answer this question, we plot in figure 4.23 the potential at the mid-point, ϕ_m, versus the potential of its neighbor, ϕ_{m+1}, as calculated with SPMC simulations. It is clear from this plot that there exists a very strong correlation between the two potentials. This fact can be used advantageously when the calculation of electric fields is needed. In figure 4.24 we plot the difference in the two potentials, $\phi_{m+1} - \phi_m$ versus the mid-point potential ϕ_m. The standard deviation in the difference is much smaller than the standard deviation in either potential, a statistical consequence of the very strong correlation displayed in figure 4.23. Since the electric field

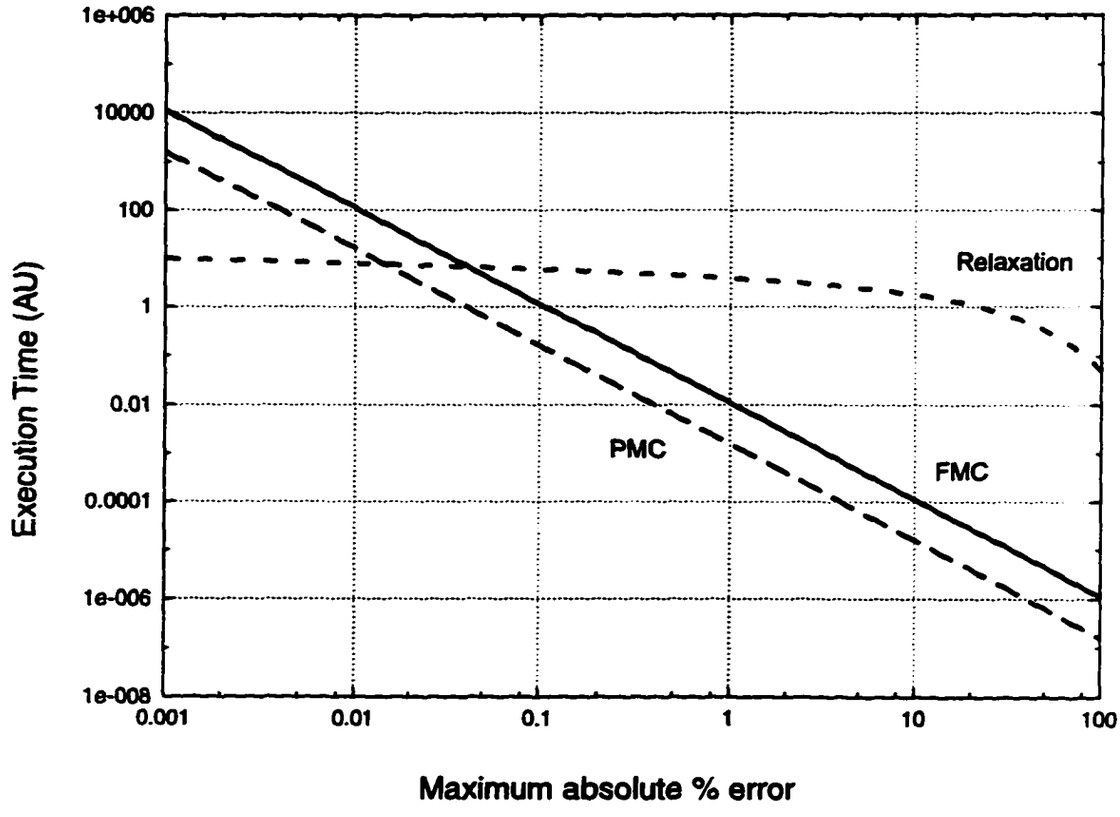

Figure 4.21: Execution time (in arbitrary units) of the Partitioned Monte Carlo, Full Monte Carlo and Relaxation methods, as a function of the maximum absolute % error. The grid spacing is N_P=20.

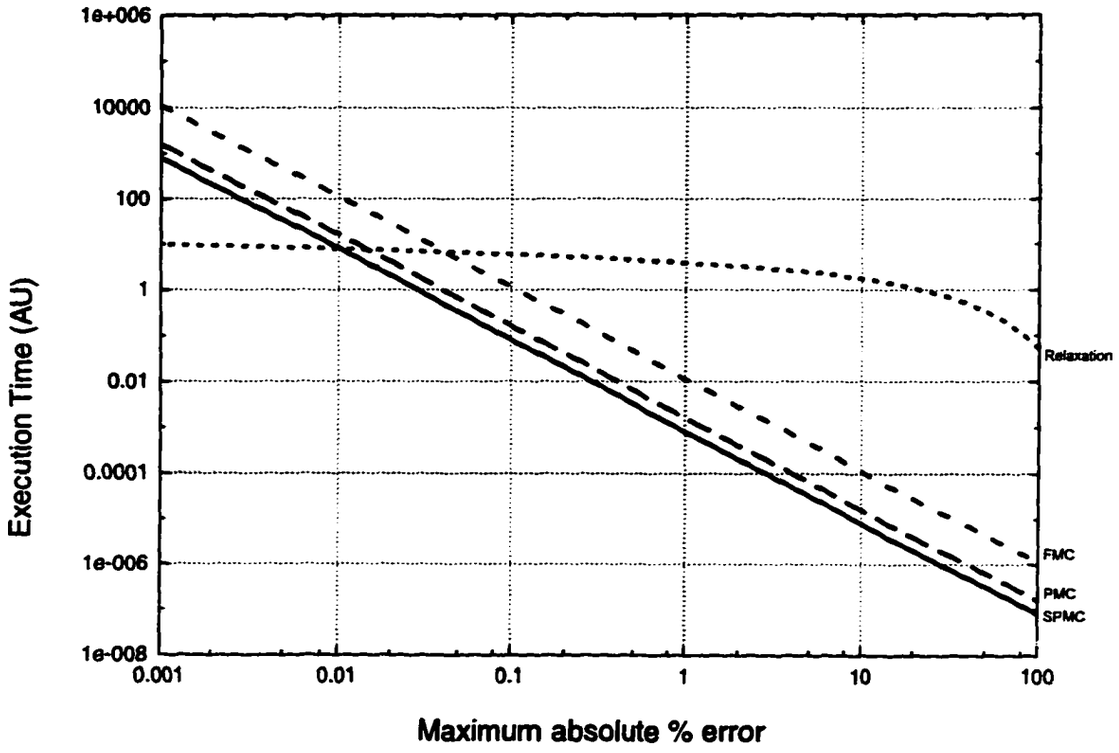

Figure 4.22: Execution time (in arbitrary units) of the Single-Point Monte Carlo, Partitioned Monte Carlo, Full Monte Carlo and Relaxation methods, as a function of the maximum absolute % error. The grid spacing is N_P=20.

is simply the gradient of the potential, this observation is of extreme importance and makes SPMC a very strong candidate for simulations where electric fields are required in well defined areas of a device.

4.1.5 The Monte Carlo Relaxation Method

Finally, we implemented the hybrid MCR method, as outlined in section 3.4.5. We chose to implement the Full Monte Carlo method for the stochastic part of the algorithm. This allows us to concentrate on the issues of the probabilistic versus the deterministic part of the algorithm and not be overly concerned with secondary optimizations. Furthermore we chose the second approach outlined in section 3.4.5, i.e. we started with the Monte Carlo method and when an appropriate level of accuracy was obtained, we switched to the relaxation method for faster convergence. It is obvious from the previous discussion and the results of relaxation convergence, (3.15), that the particular level of accuracy at which the transition occurs depends on the specific problem at hand. Thus, the results we present, shown in figure 4.25, are specific to the one-dimensional case, but still they are highly informative.

It is clear that MCR combines the best of both worlds – stochastic, during the early part of the simulation for initial rapid convergence and deterministic, during the latter part for improved accuracy. Further optimization is still, however possible. For example the SPMC method is faster so a SPMC-Relaxation combination would produce further speed improvements.

4.2 Two-Dimensional Simulations

4.2.1 Rectangular Geometries

We consider now a two-dimensional electrostatic problem that has been presented previously in the literature [109, 1], namely the solution to Laplace's equation in the rectangle $[0, a] \times [0, b]$ subject to the boundary conditions $\phi(x, b) = 1000\text{V}$ and $\phi = 0$ everywhere

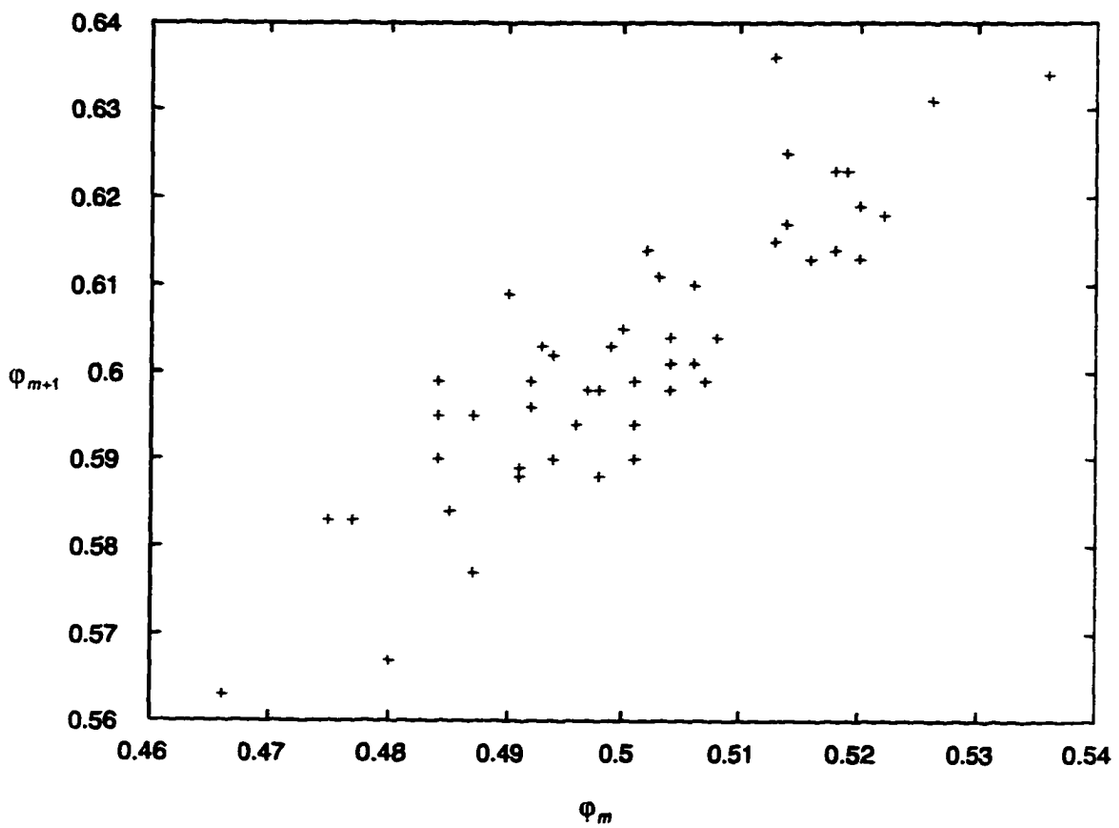

Figure 4.23: Correlation between the potential at mid-point, ϕ_m, versus the potential of its neighbor, ϕ_{m+1}. The grid spacing is N_P=20.

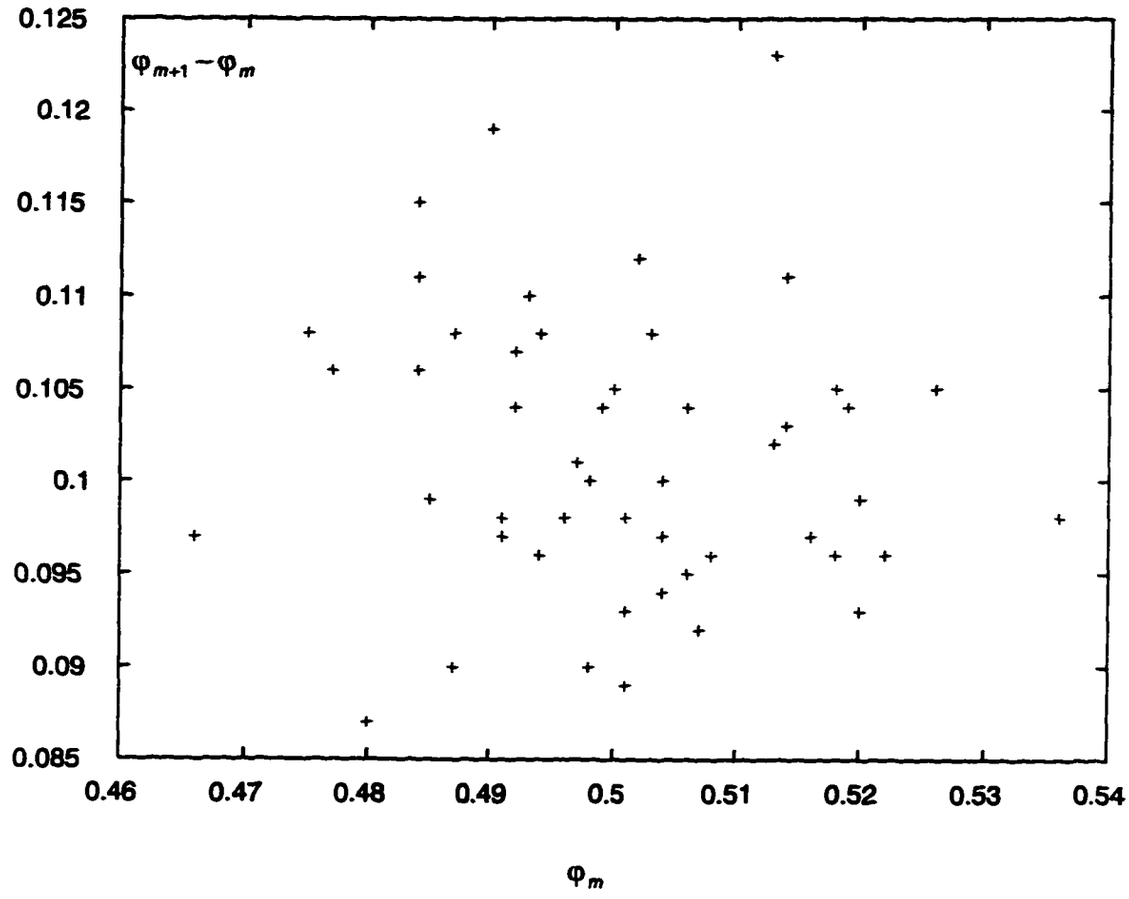

Figure 4.24: Difference in the two potentials, $\phi_{m+1} - \phi_m$ versus the mid-point potential ϕ_m. The grid spacing is N_P=20. Note the difference in scale of the two axes.

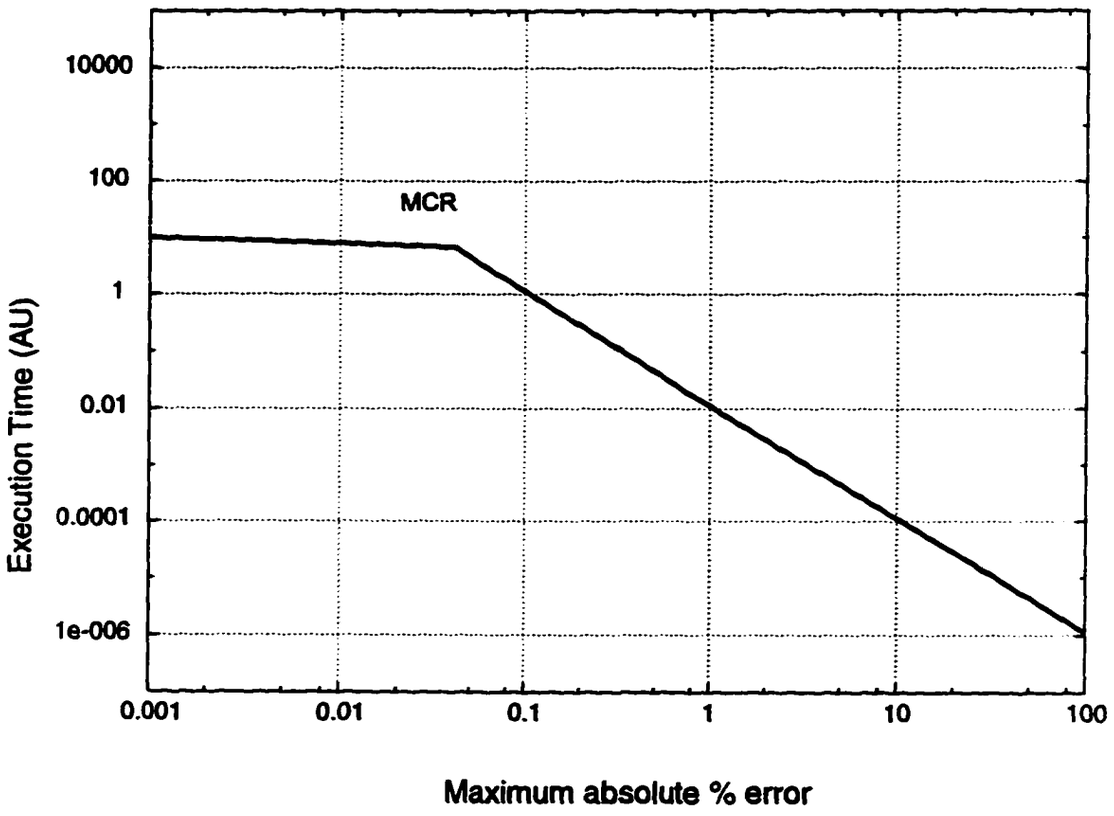

Figure 4.25: Execution time (in arbitrary units) of the Monte Carlo Relaxation method, as a function of the maximum absolute % error. The grid spacing is N_P=20.

else. As in [109, 1], we selected $a = 20$ and $b = 10$. This problem has an analytic solution [1]:

$$\phi(x,y) = \frac{4V_0}{\pi}\sum_{n=0}^{\infty}\frac{\sinh(ky)\sin(kx)}{(2n+1)\sinh(kb)}, \tag{4.3}$$

where $k = (2n+1)\pi/a$ and $V_0 = 1000$. This exact solution is shown in figure 4.26.

We implemented a Full Monte Carlo (FMC) solution of this problem. First, we will concentrate on the average number of steps to reach any boundary point for a random walk that starts at (x,y). It is shown in appendix A that this function, $U(x,y)$, satisfies the equation:

$$\nabla^2 U = -\frac{4}{h^2}, \tag{4.4}$$

where $U = 0$ on the boundary and h is the grid spacing. Equation (4.4) can be readily solved via the Green's function technique [1] to yield:

$$U(x,y) = \frac{64h^2}{ab}\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\frac{1}{kg}\frac{\sin(kx)\sin(gy)}{k^2+g^2}, \tag{4.5}$$

where $k = (2n+1)\pi/a$ and $g = (2m+1)\pi/b$.

Now, this is a doubly infinite series which, however, converges rapidly. We show in figure 4.27 the first few terms of this series along the x-direction for a plane that passes through the middle of the rectangle at $y = b/2 = 5$. It is seen from this figure and numerical calculations we performed that the series converges rapidly. We do not make a significant error, therefore, if we maintain only the first 4 terms in the summation, for n and m ranging from 0 to 1.

In figure 4.28 we show the first 4 terms of (4.5) taking into account both its x and y dependence and in figure 4.29 we show the corresponding data from our FMC simulations. The differences observed are clearly due to the statistical nature of Monte Carlo simulations.

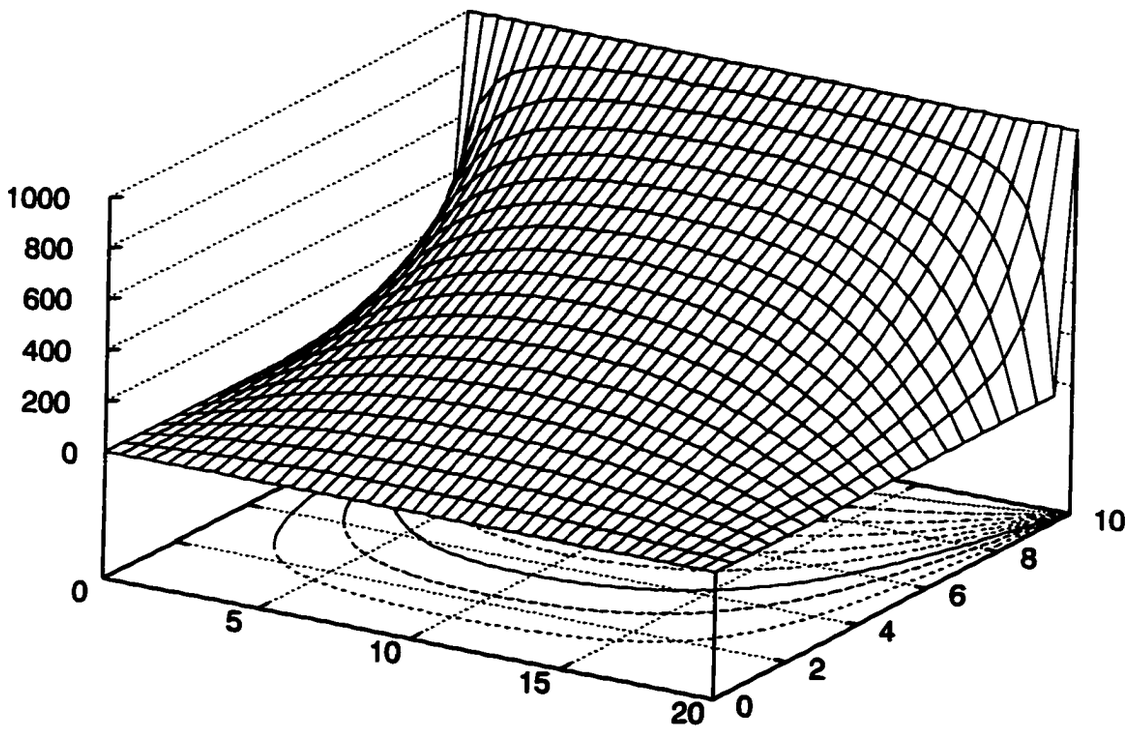

Figure 4.26: Exact solution (equation 4.3) to prototype rectangular problem. The curves at the bottom of the figure are equipotential contours in steps of 100V.

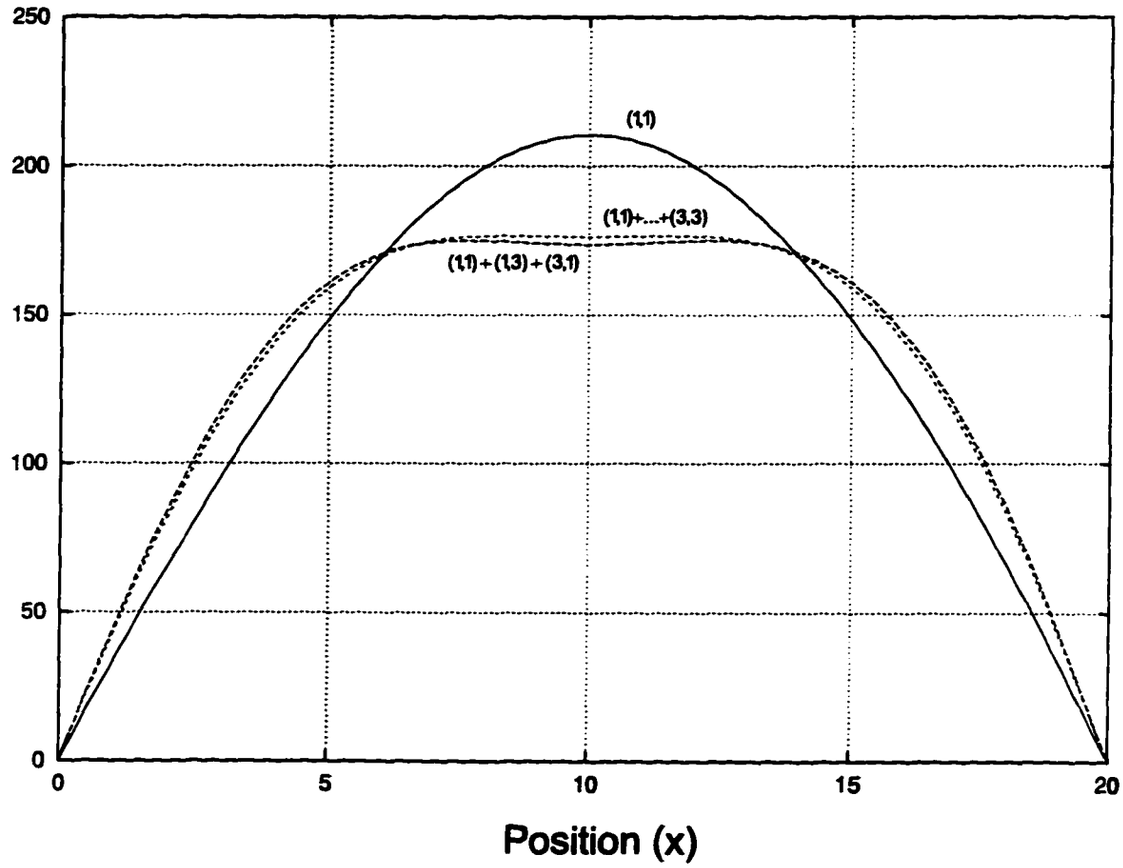

Figure 4.27: The first few terms of (4.5) along the line $(x, b/2)$. The series converges rapidly close to the value obtained by maintaining only the first 4 terms.

Our FMC simulation results are shown in the sequence of figures 4.30-4.33 for numbers of random walks varying from $N_W = 10$ to $N_W = 10^4$. The corresponding execution times varied from $T < 1$ sec. to $T = 311$ sec. To maintain grid uniformity the spacing used was $h = 0.5$ and the grid itself was 41×21.

Inspection of figures 4.26 and 4.33, which represent the exact and FMC solution with $N_W = 10^4$, respectively, reveals the accuracy of the method. More precise comparison is shown in the sequence of figures 4.34-4.37, where the potential along the y direction is shown for a plane passing through $x = a/2$. On these figures we compare not only the FMC solution versus the exact solution, but we also include relaxation data published by Gash [1]. The superior performance of FMC versus relaxation alone is clearly evident. On the other hand, when we duplicated the work of Gash [1] by implementing the relaxation method for this problem, it was found that it took $T = 198$ sec. for the method to converge within a tolerance of 2 Volts, the same value used by Gash [1].

Finally, we show in figure 4.38 the fraction of random walks that pass through point (x, y), having started at the center of the rectangle $(a/2, b/2)$. It is seen from the figure that near the center this fraction is isotropic and approximately equal to $1/4$ but as the distance increases, this isotropy is destroyed and the fraction is heavily influenced by the geometry of the domain. In addition, this value becomes very small, clearly indicating the unsuitability of Single-Point Monte Carlo (SPMC) if the potential is sought in the entire domain. If the potential is needed, however, only near the center of the domain, SPMC is extremely efficient. Using the statistics obtained from $N_W = 10^3$ random walks, we performed a SPMC simulation along the plane $(a/2, y)$. Although the overall results are not as good as those of the corresponding FMC with $N_W = 10^3$ (see figure 4.36), the agreement near $y = b/2$ is excellent and the execution time was only $T = 4$ sec., compared to $T = 31$ sec. for the FMC and $T = 198$ sec. for the relaxation method. These results are shown in figures 4.39 and 4.40 and a comparison of the Relaxation and Monte Carlo methods for this case is shown in table 2.

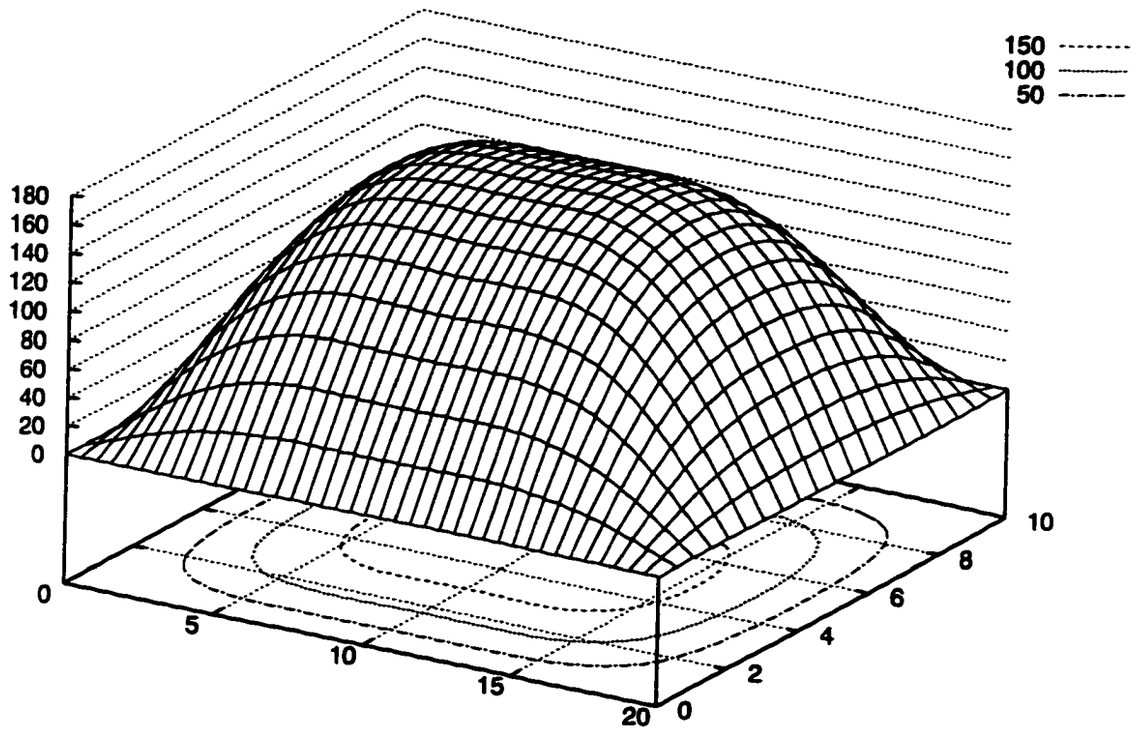

Figure 4.28: Two dimensional representation of (4.5), maintaining only the first 4 terms. The surface at any point (x, y) denotes the average number of steps for a random walk starting at (x, y) to reach the boundary. The curves at the bottom of the figure are constant value contours.

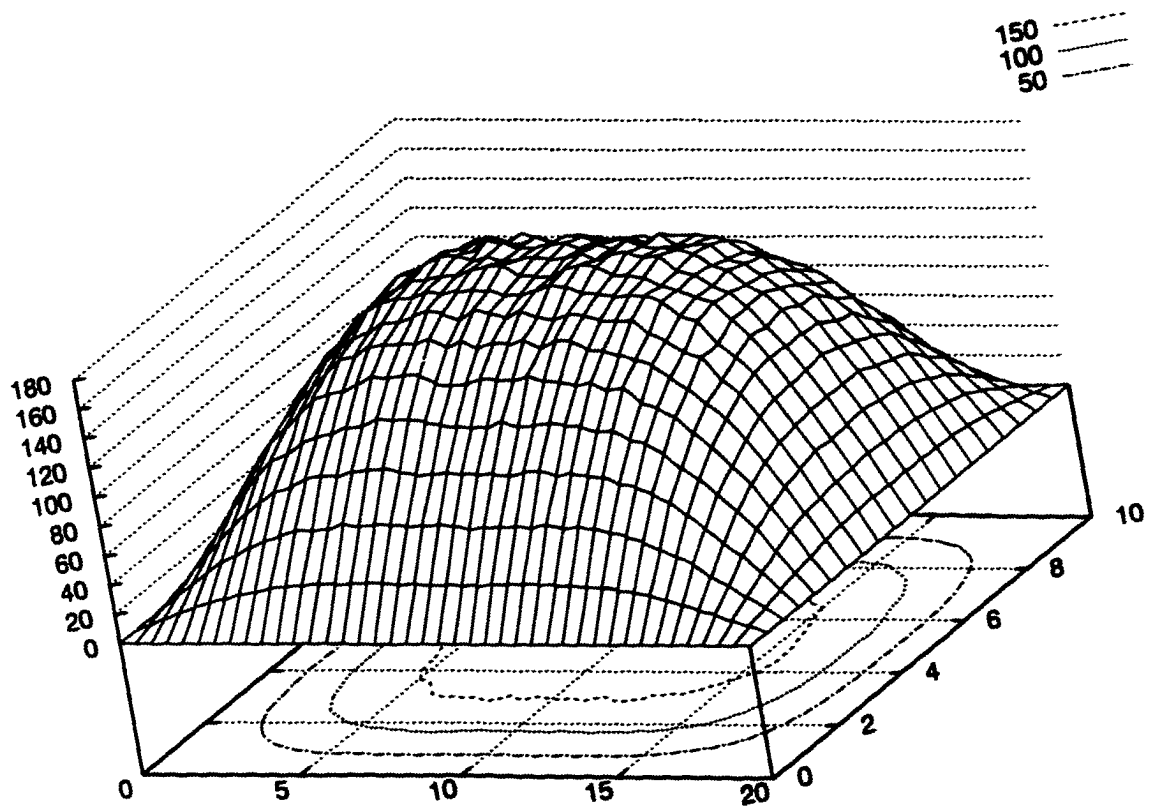

Figure 4.29: FMC simulation results for the number of steps to reach any boundary. These results where obtained from $N_W = 10^4$ random walks. The curves at the bottom of the figure are constant value contours.

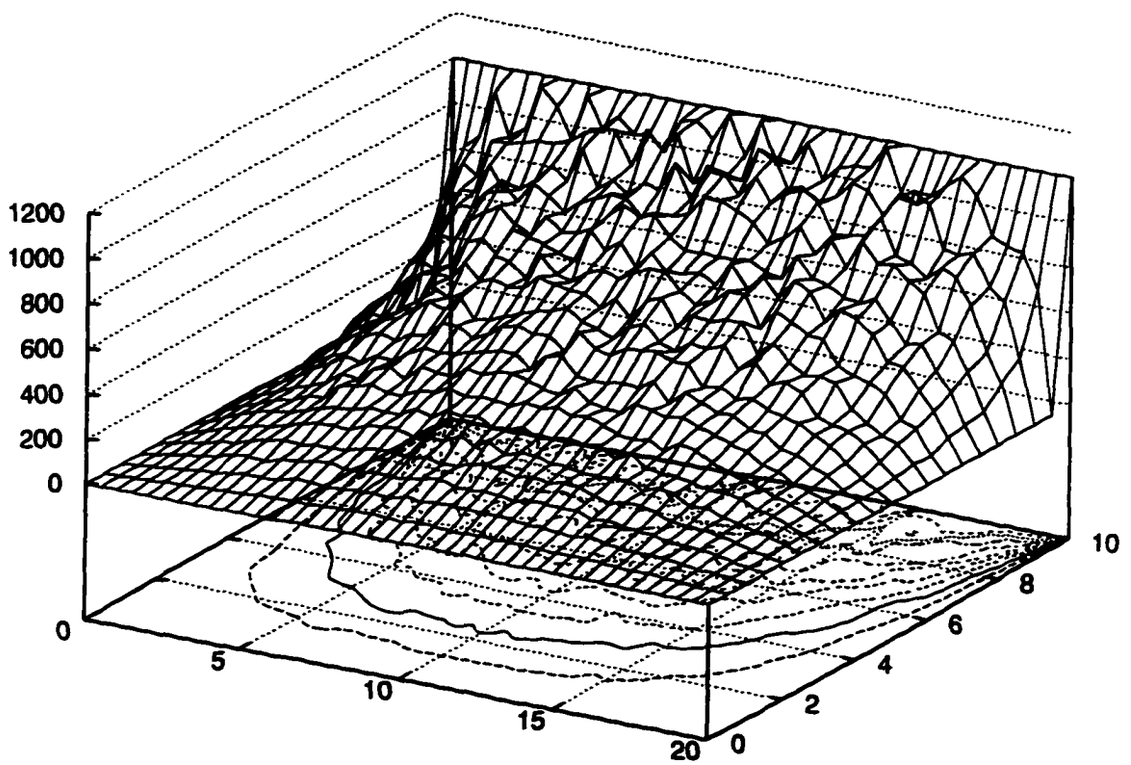

Figure 4.30: FMC simulation results for the prototype rectangular problem. There were $N_W = 10$ random walks started per point and the underlying grid was 41×21. Total execution time was $T < 1$ sec. The curves at the bottom of the figure are equipotential contours in 100V increments.

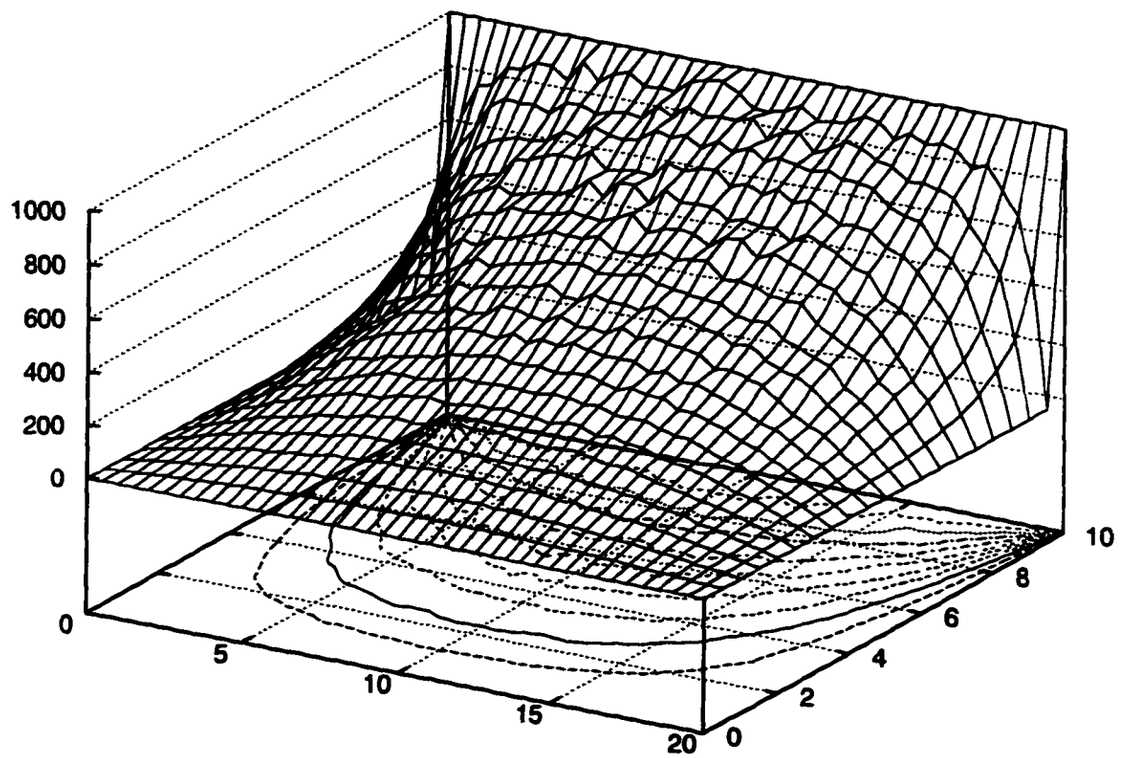

Figure 4.31: FMC simulation results for the prototype rectangular problem. There were $N_W = 100$ random walks started per point and the underlying grid was 41×21. Total execution time was $T = 3$ sec. The curves at the bottom of the figure are equipotential contours in 100V increments.

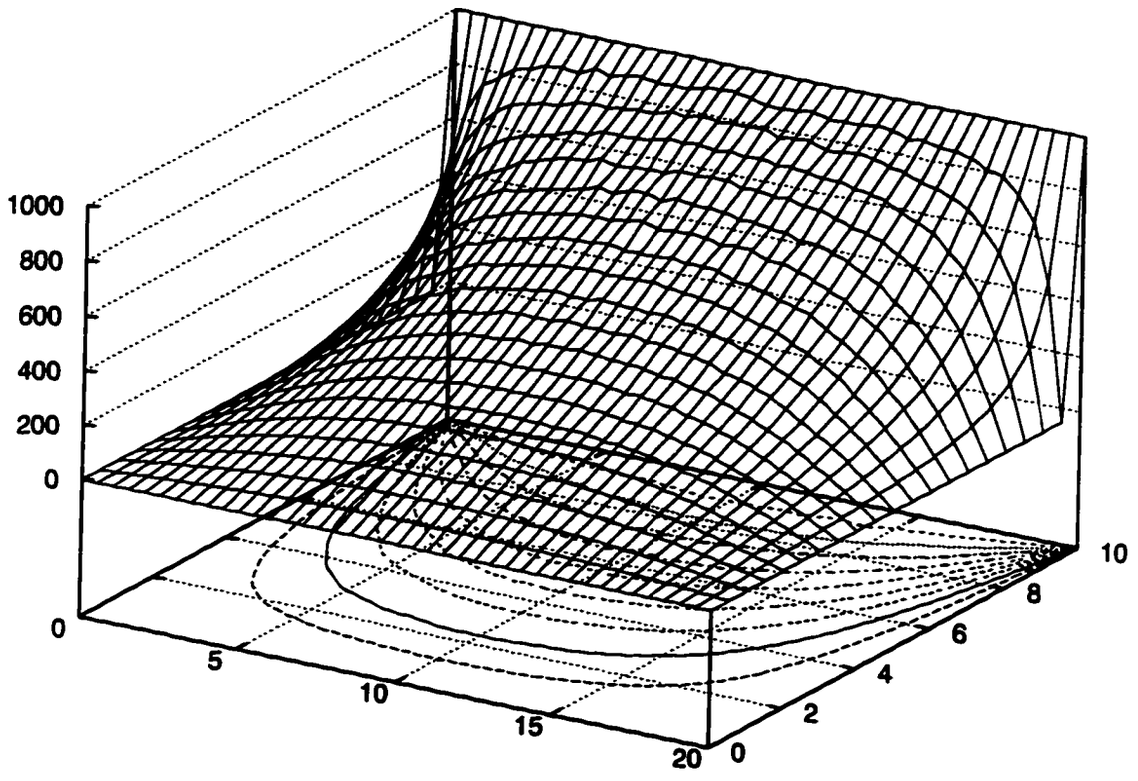

Figure 4.32: FMC simulation results for the prototype rectangular problem. There were $N_W = 10^3$ random walks started per point and the underlying grid was 41×21. Total execution time was $T = 31$ sec. The curves at the bottom of the figure are equipotential contours in 100V increments.

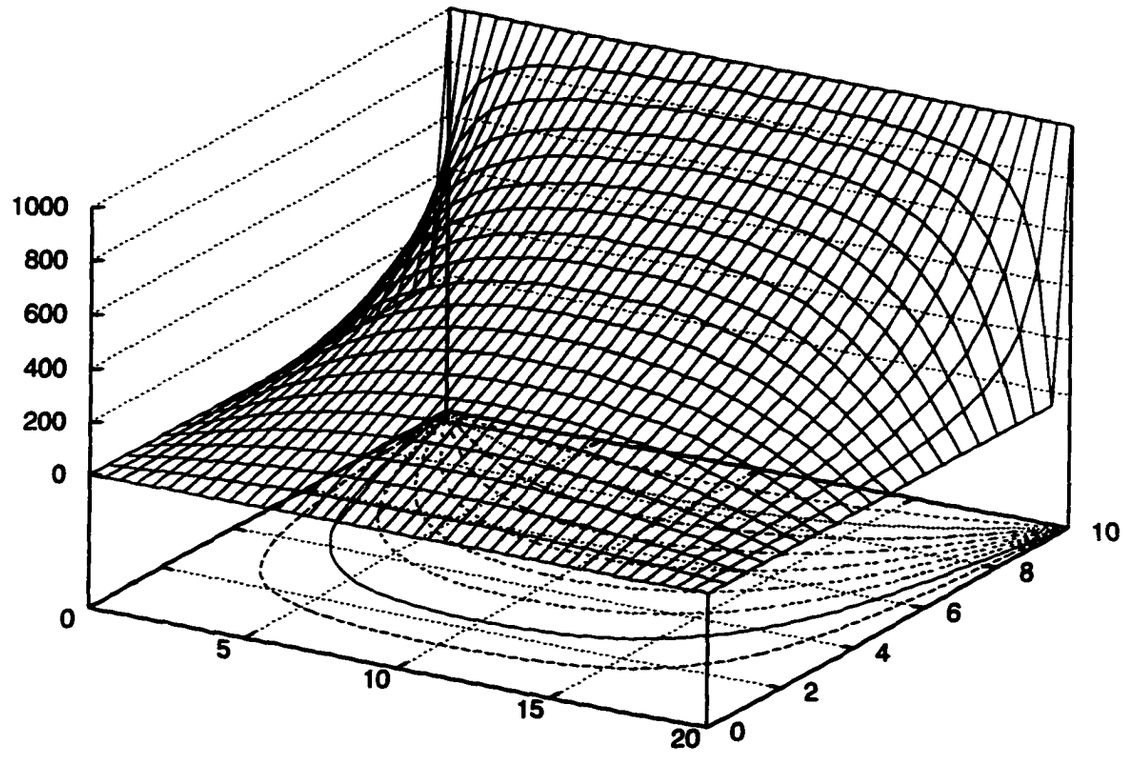

Figure 4.33: FMC simulation results for the prototype rectangular problem. There were $N_W = 10^4$ random walks started per point and the underlying grid was 41×21. Total execution time was $T = 311$ sec. The curves at the bottom of the figure are equipotential contours in 100V increments.

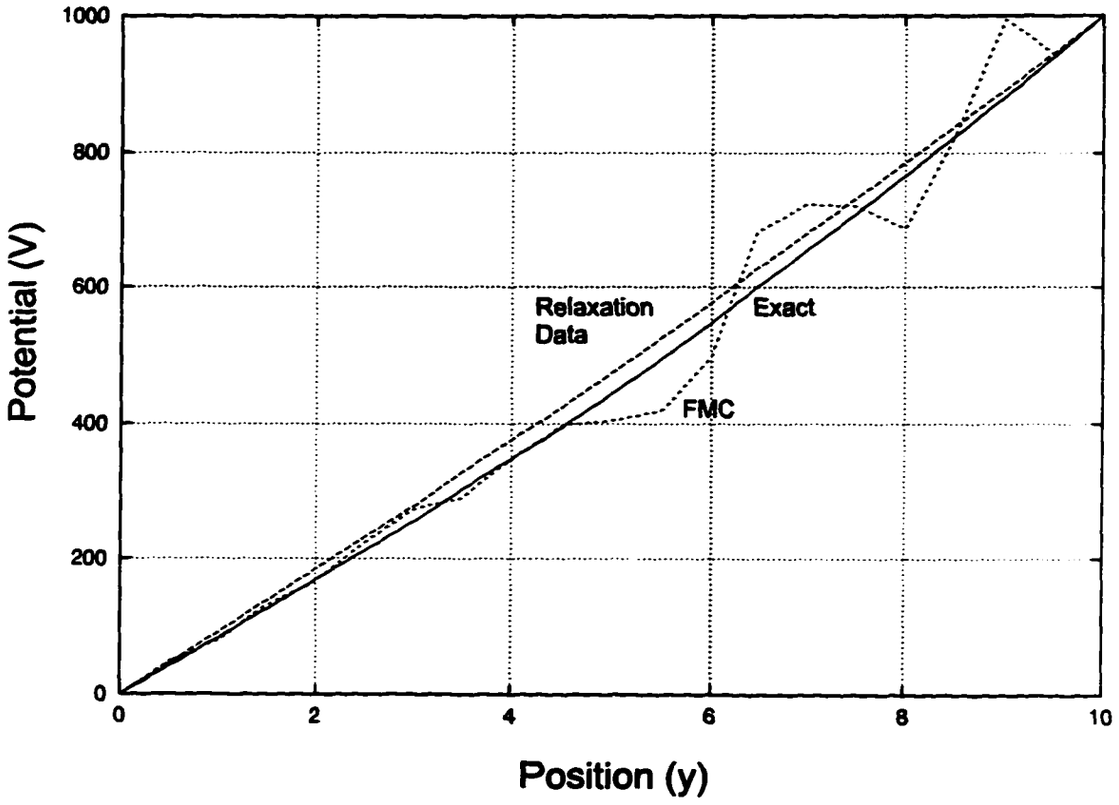

Figure 4.34: FMC simulation results for the prototype rectangular problem. There were $N_W = 10$ random walks started per point and the underlying grid was 41×21. The result along the plane $(a/2, y)$ is shown. Relaxation data from Gash [1] are also included.

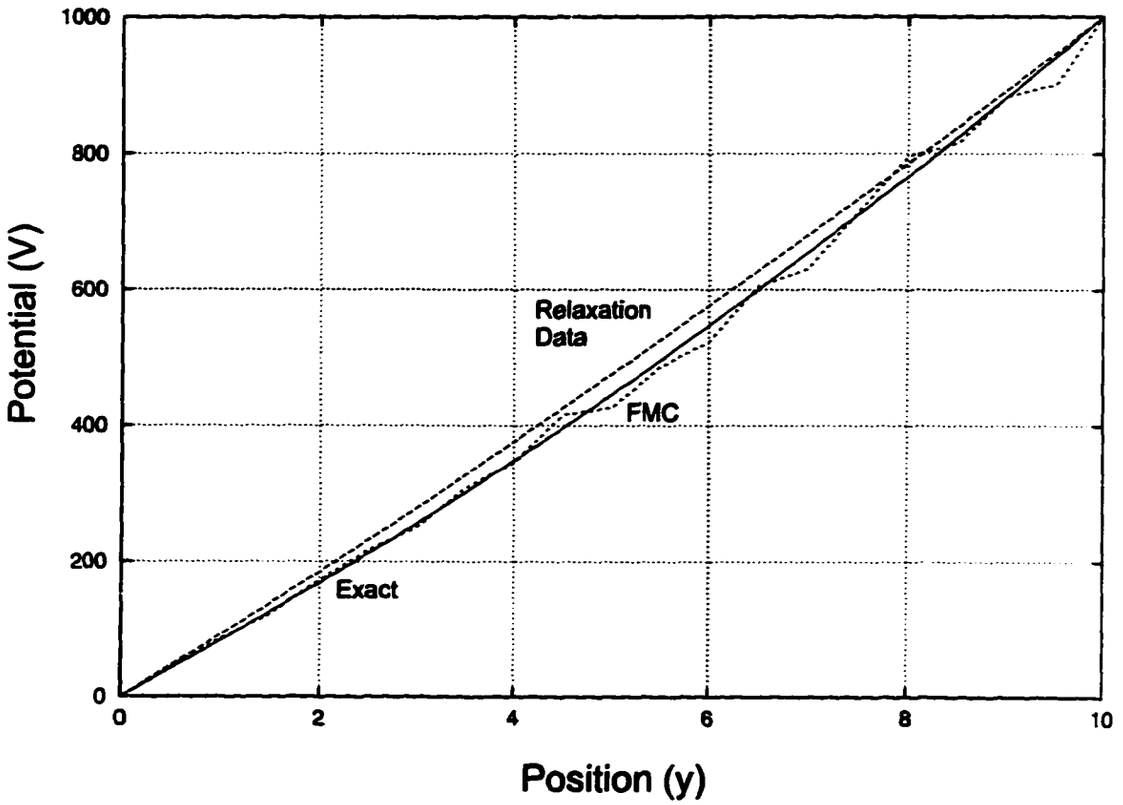

Figure 4.35: FMC simulation results for the prototype rectangular problem. There were $N_W = 100$ random walks started per point and the underlying grid was 41 × 21. The result along the plane $(a/2, y)$ is shown. Relaxation data from Gash [1] are also included.

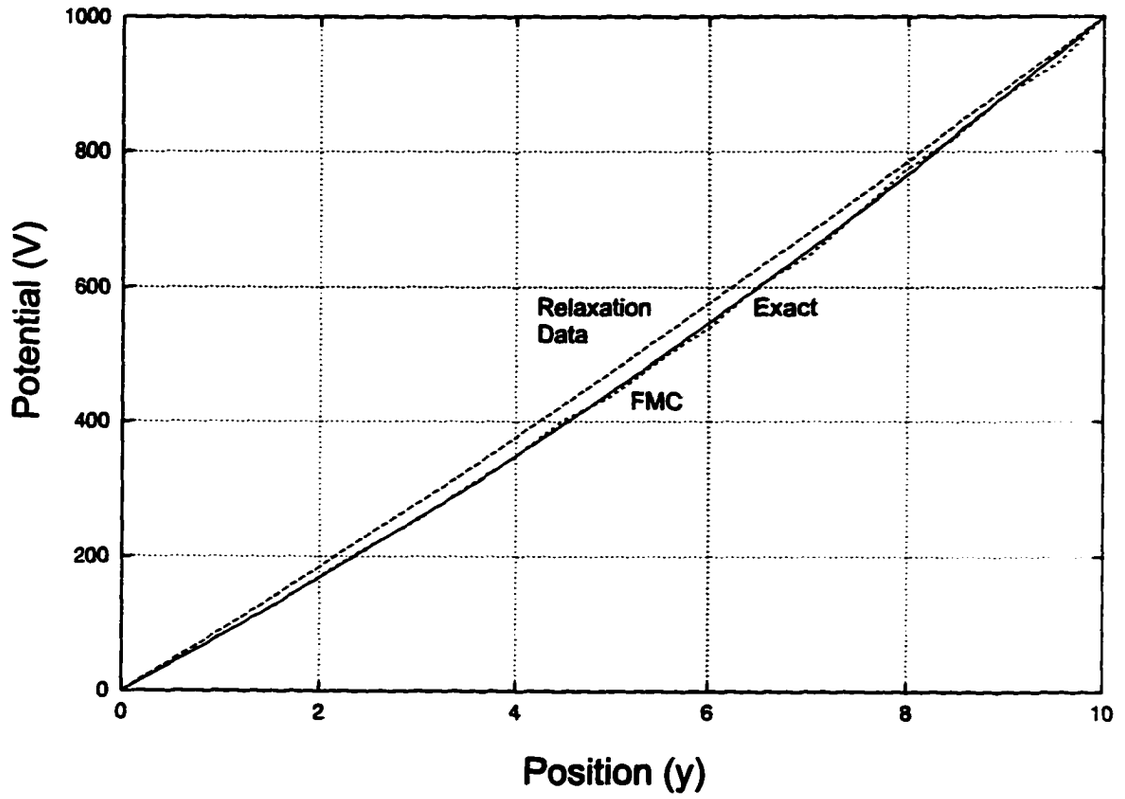

Figure 4.36: FMC simulation results for the prototype rectangular problem. There were $N_W = 10^3$ random walks started per point and the underlying grid was 41 × 21. The result along the plane $(a/2, y)$ is shown. Relaxation data from Gash [1] are also included.

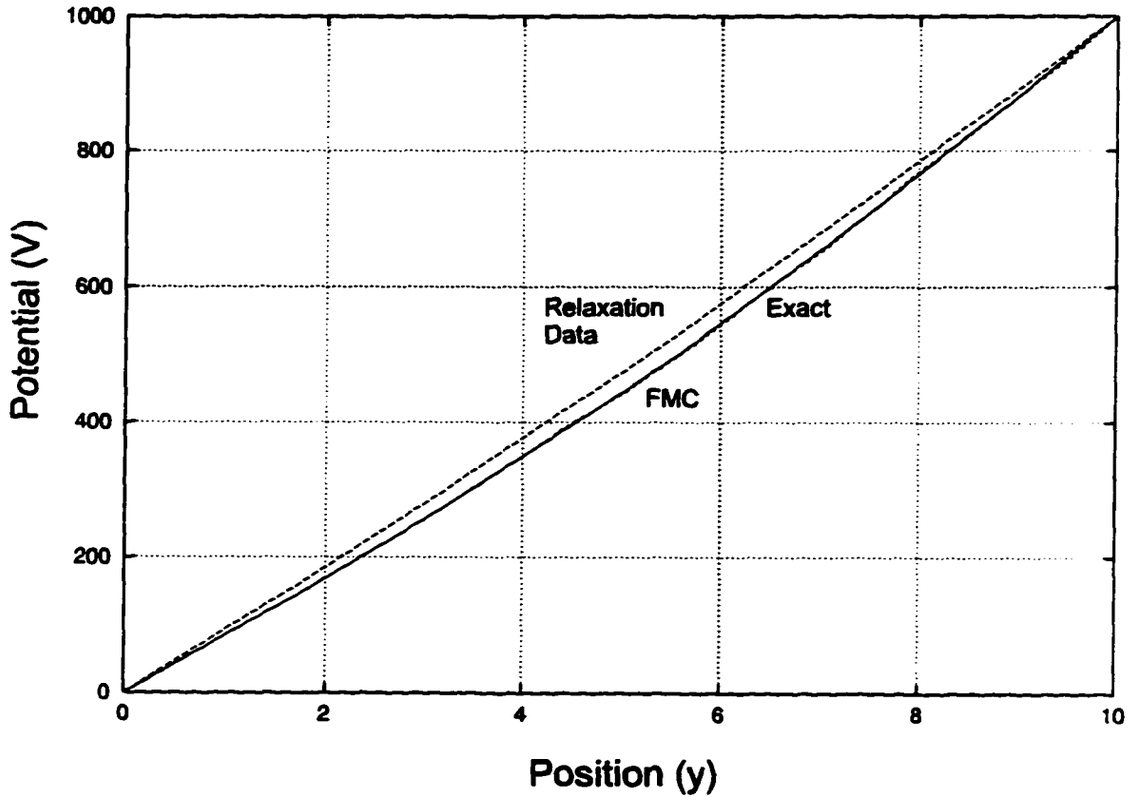

Figure 4.37: FMC simulation results for the prototype rectangular problem. There were $N_W = 10^4$ random walks started per point and the underlying grid was 41×21. The result along the plane $(a/2, y)$ is shown. Relaxation data from Gash [1] are also included.

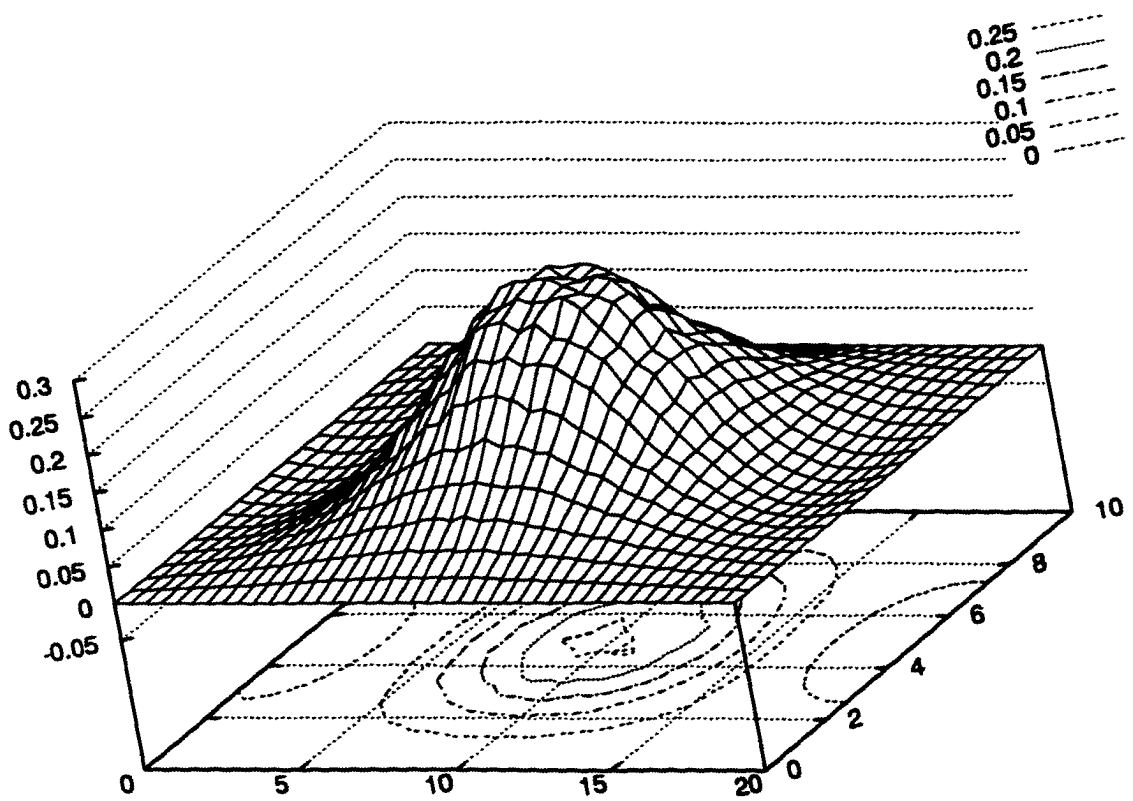

Figure 4.38: Fraction of random walks that pass through point (x, y) having started at the center of the rectangle. The data were collected for $Nw = 10^3$.

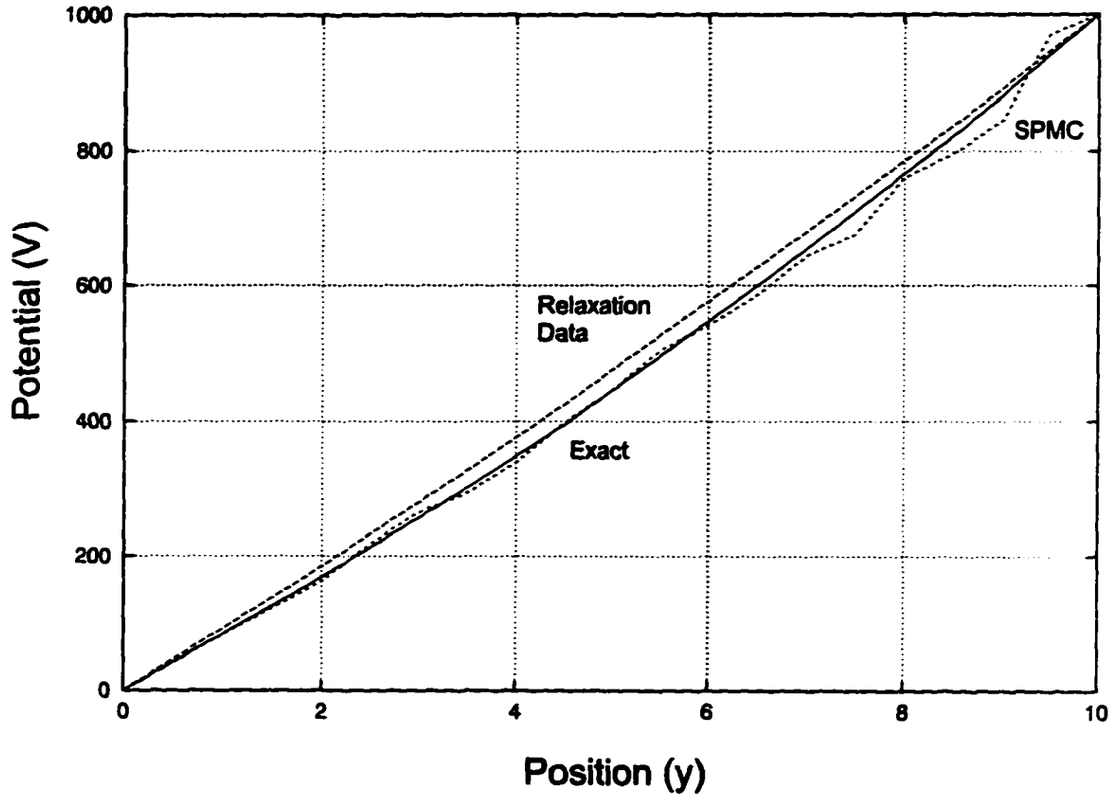

Figure 4.39: SPMC simulation results for the prototype rectangular problem. There were $N_W = 10^3$ random walks started per point and the underlying grid was 41 × 21. The result along the plane $(a/2, y)$ is shown. Relaxation data from Gash [1] are also included.

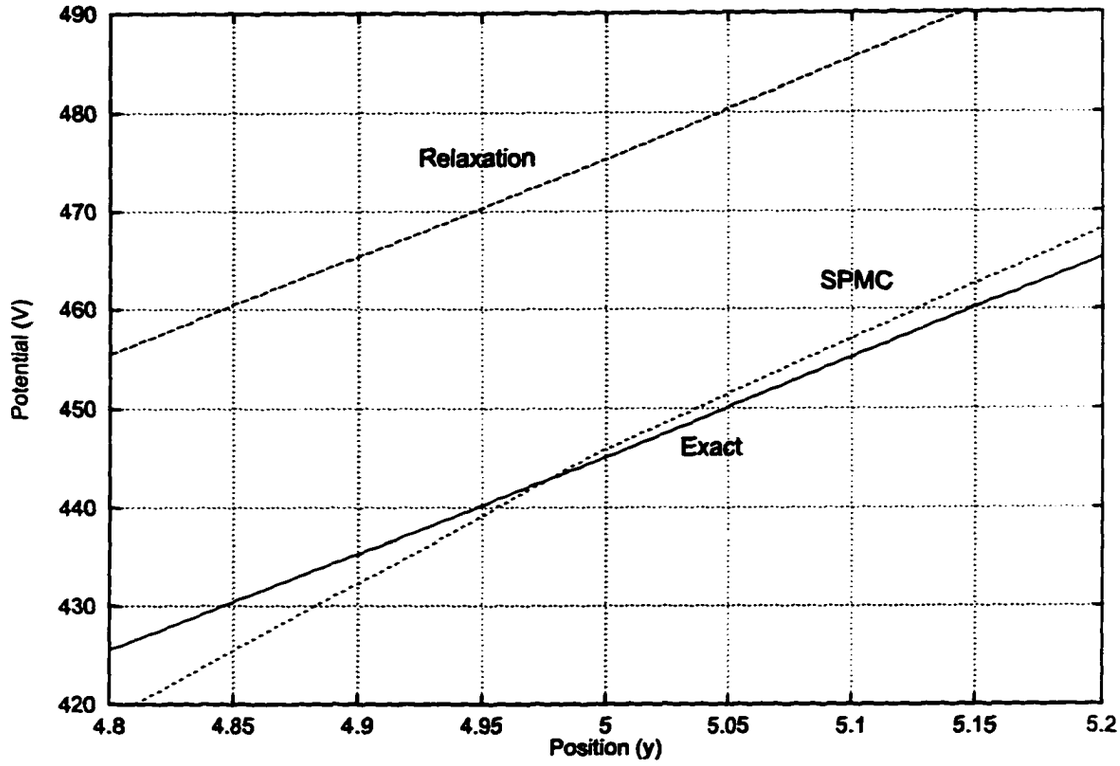

Figure 4.40: Detailed view of the center point region of figure 4.39.

Table 2: Comparison of Relaxation versus SPMC methods for a prototype rectangular problem [1].

Method	Relaxation	SPMC
Time (sec.)	198	4
Error (Volts)	30.2	1

4.2.2 Circular Geometries

As a further demonstration of the FMC method's applicability, we consider electrostatic lenses and deflectors with circular cross-sections. If the length of the lens is large compared to the lateral dimensions of the electron beam through the device, the geometry can be considered as essentially two-dimensional. An example of a quadrupole lens [110] is shown in figure 4.41. The circular circumference of the lens is divided in four segments, with each segment held at a voltage V or $-V$ and the radius of the device has been normalized to unity.

We implemented a Full Monte Carlo solution to determine the electrostatic potential in the interior of the lens when the applied voltage was ± 100 Volts. It should be emphasized here that the problem was treated in cartesian coordinates, (x, y), even though the cross-section is circular and a more appropriate choice would be polar coordinates (ρ, ϕ). In polar form, however, the Laplacian is:

$$\nabla^2 = \frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho \frac{\partial}{\partial \rho}\right) + \frac{1}{\rho^2}\frac{\partial^2}{\partial \phi^2}. \tag{4.6}$$

In this form the Laplacian forms a general linear elliptic operator of order two, the study of which we undertake in chapter 5.

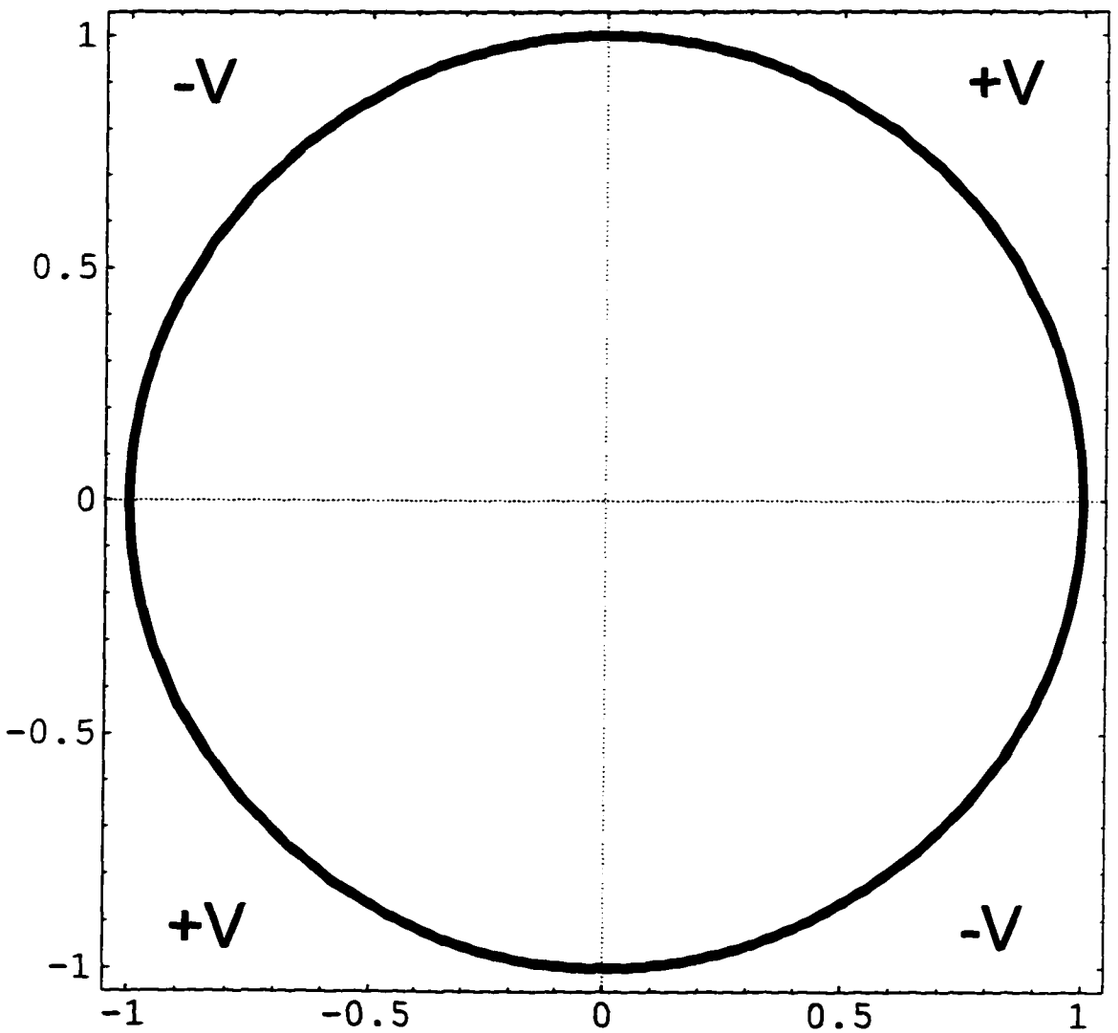

Figure 4.41: Cross-section of a quadrupole lens. The lens consists of four segments, each held at a constant potential, as indicated.

The FMC solution for the quadrupole lens, with $h = 0.025$, corresponding to an 81×81 grid, is shown in figure 4.42. The lens action of the device is clear from the form of the equipotential contours: any deviation from the center of the lens results in a force, due to the potential gradient, that focuses the electron beam.

This potential can be also found analytically. The solution is [11]:

$$\Phi = \frac{2V}{\pi} \tan^{-1}\left(\frac{2\rho^2 \sin(2\phi)}{1-\rho^4}\right), \tag{4.7}$$

where $\rho^2 = x^2 + y^2$ and $\phi = \tan^{-1}(y/x)$. A plot of the analytical solution is shown in figure 4.43. Comparison of figures 4.42 and 4.43 reveals the high precision of the FMC method: the two plots are virtually indistinguishable.

4.2.3 Effective Use of FMC in Deflector Simulations

Finally, as a further demonstration of the flexibility of FMC, we consider an electrostatic deflector described by Szilagyi [69, 111]. A cross-section of the deflector is shown in figure 4.44. The deflector consists of eight equal segments, each held at a potential determined by a linear combination of two externally applied voltages, V_x and V_y. The variable p is used to adjust the "mixing" of the two potentials to obtain the desired deflection characteristics. In the subsequent discussion, $V_x = 100\text{V}$ and $V_y = 50\text{V}$, with p being variable.

We implemented an FMC solution of this octupole deflector. Instead of solving the problem from the beginning, however, as would be necessary with any other deterministic method commonly used, we reused the statistics obtained during the solution of the quadrupole lens described earlier in this section. Although the initial solution for the quadrupole required a CPU time of 875 sec., due partially to the fine spacing of the grid (81×81) and the large number of random walks ($N_W = 10^4$), every subsequent calculation required less than 1 sec. of CPU time, the resolution of the timer used. It should be also noted that no use of symmetry was made in our calculations. In a computationally intensive environment, the high degree of symmetry of a circle would be used to reduce the CPU

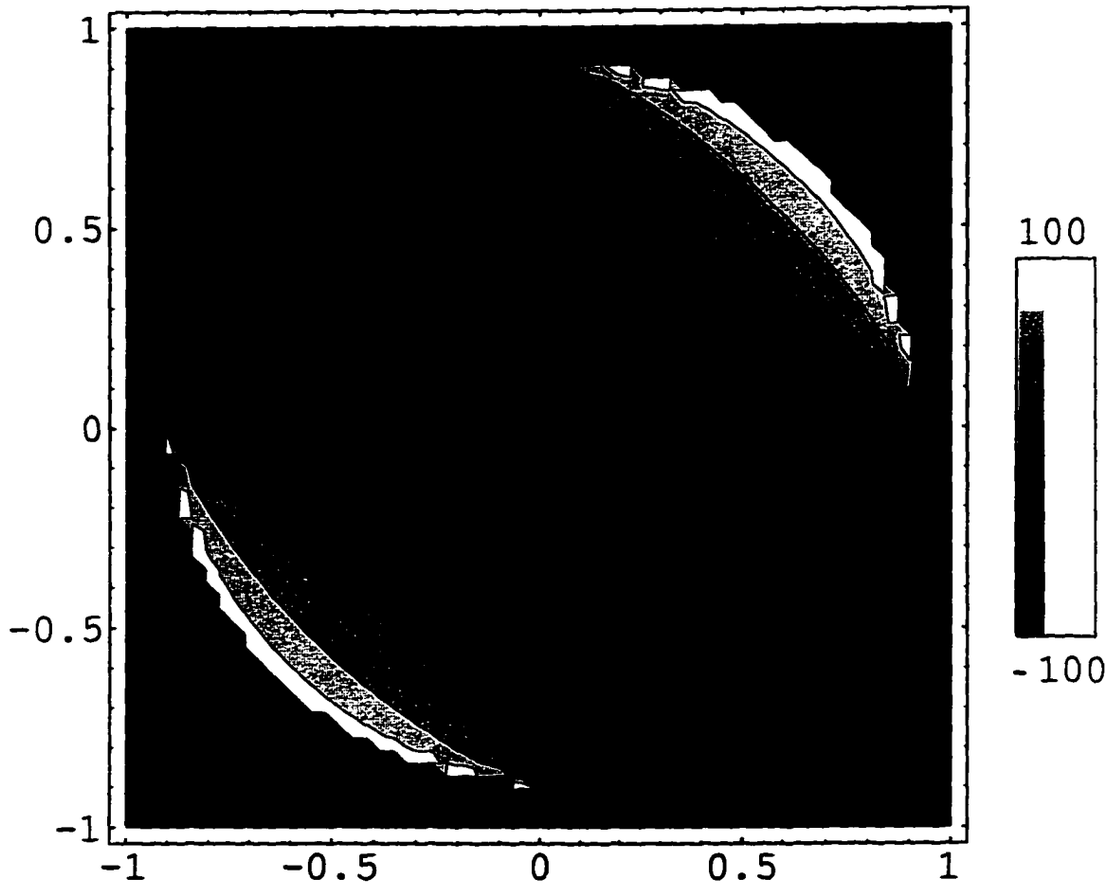

Figure 4.42: FMC solution of the quadrupole lens of figure 4.41. The applied voltage is $V = 100$V, the grid spacing is $h = 0.025$ and there were 10^4 random walks started per point. The curved lines show equipotential contours.

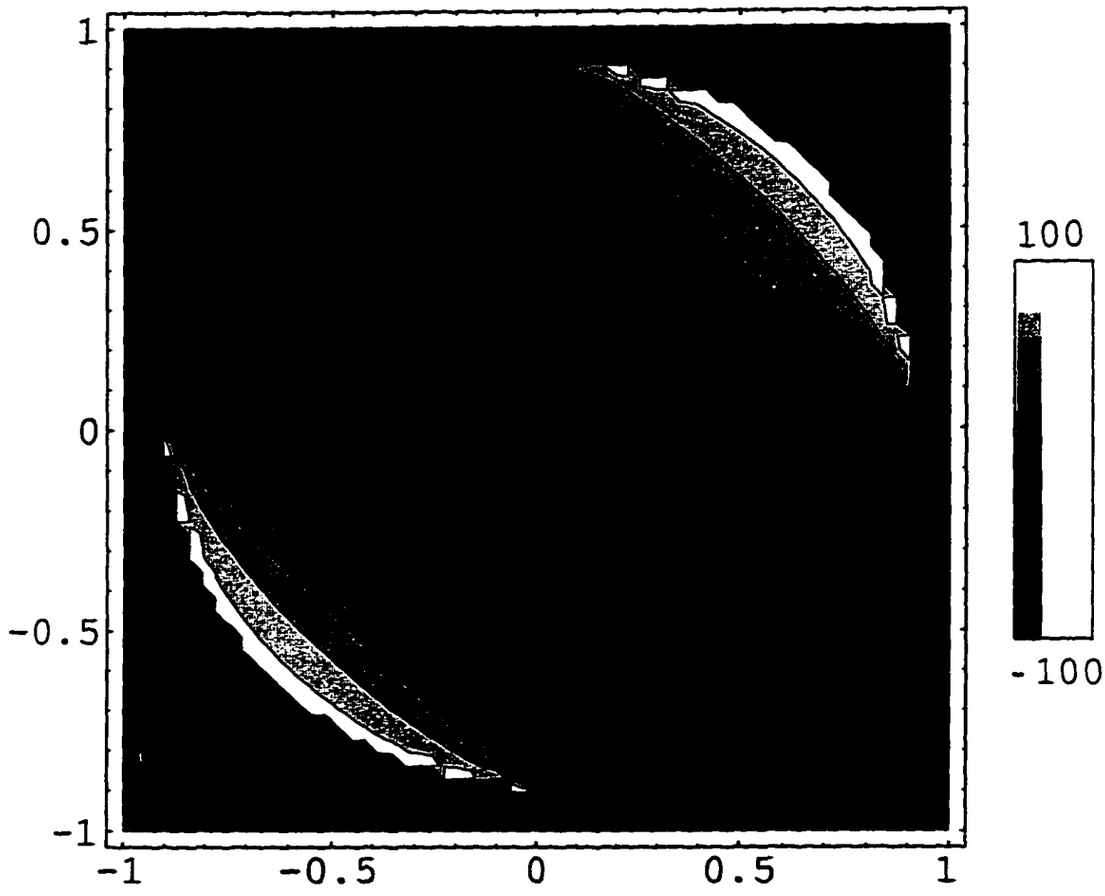

Figure 4.43: Analytical solution of the quadrupole lens of figure 4.41.

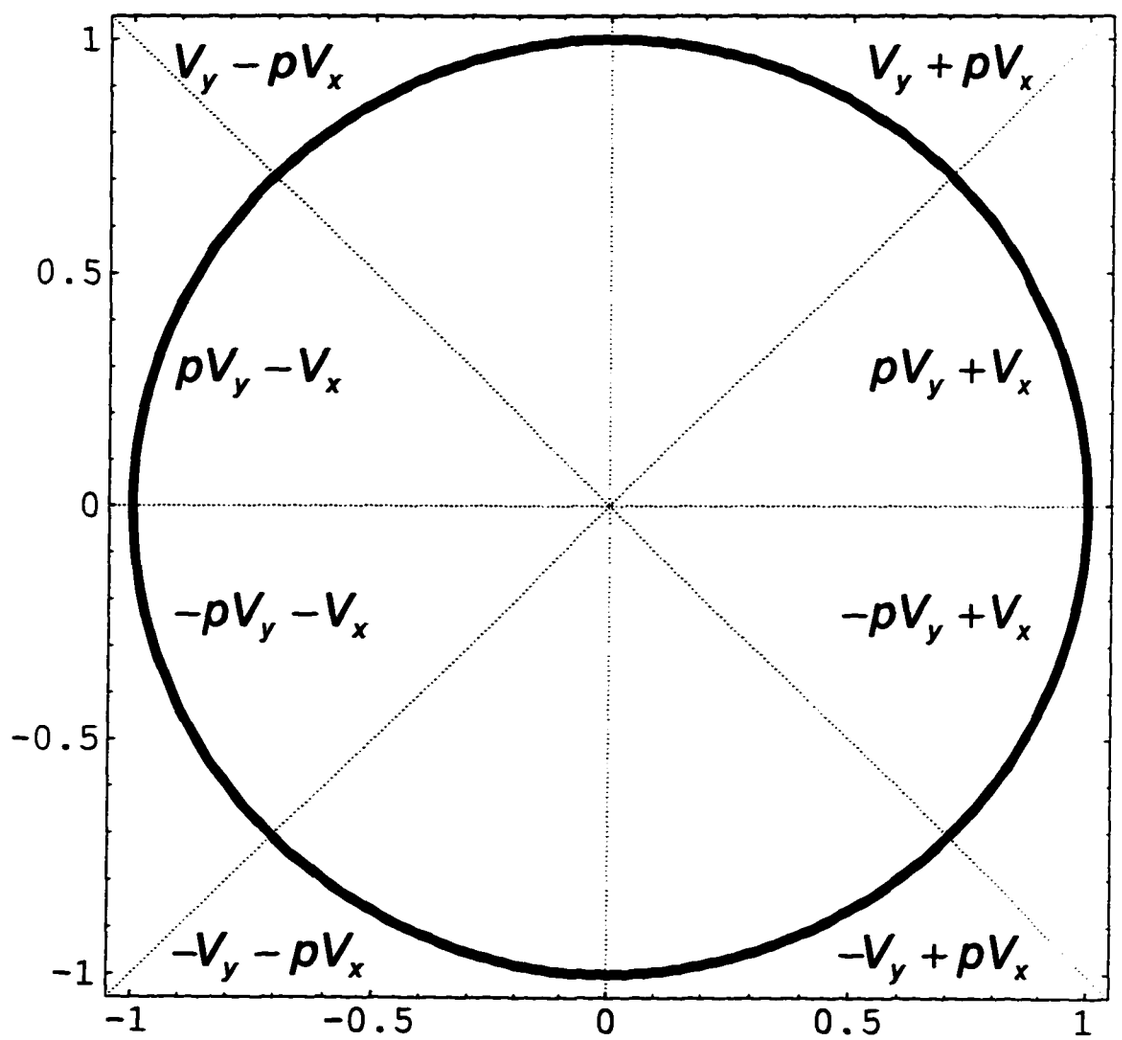

Figure 4.44: Cross-section of an octupole deflector. The lens consists of eight segments, each held at a constant potential, as indicated.

time by factors of 2, 4 or more. The advantageous use of symmetry is unique to the Monte Carlo method because it depends exclusively on the symmetry properties of the geometry of the domain and not on the applied boundary conditions, as is necessary with other, conventional techniques. For the circular geometries we examine in this section, for example, it will be sufficient to perform random walks and collect statistics along a single radius and use the same statistics for the other parts of the device. Thus random walks from only 41, instead of $81 \times 81 = 6561$ grid points will suffice, resulting in dramatic decrease in computational time. For this approach to be viable, however, the grid must be conformal to polar and not rectangular coordinates. How the Monte Carlo method is extended in this case is shown in chapter 5.

FMC solutions for values of $p = 0$, $p = \sqrt{2} - 1$ and $p = 1$ are shown in figures 4.45-4.47. The case for $p = 1$ corresponds to a quadrupole deflector, as seen from figure 4.47. The case for $p = \sqrt{2} - 1$ corresponds to a true octupole deflector and it can be shown that it results in the most uniform deflection field [111] as is clearly seen from the flat and uniform equipotential contours in figure 4.46. From this discussion the power of FMC as a design tool is apparent.

4.3 Summary

We presented results of numerical experiments in which our algorithms were tested. We performed an exhaustive analysis of one-dimensional prototype simulations and used the results as guides for more intensive (and more interesting) two-dimensional simulations. We also demonstrated the viability of our method by solving Laplace's equation stochastically in non-trivial geometries and compared our results and execution times with traditional approaches. Where applicable, we compared with analytically known solutions. Finally, we demonstrated the practicality of our work by applying it to the simulation of actual electrostatic deflectors and lenses which have been presented in the literature.

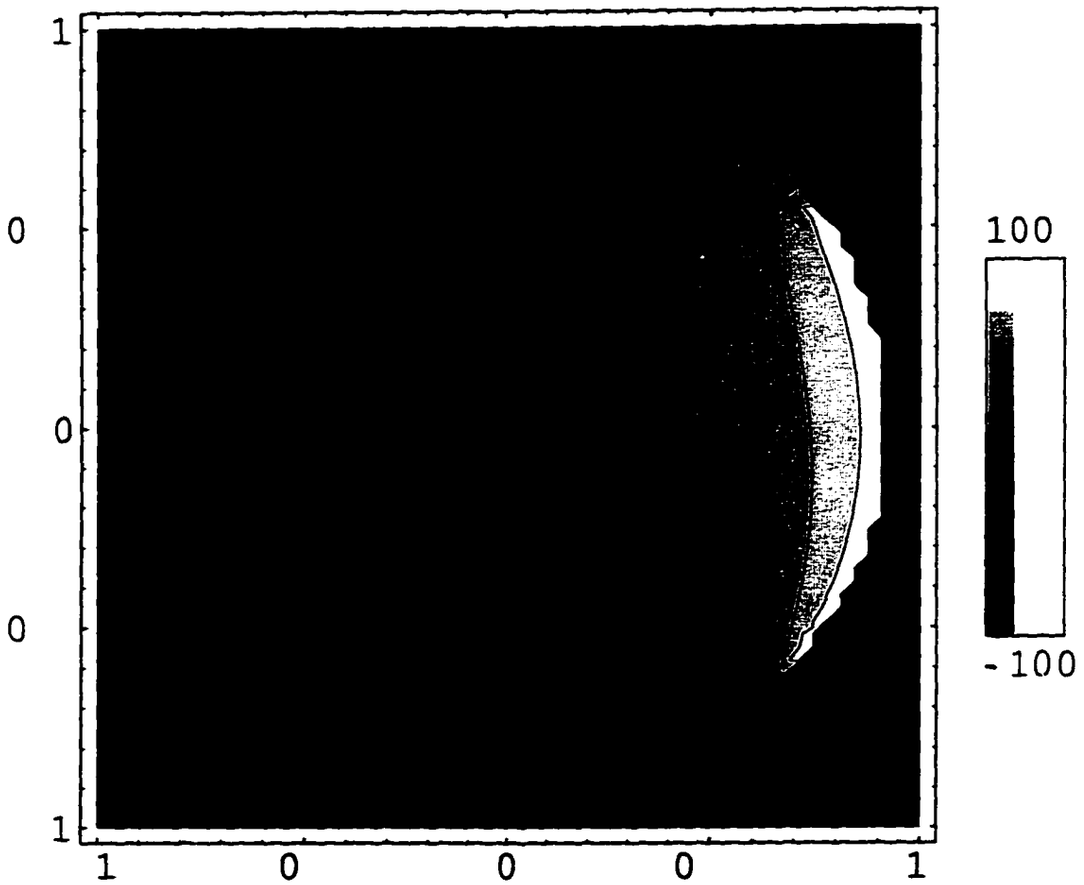

Figure 4.45: FMC solution of the octupole deflector of figure 4.44. The applied voltages were $V_x = 100$V, $V_y = 50V$ and $p = 0$, the grid spacing is $h = 0.025$ and there were 10^4 random walks started per point.

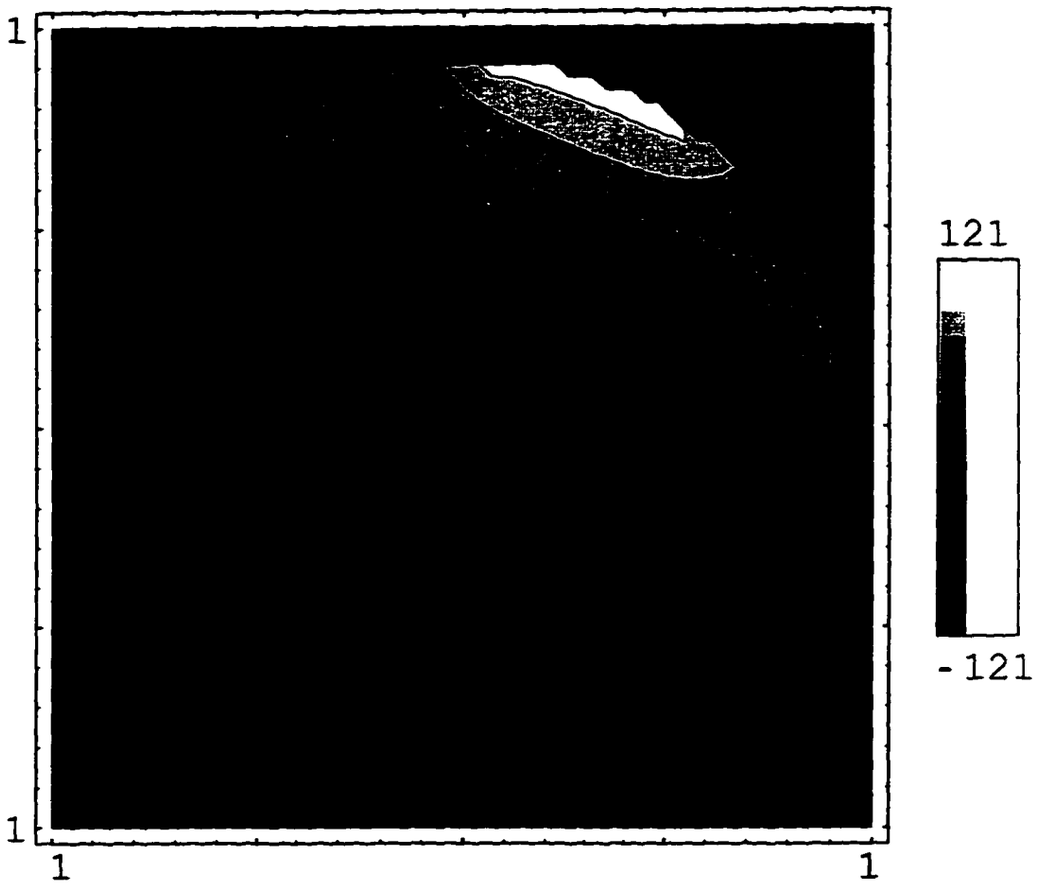

Figure 4.46: FMC solution of the octupole deflector of figure 4.44. The applied voltages were $V_x = 100$V, $V_y = 50V$ and $p = \sqrt{2} - 1$, the grid spacing is $h = 0.025$ and there were 10^4 random walks started per point.

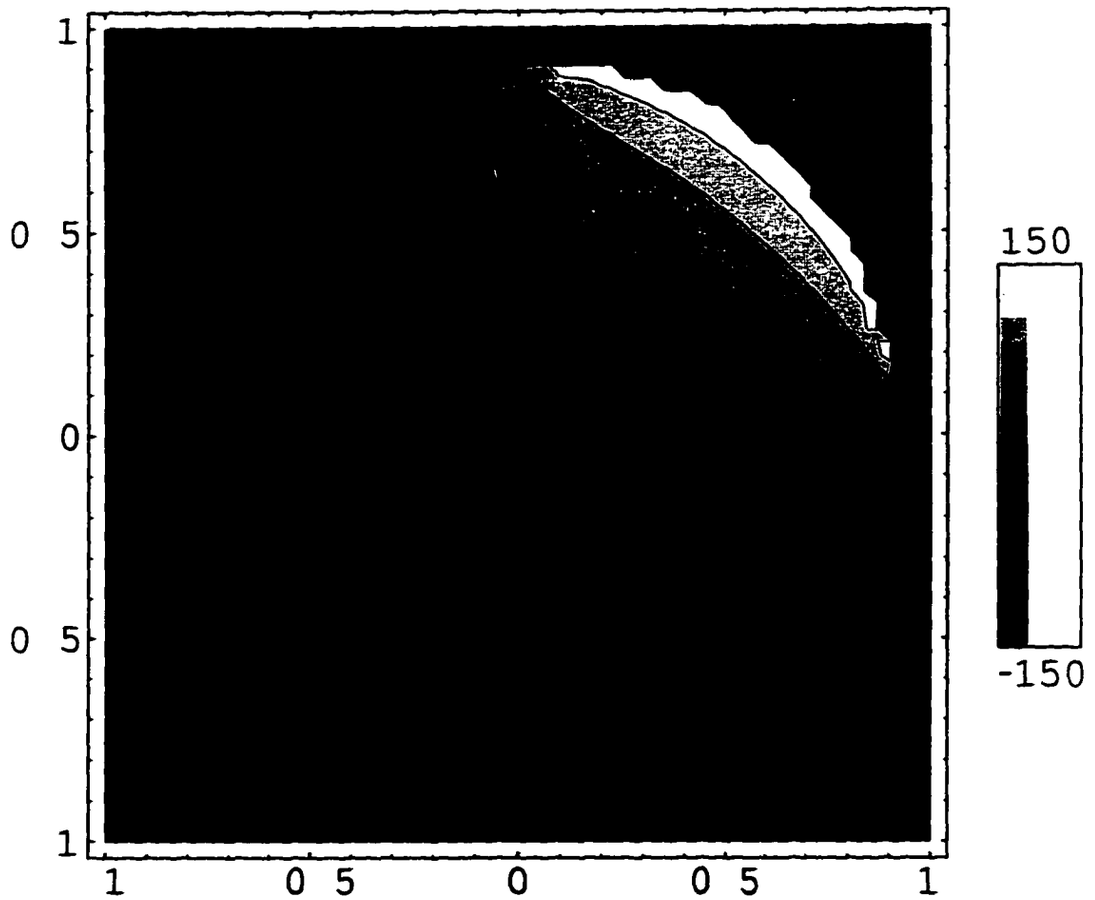

Figure 4.47: FMC solution of the octupole deflector of figure 4.44. The applied voltages were $V_x = 100$V, $V_y = 50V$ and $p = 1$, the grid spacing is $h = 0.025$ and there were 10^4 random walks started per point.

5 The General Linear Elliptic Problem

The functional form of the Laplacian operator in various coordinate systems is rarely as simple as in the case of rectangular systems. This involved functional representation forces us to consider more general cases of second-order operators, particularly those that correspond to the Laplacian, elliptic linear differential operators.

5.1 Preliminaries

It was pointed out in section 2.3 that the Laplacian operator in rectangular coordinates, which has formed the focus of our study so far, is the simplest case of a General Linear Elliptic (GLE) operator of order 2, $L[\cdot]$, which in two dimensions is of the form:

$$L[\phi] = \sum_{i,j=0}^{2} a_{ij} \frac{\partial^2 \phi}{\partial x_i \partial x_j}, \tag{2.10}$$

where the coefficients a_{ij} are functions of the coordinates $(x_1, x_2) = (x, y)$ and satisfy the *ellipticity condition*:

$$\mathbf{x}^T A \mathbf{x} \geq 0 \qquad \text{for} \qquad \mathbf{x} \neq 0, \tag{2.11}$$

with A being the matrix (a_{ij}) and $\mathbf{x}^T$ the vector $[x, y]$.

It is customary to write (2.10) as:

$$L[\phi] = a(x,y)\frac{\partial^2 \phi}{\partial x^2} + 2b(x,y)\frac{\partial^2 \phi}{\partial x \partial y} + c(x,y)\frac{\partial^2 \phi}{\partial y^2} + d(x,y)\frac{\partial \phi}{\partial x} + e(x,y)\frac{\partial \phi}{\partial y}, \tag{5.1}$$

where the coefficients a, b, c, d and e are functions of the position x, y. The ellipticity condition (2.11) can then be shown [112] to be equivalent to:

$$b^2 - ac < 0, \tag{5.2}$$

for all points (x, y) in the domain of the problem.

The corresponding general linear elliptic Dirichlet problem of order 2 is given by:

$$L[\phi(x,y)] = -f(x,y), \tag{2.12}$$

where, $\phi(x,y)$ is an unknown function defined in a domain Ω, satisfying Dirichlet boundary conditions on the boundary $\partial\Omega$ of the domain, and $f(x,y)$ is a known function in $\bar{\Omega} = \Omega \cup \partial\Omega$.

In the remainder of this chapter we will provide the theoretical framework for a Monte Carlo solution of (2.12) and present simulation results for sample problems.

5.2 Discretization of the Operator

Before we are able to solve the general elliptic problem via digital computer, we must first provide a discretized form of the elliptic operator. In the case of the Laplacian, in one dimension, the discretized form was particularly simple, equation (4.1):

$$\nabla^2\phi = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2}. \tag{4.1}$$

In two dimensions, assuming that we have a uniform grid of spacing h and that we are interested in the discrete approximation of L at point (i,j), the situation is significantly more involved: we must use the values of ϕ at a total of 9 points: point (i,j) itself, its 4 nearest neighbors $(i-1,j)$, $(i+1,j)$, $(i,j-1)$ and $(i,j+1)$ and the 4 diagonal points $(i-1,j-1)$, $(i+1,j-1)$, $(i+1,j+1)$ and $(i-1,j+1)$ thus resulting in a 9-point approximation which is accurate to an order of h^2 ($O(h^2)$).

Using the finite differences approximations for each of the second order partial derivatives in (5.1) [91]:

$$\frac{\partial^2\phi}{\partial x^2} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{h^2} + O(h^2), \tag{5.3}$$

$$\frac{\partial^2\phi}{\partial y^2} = \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{h^2} + O(h^2), \tag{5.4}$$

$$\frac{\partial^2\phi}{\partial x \partial y} = \frac{\phi_{i+1,j+1} - \phi_{i+1,j-1} - \phi_{i-1,j+1} + \phi_{i-1,j-1}}{4h^2} + O(h^2), \tag{5.5}$$

and likewise for the first order partial derivatives:

$$\frac{\partial\phi}{\partial x} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h} + O(h^2), \tag{5.6}$$

$$\frac{\partial\phi}{\partial y} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h} + O(h^2), \tag{5.7}$$

and substituting in (5.1), we obtain the discrete approximation of the general linear elliptic operator L at point i, j, accurate to $O(h^2)$:

$$L[\phi]_{i,j} \simeq \alpha_{i,j}\left[\sum_{m,n=-1}^{m,n=1}{}' \frac{\alpha_{i+m,j+n}}{\alpha_{i,j}}\phi_{i+m,j+n} - \phi_{i,j}\right], \tag{5.8}$$

where the prime in the summation indicates the fact that the term with $m = n = 0$ is not included. The various coefficients in (5.8) are:

$$\alpha_{i,j} = \frac{2}{h^2}\left(a_{i,j} + c_{i,j}\right), \tag{5.9}$$

$$\alpha_{i-1,j} = \frac{a_{i,j}}{h^2} - \frac{d_{i,j}}{2h}, \tag{5.10}$$

$$\alpha_{i,j-1} = \frac{c_{i,j}}{h^2} - \frac{e_{i,j}}{2h}, \tag{5.11}$$

$$\alpha_{i+1,j} = \frac{a_{i,j}}{h^2} + \frac{d_{i,j}}{2h}, \tag{5.12}$$

$$\alpha_{i,j+1} = \frac{c_{i,j}}{h^2} + \frac{e_{i,j}}{2h}, \tag{5.13}$$

and,

$$\alpha_{i+1,j+1} = \alpha_{i-1,j-1} = -\alpha_{i+1,j-1} = -\alpha_{i-1,j+1} = \frac{b_{i,j}}{2h^2}. \tag{5.14}$$

For reasons that will become apparent shortly, we wish to associate the previously defined

ratios $\alpha_{i+m,j+n}/\alpha_{i,j}$ appearing in (5.8) with probabilities. Immediately it is seen from (5.14) that regardless of the sign of $b(x,y)$, at least two of these terms will be negative. It is, however, possible [104] to eliminate the cross-term $\partial^2\phi/\partial x\partial y$ from (5.1) via the change of variables:

$$x' = \eta - \zeta \qquad \text{and} \qquad y' = \eta + \zeta, \tag{5.15}$$

where the auxiliary variables η and ζ satisfy the following partial differential equations:

$$\frac{\partial\eta/\partial x}{\partial\eta/\partial y} = \frac{-b - \sqrt{b^2 - ac}}{a}, \tag{5.16}$$

$$\frac{\partial\zeta/\partial x}{\partial\zeta/\partial y} = \frac{-b + \sqrt{b^2 - ac}}{a}. \tag{5.17}$$

Garabedian [104] shows that these equations are always solvable in principle. We will assume therefore in the following discussion that $b = 0$. In this case, (5.8) becomes:

$$L[\phi]_{i,j} \simeq \alpha_{i,j}\left[\lambda_{i-1,j}\phi_{i-1,j} + \lambda_{i+1,j}\phi_{i+1,j} + \lambda_{i,j-1}\phi_{i,j-1} + \lambda_{i,j+1}\phi_{i,j+1} - \phi_{i,j}\right], \tag{5.18}$$

with the λ's given by:

$$\lambda_{i-1,j} = \frac{2a_{i,j} - d_{i,j}h}{4(a_{i,j} + c_{i,j})}, \tag{5.19}$$

$$\lambda_{i,j-1} = \frac{2c_{i,j} - e_{i,j}h}{4(a_{i,j} + c_{i,j})}, \tag{5.20}$$

$$\lambda_{i+1,j} = \frac{2a_{i,j} + d_{i,j}h}{4(a_{i,j} + c_{i,j})}, \tag{5.21}$$

$$\lambda_{i,j+1} = \frac{2c_{i,j} + e_{i,j}h}{4(a_{i,j} + c_{i,j})}. \tag{5.22}$$

Furthermore, because of the ellipticity condition $-ac < 0$, a and c have always the same sign and we can always manipulate (5.1) so that $a, c > 0$. It can be shown that, under these conditions [112], values of h exist for which all the λ's are positive and < 1. Simple

algebraic manipulation also shows that the sum of all the λ's is 1. These facts, coupled with the already known probabilistic interpretation of the discretization coefficients for the Laplacian, point towards an assignment of probability values to the $\lambda's$. In appendix B we provide full theoretical justification of this assignment. The exact interpretation of these probability values will be given shortly.

We are now ready to present the extension to the main theorem, with the help of which we will perform Monte Carlo simulations for the solution of the general elliptic problem.

Extended Main Theorem. *The solution to the interior Dirichlet problem* $L[\phi(\mathbf{x})] = -f(\mathbf{x})$ *for* $\mathbf{x} \in \Omega$, *with* $\phi(\mathbf{x}) = g(\mathbf{x})$ *for* $\mathbf{x} \in \partial\Omega$, *is given by:*

$$\phi(\mathbf{x}) = \left(\frac{1}{\alpha_0(\mathbf{x})}\right) \mathbf{E}_{\mathbf{x}} \left[\sum_{n=0}^{\tau-1} f(\mathbf{x}_n)\right] + \mathbf{E}_{\mathbf{x}} \left[g(\mathbf{x}_\tau)\right]. \tag{5.23}$$

where $\mathbf{E}_{\mathbf{x}} \left[g(\mathbf{x}_\tau)\right]$ *is the expectation value for the function* g *of a random walk that starts at* $\mathbf{x}$ *and terminates when it reaches the boundary* $\partial\Omega$ *at time* τ, *and* $\mathbf{x}_n$ *is the position of the random walk at the* n*-th step.*

We provide a proof of the Main Theorem in appendix B along with some important extensions. To the best of our knowledge, this proof and extensions are given for the first time. In practical terms, this theorem provides us with a simple algorithm to determine the unknown function ϕ at any given point $\mathbf{x} \in \Omega$:

- Start a random walk from $\mathbf{x}$. This random walk is non-simple: the probability that, starting from point $\mathbf{x}$, the walk moves to point $\mathbf{x}'$, where $\mathbf{x}'$ is any of the 4 neighbors of $\mathbf{x}$, is no longer equal to 1/4, but is given instead by the values of the λ's in (5.19)-(5.22).

- When the random walk reaches the boundary $\partial\Omega$ at point $\mathbf{q}$, it terminates and a new random walk is started from $\mathbf{x}$. Keep track of the value $g(\mathbf{q})$ and the values of $f(\mathbf{x})$ during the entire walk.
- After a large number of random walks, the value of ϕ at $\mathbf{x}$ is just the average of $g(\mathbf{p})$, with a correction term whose magnitude depends on the source f.

5.3 Applications

5.3.1 A Linear Elliptic Problem

As an application of the preceding theory, we consider the following linear elliptic partial differential equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - 2\frac{\partial u}{\partial x} = 0, \tag{5.24}$$

with the solution $u(x, y)$ desired on the unit square $[0, 1] \times [0, 1]$, subject to the boundary conditions $u(0, y) = \sin(\pi y)$ and $u = 0$ on the remaining sides of the square. It is also assumed that a uniform grid of spacing h has been defined.

Equation (5.24) has a closed form solution [113]:

$$u(x, y) = e^x \frac{\sinh \sigma(1 - x)}{\sinh \sigma} \sin(\pi y), \tag{5.25}$$

where $\sigma = \sqrt{1 + \pi^2}$.

Comparing (5.24) with the canonical form, (5.1), we have $a = 1$, $b = 0$, $c = 1$, $d = -2$, $e = 0$ and $f = 0$. From (5.9), we have:

$$\alpha_{i,j} = \frac{4}{h^2}, \tag{5.26}$$

while the transition probabilities for every step of the random walk, i.e. the λ's, are given

by (5.19)-(5.22):

$$\lambda_{i-1,j} = \frac{1+h}{4}, \tag{5.27}$$

$$\lambda_{i,j-1} = \frac{1}{4}, \tag{5.28}$$

$$\lambda_{i+1,j} = \frac{1-h}{4}, \tag{5.29}$$

$$\lambda_{i,j+1} = \frac{1}{4}. \tag{5.30}$$

These equations reveal the destruction of the symmetry of the random walk due to the presence of the $\partial u/\partial x$ term in (5.24). It is interesting to note that only the propagation of the random walk along the x direction has been affected, with the probabilities of propagation along the y direction remaining constant at $1/4$, their value unaffected by the first-derivative term. Furthermore, (5.29) imposes the restriction $h < 2$, which is not a concern here since the domain is only 1 unit long and wide. It could be an issue, however, if the domain were extremely large. In that case, proper normalization of the coordinates would be required. Typically, for a 20×20 mesh, $h = 0.05$ and the effect of the extra derivative term on the propagation along the x direction is very small: 0.2438 vs. 0.2562 for steps to the right and left, respectively.

The exact solution of (5.24), equation (5.25) is plotted in figure 5.1.

We implemented a Full Monte Carlo solution of (5.24) via the algorithm presented in the extended main theorem. The results, for various numbers of random walks, from $N_W = 10$ to $N_W = 10^4$ are shown in the sequence of figures 5.2-5.5. The grid on which the simulations were performed was 21×21 which corresponds to $h = 0.05$.

To obtain a better view of the convergence of the method, a comparison of contour plots of the true and FMC solution are plotted in figures 5.6-5.9. The characteristic fast initial convergence of FMC is apparent in this sequence of plots.

We also plot the average number of steps to reach any boundary from point (x, y). As proven in Appendix B, this number, $U(x, y)$, is the solution to the boundary value

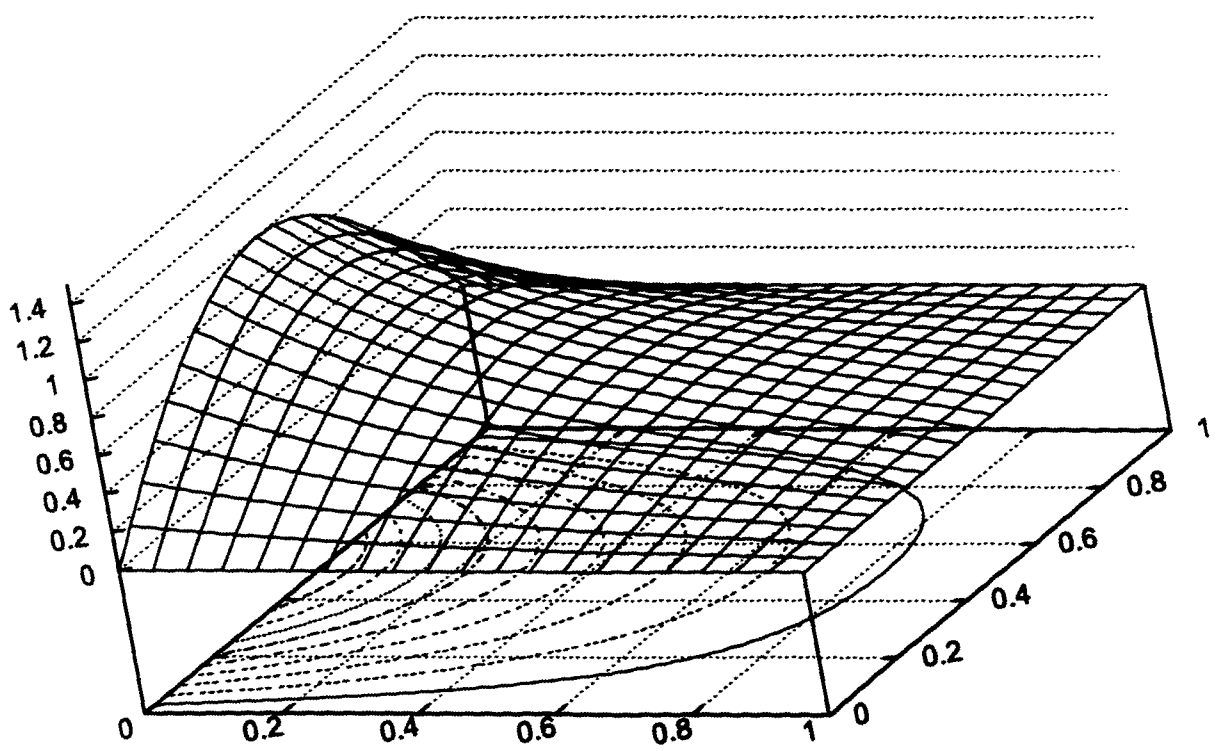

Figure 5.1: Exact solution to (5.24). Also shown are the contours at $u = 0.1 \ldots 0.9$.

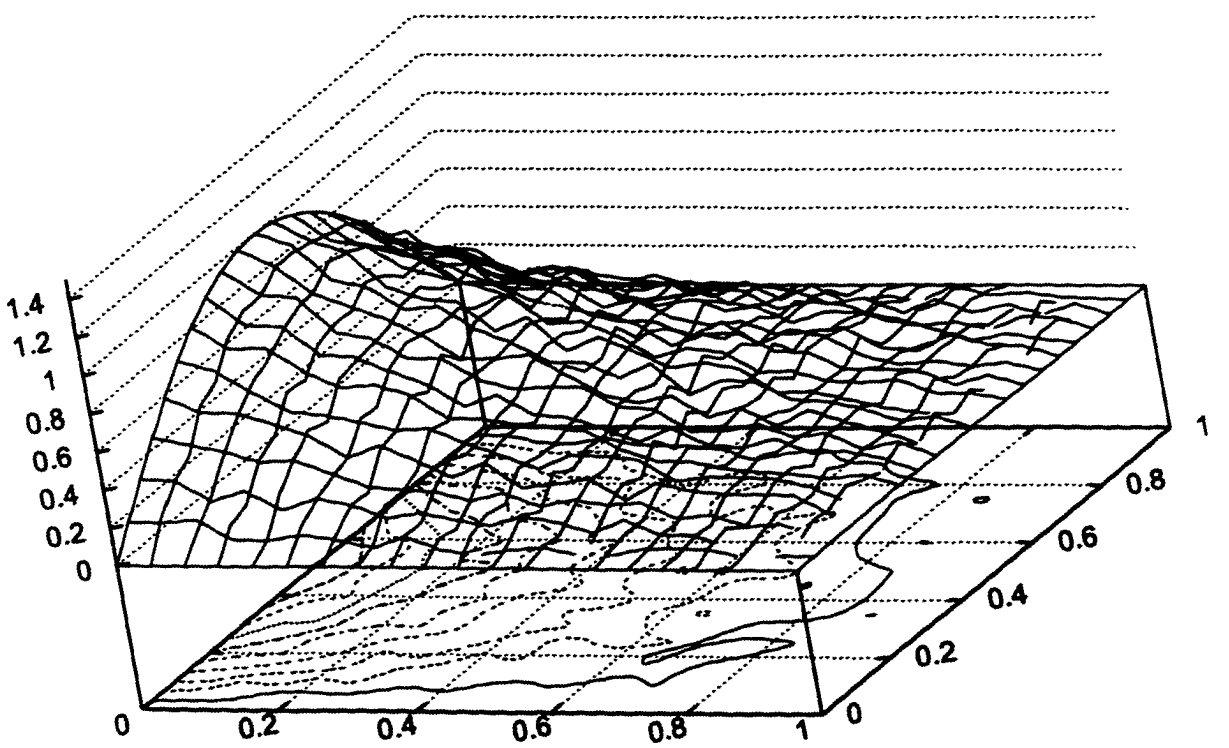

Figure 5.2: FMC solution to (5.24). There were $N_W = 10$ random walks started per point, and the underlying grid was 21×21. The projections on the bottom of the figure are constant value contours.

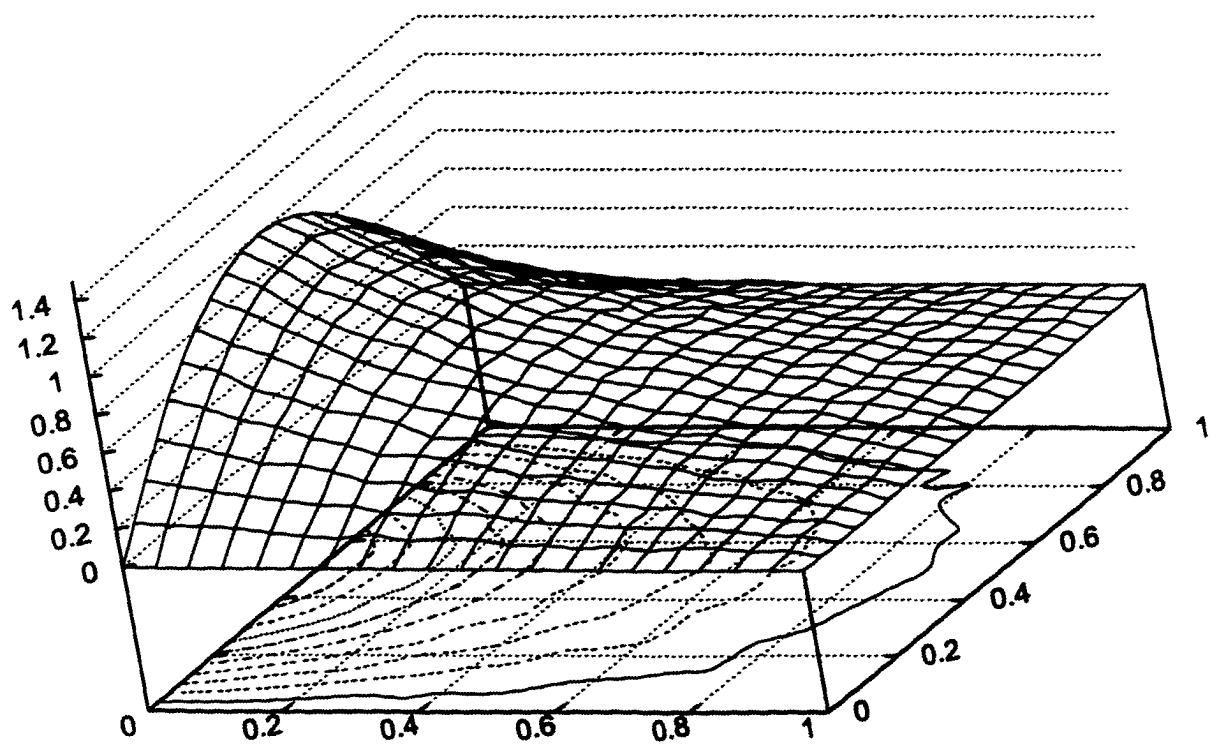

Figure 5.3: FMC solution to (5.24). There were $N_W = 100$ random walks started per point, and the underlying grid was 21×21. The projections on the bottom of the figure are constant value contours.

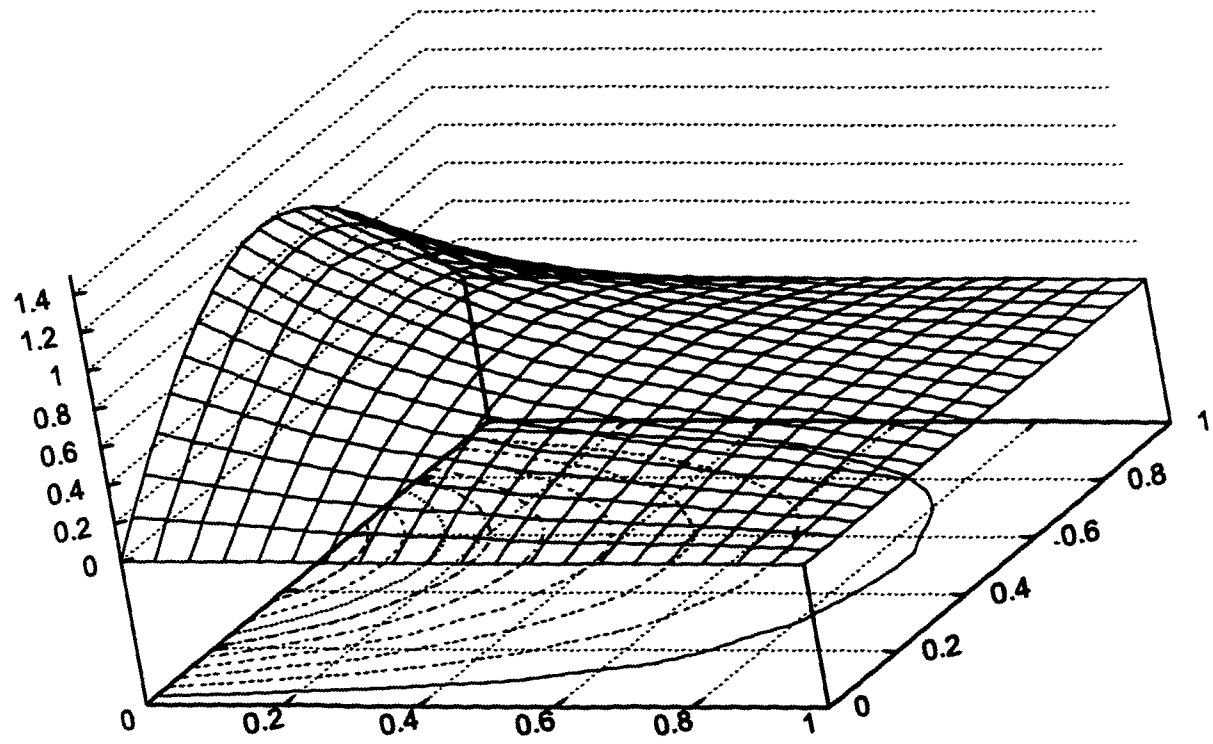

Figure 5.4: FMC solution to (5.24). There were $N_W = 10^3$ random walks started per point, and the underlying grid was 21×21. The projections on the bottom of the figure are constant value contours.

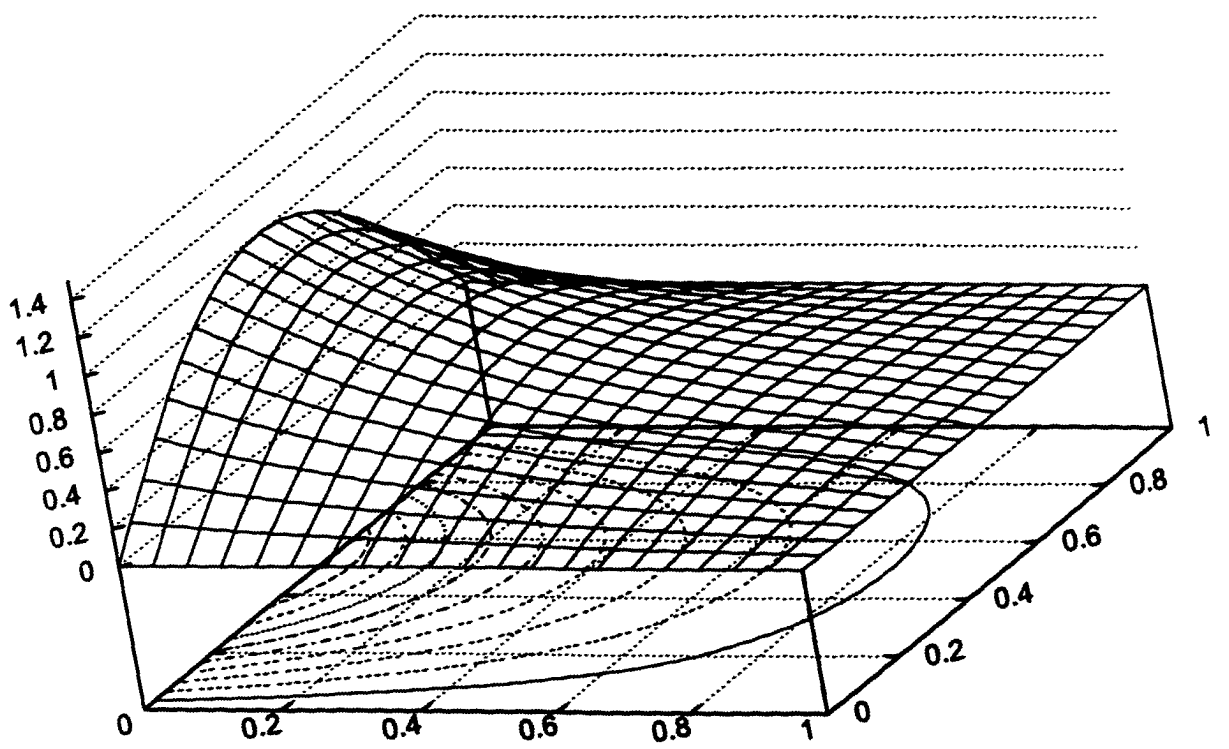

Figure 5.5: FMC solution to (5.24). There were $N_W = 10^4$ random walks started per point, and the underlying grid was 21×21. The projections on the bottom of the figure are constant value contours.

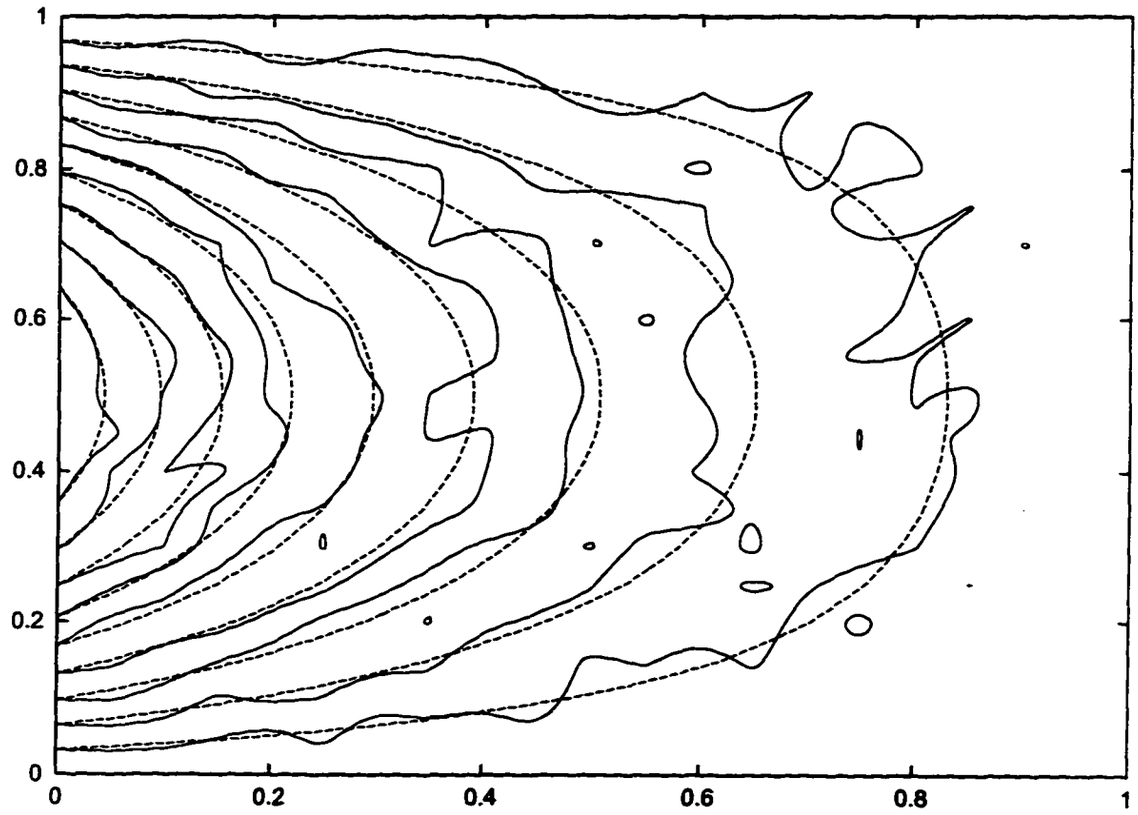

Figure 5.6: FMC solution to (5.24). There were $N_W = 10$ random walks started per point, and the underlying grid was 21×21. Contours of the true (dashed line) vs. calculated (solid line) solution are shown.

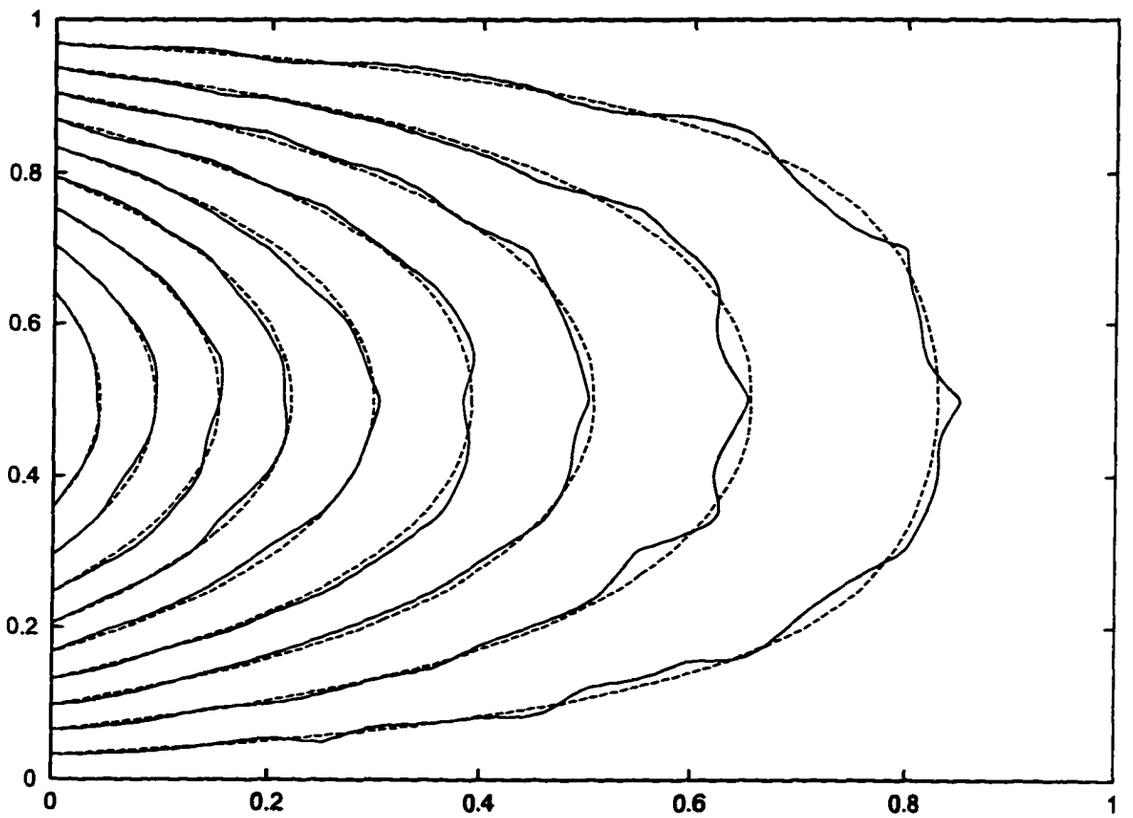

Figure 5.7: FMC solution to (5.24). There were $N_W = 100$ random walks started per point, and the underlying grid was 21×21. Contours of the true (dashed line) vs. calculated (solid line) solution are shown.

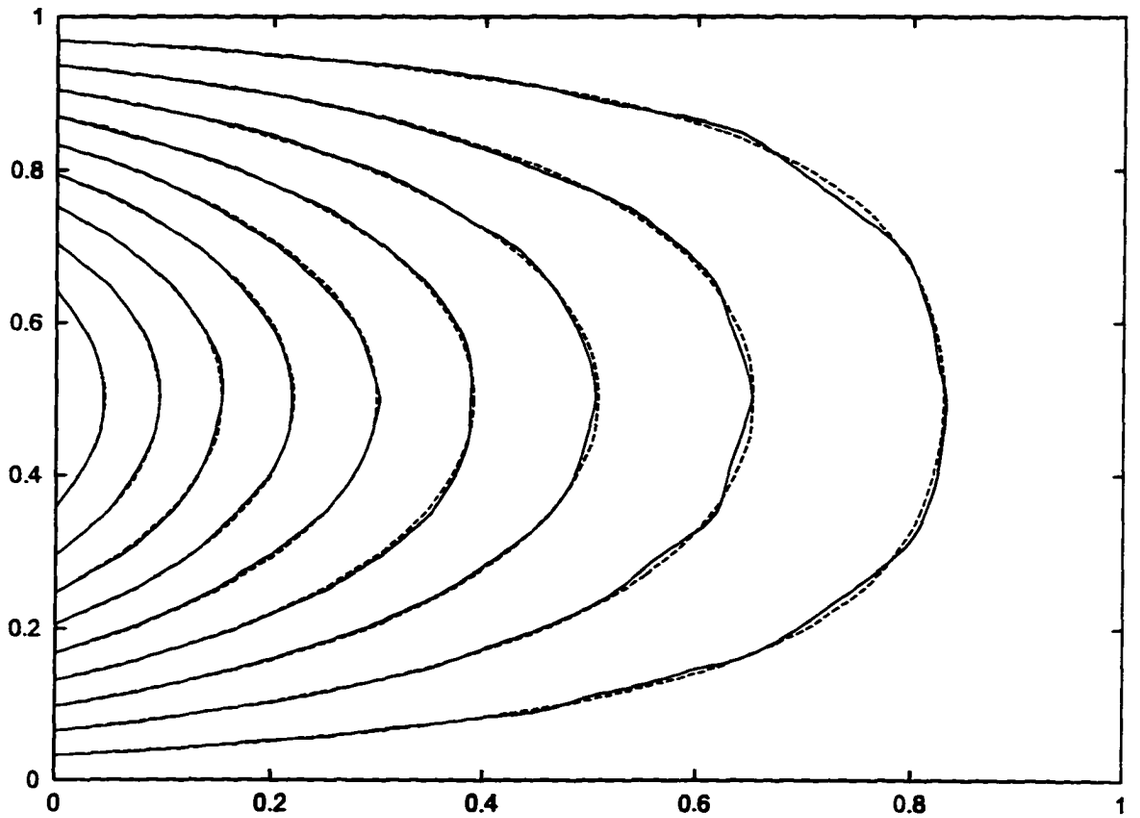

Figure 5.8: FMC solution to (5.24). There were $N_W = 10^3$ random walks started per point, and the underlying grid was 21×21. Contours of the true (dashed line) vs. calculated (solid line) solution are shown.

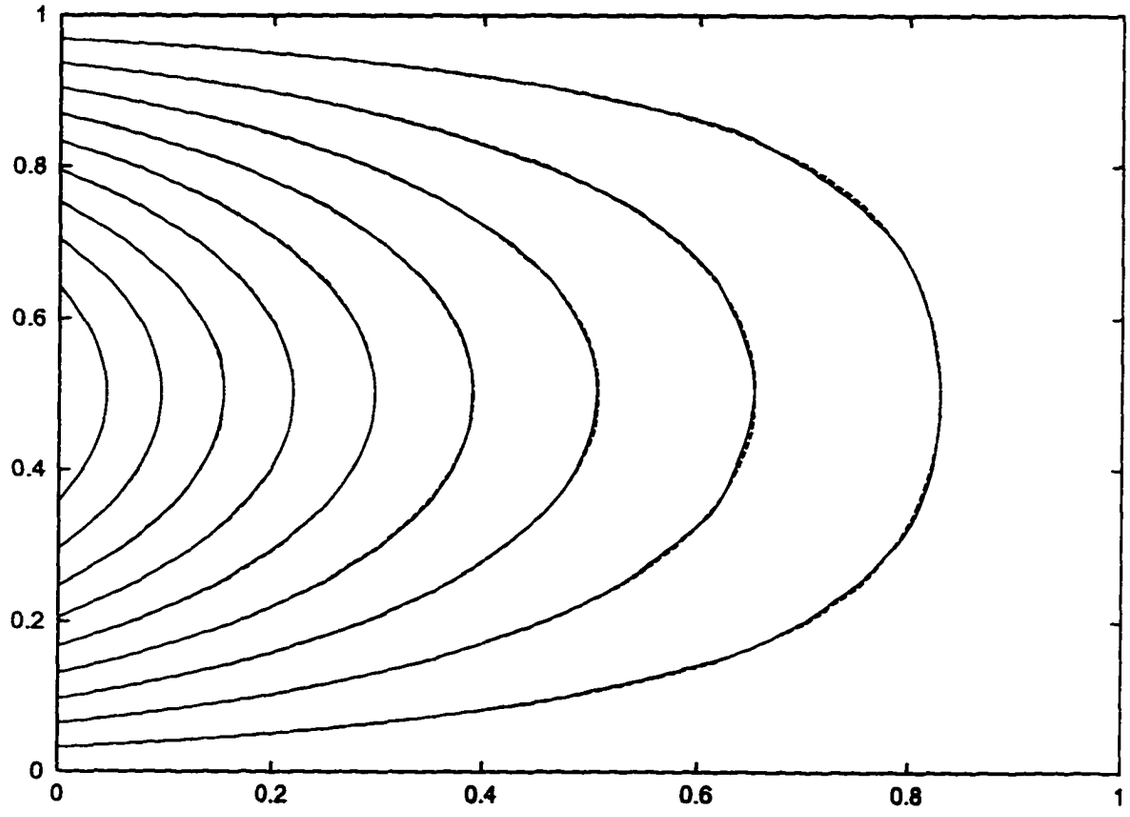

Figure 5.9: FMC solution to (5.24). There were $N_W = 10^4$ random walks started per point, and the underlying grid was 21×21. Contours of the true (dashed line) vs. calculated (solid line) solution are shown.

problem:

$$\nabla^2 U - 2\frac{\partial U}{\partial x} = -\frac{4}{h^2}, \tag{5.31}$$

subject to the condition $U = 0$ on the unit square. The numerical results, for $N_W = 10^4$ although this is independent of N_W, are shown in figure 5.10.

The same information is conveyed in figure 5.11 in the form of a contour plot. In this instance, however, the plot clearly indicates that the maximum number of steps occurs not in the middle of the square, as it would for the Laplacian operator, but slightly offset to the right of the middle point. This is expected, since as we saw earlier, the walk is slightly more likely to take a left step than a right step.

Finally, as a demonstration of the power of the FMC method, we solved the same problem as in (5.24) but with different boundary conditions: the sine wave along the left boundary is "rotated" along the length of the square so that $u(0, y) = \sin(\pi(y - 0.5))$ for $0.5 < y < 1$, $u(1, x) = \cos(\pi x)$ for $0 < x < 0.5$ and $u = 0$ elsewhere on the unit square. The behavior of the solution for this problem is quite different than the solution presented earlier. With any other method, it is necessary to reformulate and solve the entire problem from the beginning. FMC, however, allows the use of statistics obtained during the solution of the original problem to be used for the problem with modified boundary conditions. What is of importance is the geometry of the problem alone. While the initial solution at an accuracy level corresponding to 10^4 random walks per point required a computational time on a personal computer of 731 secs., the additional time for the solution with different boundary conditions was less than 1 sec. (the time measurement precision). Thus, once the FMC method has been used to evaluate the geometrical factors of the problem, the effect of individual boundary conditions can be evaluated in negligible time. The solution is shown in figure (5.12).

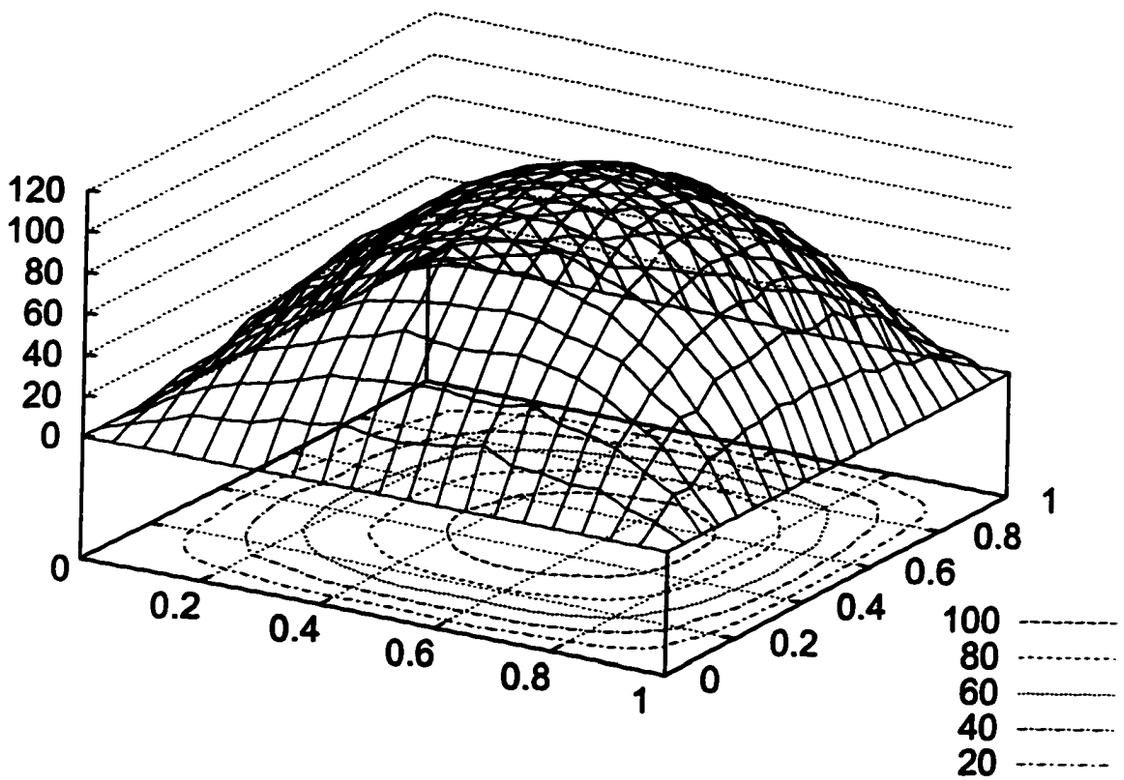

Figure 5.10: Average number of steps to reach a boundary during the FMC solution to (5.24). There were $N_W = 10^4$ random walks started per point, and the underlying grid was 21×21. The projections on the bottom of the figure are constant value contours.

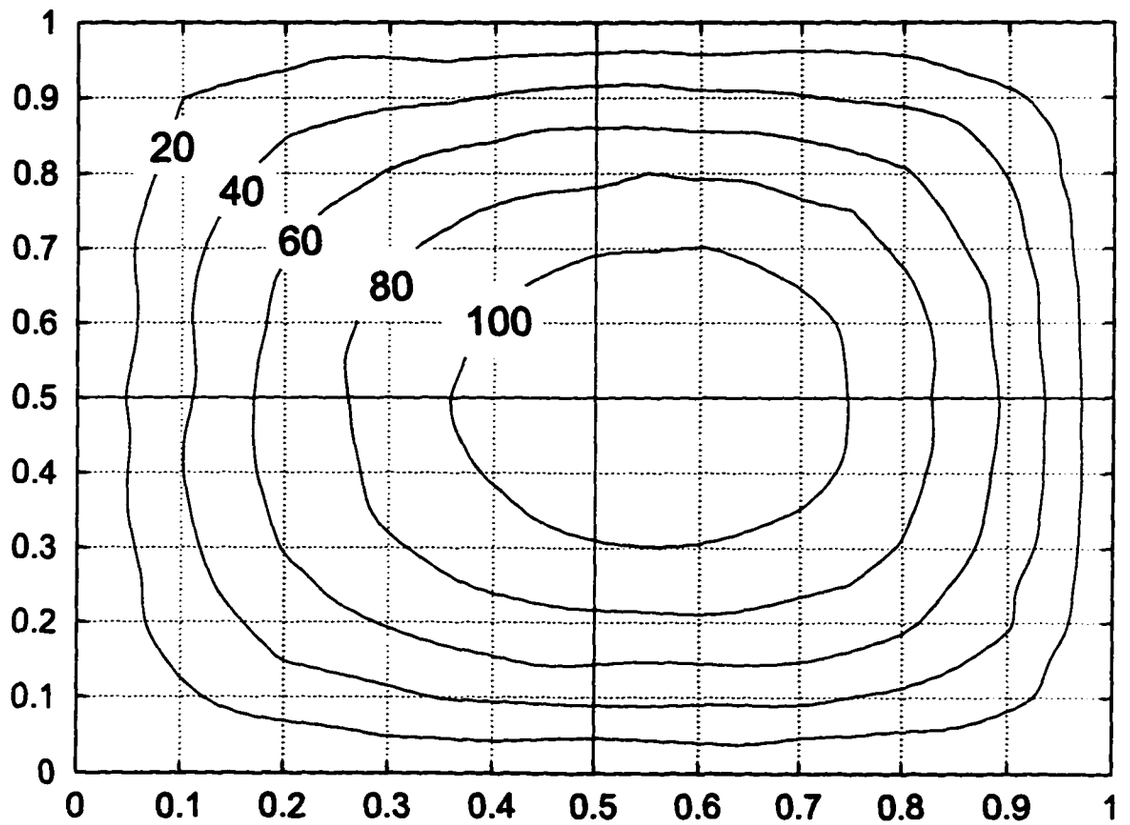

Figure 5.11: Average number of steps to reach a boundary during the FMC solution to (5.24), shown as a contour plot. There were $N_W = 10^4$ random walks started per point, and the underlying grid was 21×21.

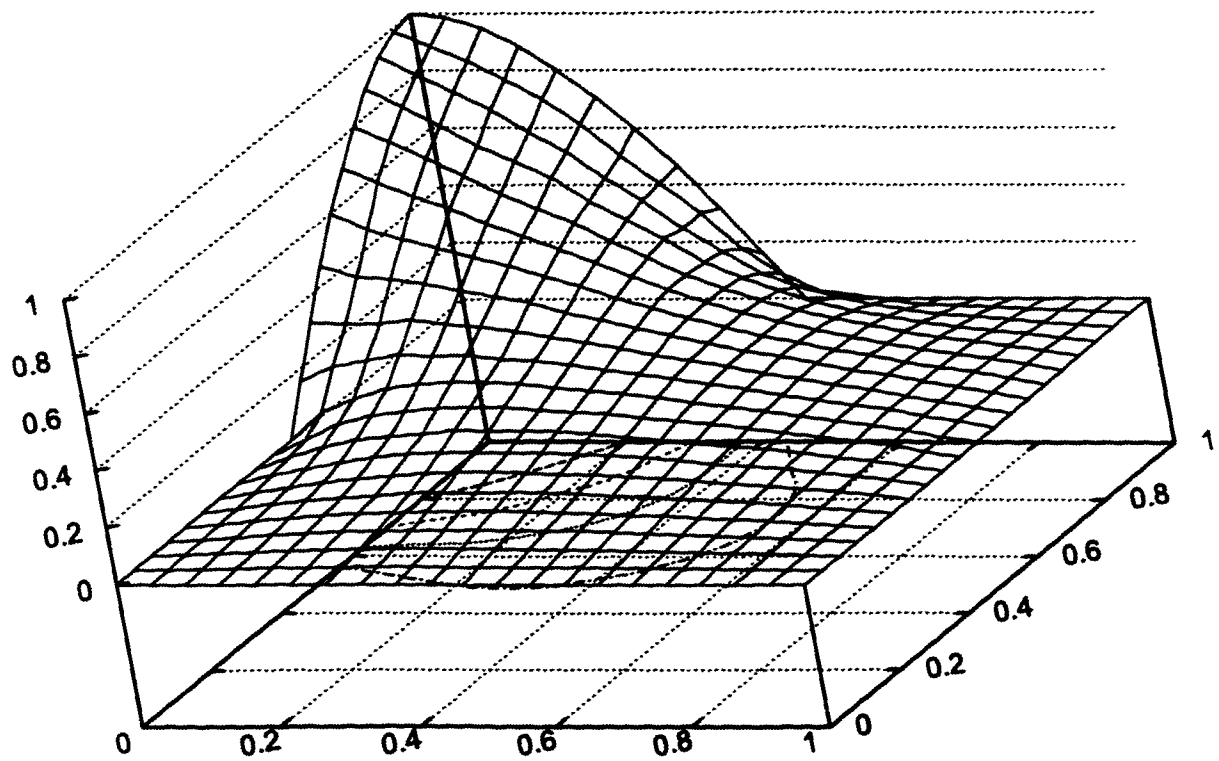

Figure 5.12: FMC solution to a modification of (5.24). There were $N_W = 10^4$ random walks started per point, and the underlying grid was 21×21. The projections on the bottom of the figure are constant value contours. The additional time to compute this solution was less than 1 second (see text).

5.3.2 Electrostatic Lenses

As a further application of the Monte Carlo method, we present simulations of two electrostatic focusing systems (lenses) which have been presented in the literature and which have actually been used in a recoil-mass time-of-flight spectrometer [2].

The geometry of the first lens is shown in figure 5.13. Because of the axisymmetric nature of the problem, cylindrical coordinates, (ρ, z), are appropriate for the expression of the Laplacian. In this coordinate system, however, the Laplacian forms a general linear elliptic operator and Laplace's equation becomes:

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial\rho^2} + \frac{1}{\rho}\frac{\partial\phi}{\partial\rho} + \frac{\partial^2\phi}{\partial z^2} = 0. \tag{5.32}$$

This expression is similar to the canonical form (5.1) with x and y replaced by z and ρ respectively, and $a = 1$, $b = 0$, $c = 1$, $d = 0$, $e = 1/\rho$ and $f = 0$. From (5.9), we have:

$$\alpha_{i,j} = \frac{4}{h^2}, \tag{5.33}$$

while the transition probabilities for every step of the random walk, i.e. the λ's, are given by (5.19)-(5.22):

$$\lambda_{i-1,j} = \frac{1}{4}, \tag{5.34}$$

$$\lambda_{i,j-1} = \frac{2 - h/\rho}{8}, \tag{5.35}$$

$$\lambda_{i+1,j} = \frac{1}{4}, \tag{5.36}$$

$$\lambda_{i,j+1} = \frac{2 + h/\rho}{8}. \tag{5.37}$$

The original Laplace's equation (5.32) has an apparently singular term:

$$\frac{1}{\rho}\frac{\partial\phi}{\partial\rho}, \tag{5.38}$$

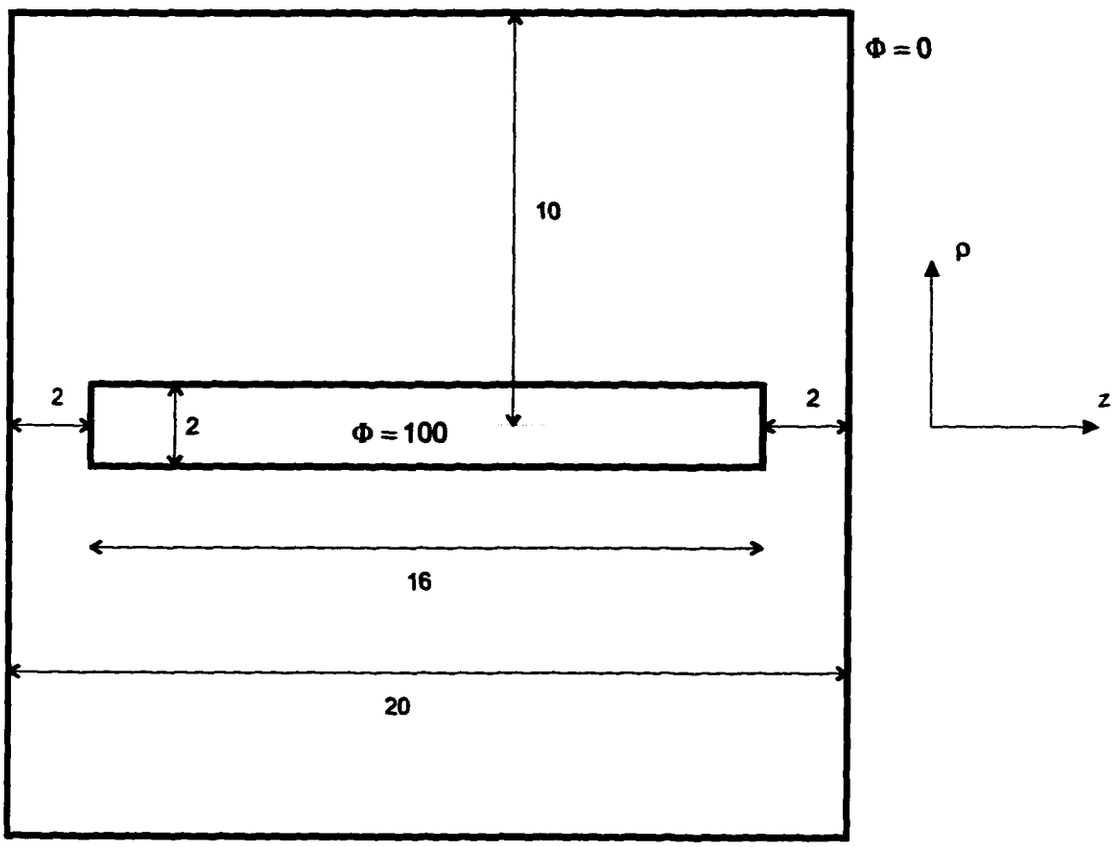

Figure 5.13: Electrostatic lens # 1 simulated in this work. A cross-section along the $z - \rho$ plane is shown. The lens is axially symmetric with respect to the z-axis and is in the form of two concentric cylinders. The inner cylinder is maintained at 100V while the outer cylinder is grounded. After [2].

along the z-axis ($\rho = 0$) and this singularity also appears in the transition probabilities of the random walk, (5.35) and (5.37). The case $\rho = 0$ requires therefore special treatment. Because of the axisymmetry of the problem,

$$\left.\frac{\partial \phi}{\partial \rho}\right|_{\rho=0} = 0, \tag{5.39}$$

and simple application of L'Hôspital's rule yields:

$$\lim_{\rho \to 0} \frac{1}{\rho} \frac{\partial \phi}{\partial \rho} = \left.\frac{\partial^2 \phi}{\partial \rho^2}\right|_{\rho=0}. \tag{5.40}$$

Hence, along the z-axis Laplace's equation (5.32) reduces to:

$$2\frac{\partial^2 \phi}{\partial \rho^2} + \frac{\partial^2 \phi}{\partial z^2} = 0. \tag{5.41}$$

Comparison with the canonical form (5.1) indicates that along the z-axis *only*, $a = 2$, $b = 0, c = 1, d = e = f = 0$ and:

$$\alpha_{i,j} = \frac{6}{h^2}, \tag{5.42}$$

$$\lambda_{i-1,j} = \frac{1}{6}, \tag{5.43}$$

$$\lambda_{i,j-1} = \frac{1}{3}, \tag{5.44}$$

$$\lambda_{i+1,j} = \frac{1}{6}, \tag{5.45}$$

$$\lambda_{i,j+1} = \frac{1}{3}. \tag{5.46}$$

Knowledge of the transition probabilities allows us to perform Monte Carlo simulations on the lens described previously. We used a 41×41 mesh, resulting in a spacing $h = 0.50$ and implemented the Monte-Carlo-Relaxation method. Initially we performed $N_W = 50$ random walks, at a CPU time of only 13 secs. The results are shown in figure 5.14. Subsequently we performed relaxation using as starting value the potential obtained from

the Monte Carlo simulations. It required only 15 iterations and an additional CPU time of 59 secs. to obtain an accuracy better than 0.1%. Since the exact solution is not known, the measure of convergence was the change in potential between successive relaxations. The result is shown in figures 5.15 and 5.16.

Finally, we show in figures 5.17 and 5.18 the average number of steps required to reach a boundary.

Following the same procedure, we simulated using MCR the lens shown in figure 5.19 which has also been presented in [2].

The MCR solution, following $N_W = 50$ random walks and 15 relaxations, is shown in figures 5.20 and 5.21. To maintain the grid uniformity, a mesh of 41×81 was used, resulting in a spacing $h = 0.50$. The MC part of the solution required 44 secs. and the relaxation part required 108 secs. for a total of 152 secs. of CPU time.

Also, we show in figures 5.22 and 5.23 the average number of steps required to reach a boundary.

Our results match the published relaxation results by DiStasio *et al.* [2]

5.4 Summary

We demonstrated the extensions of the simple theory presented in chapters 3 and 4 to problems of considerable complexity, namely general linear elliptic operators of the second order in arbitrary domains. The intimate connection between random walks (or, more generally, Brownian motion) and the elliptic operator is clearly demonstrated when the operator is discretized in the domain of the problem. This connection allowed us to extend the main theorem of chapter 2 to the general elliptic problem and provided us with probabilistic algorithms for its solution. Since a general form of the Laplacian appears in non-rectangular geometries, we turned our attention first to rectangular domains where the operator defined is of a more complex nature than the simple Laplacian, and subsequently, to domains that possess axisymmetry and which correspond to geometries encountered

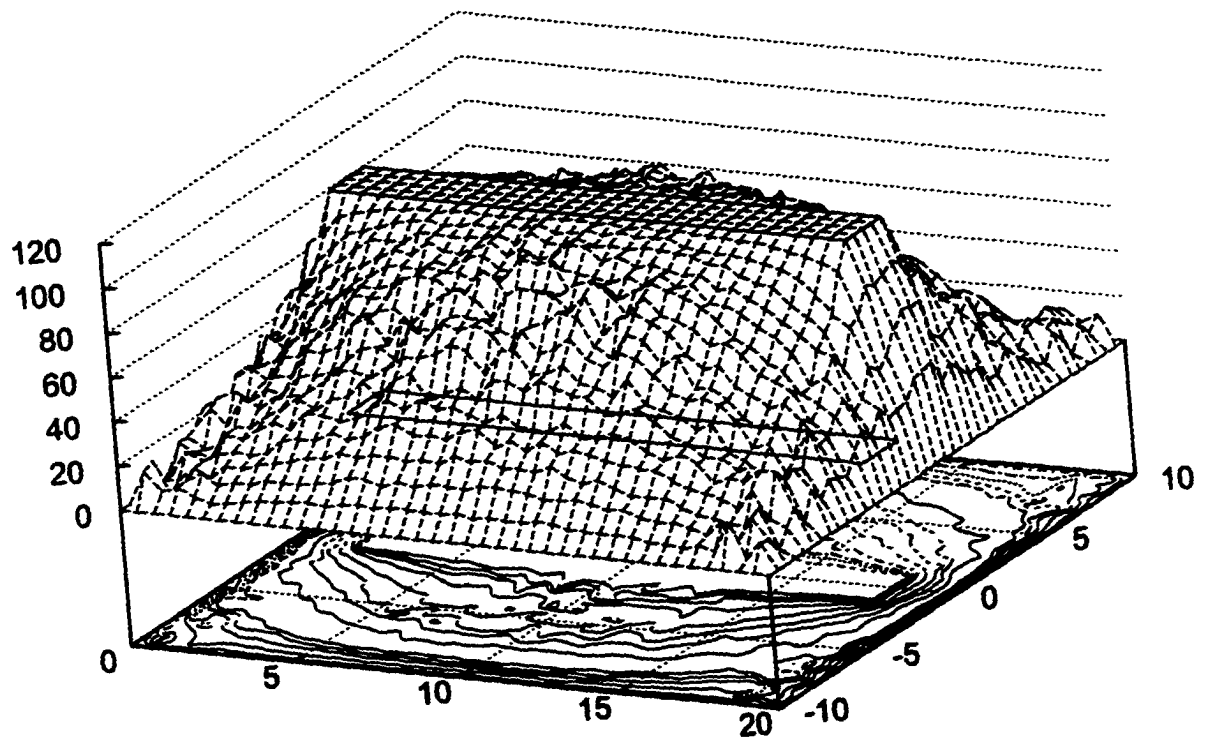

Figure 5.14: Monte Carlo solution to the electrostatic lens of figure 5.13. There were $N_W = 50$ random walks started per point, and the underlying grid was 41×41. The projections on the bottom of the figure are equipotential contours. Total CPU time was 13 secs.

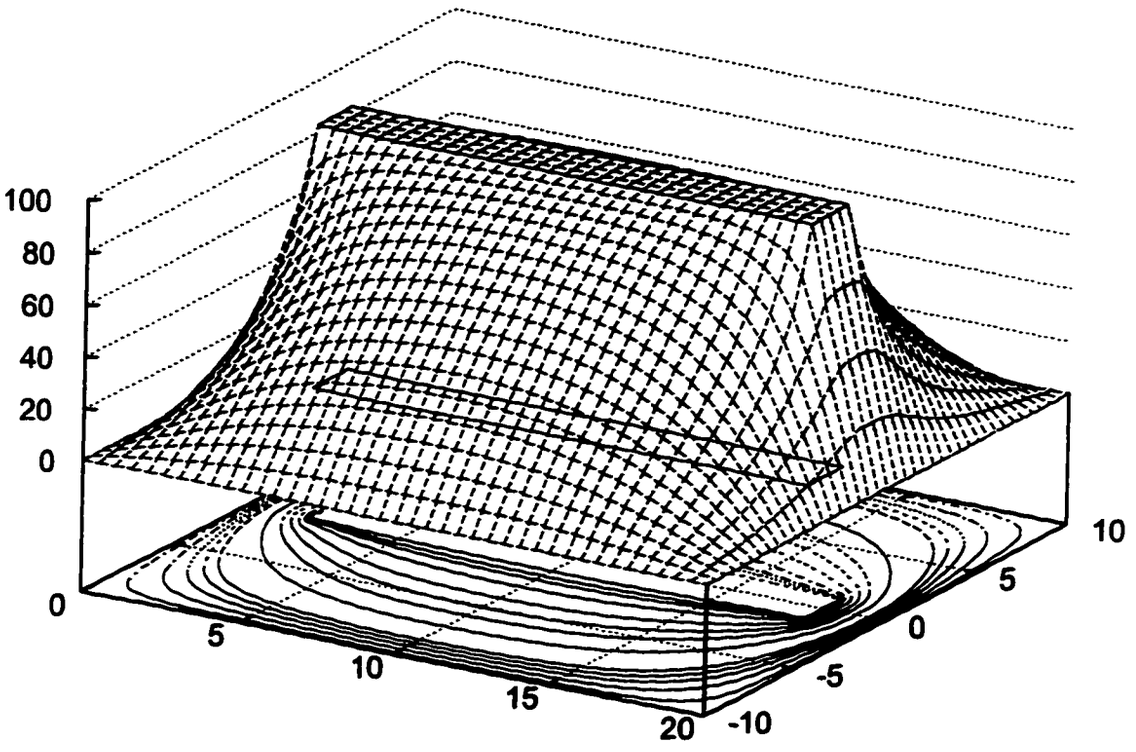

Figure 5.15: MCR solution to the electrostatic lens of figure 5.13. There were 15 relaxation iterations with starting value the potential calculated in figure 5.14. The projections on the bottom of the figure are equipotential contours. Total CPU time was 72 secs.

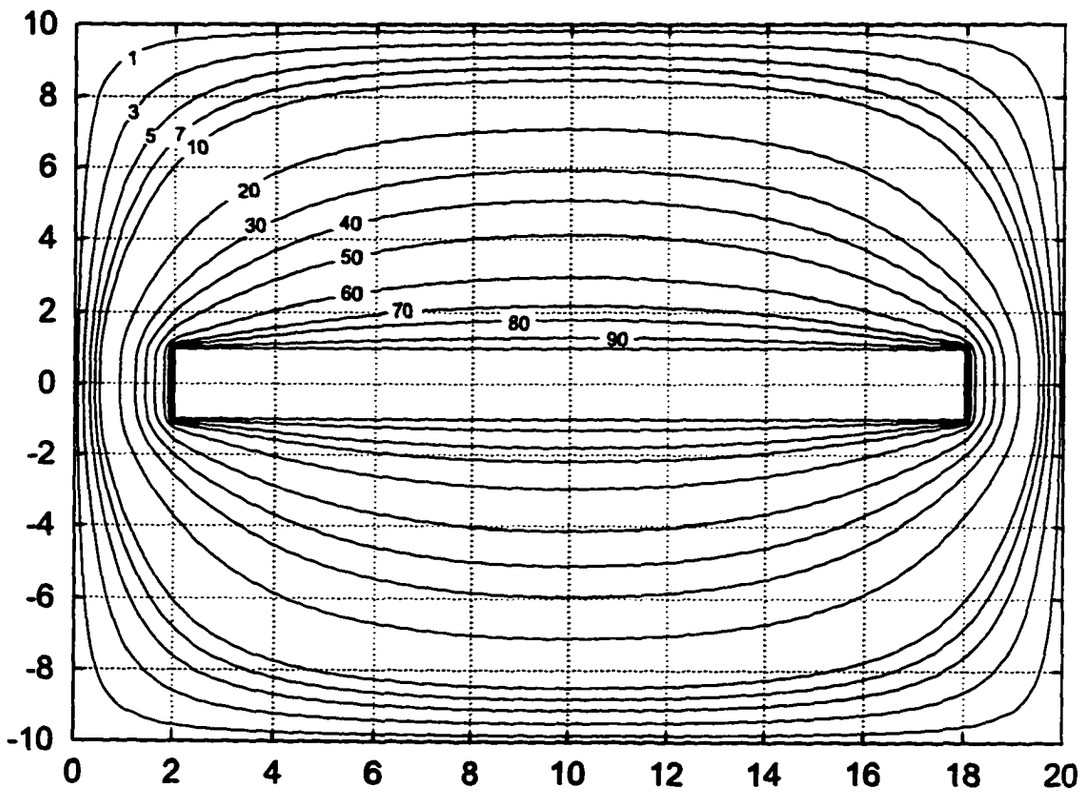

Figure 5.16: Equipotential contours of the potential depicted in figure 5.15.

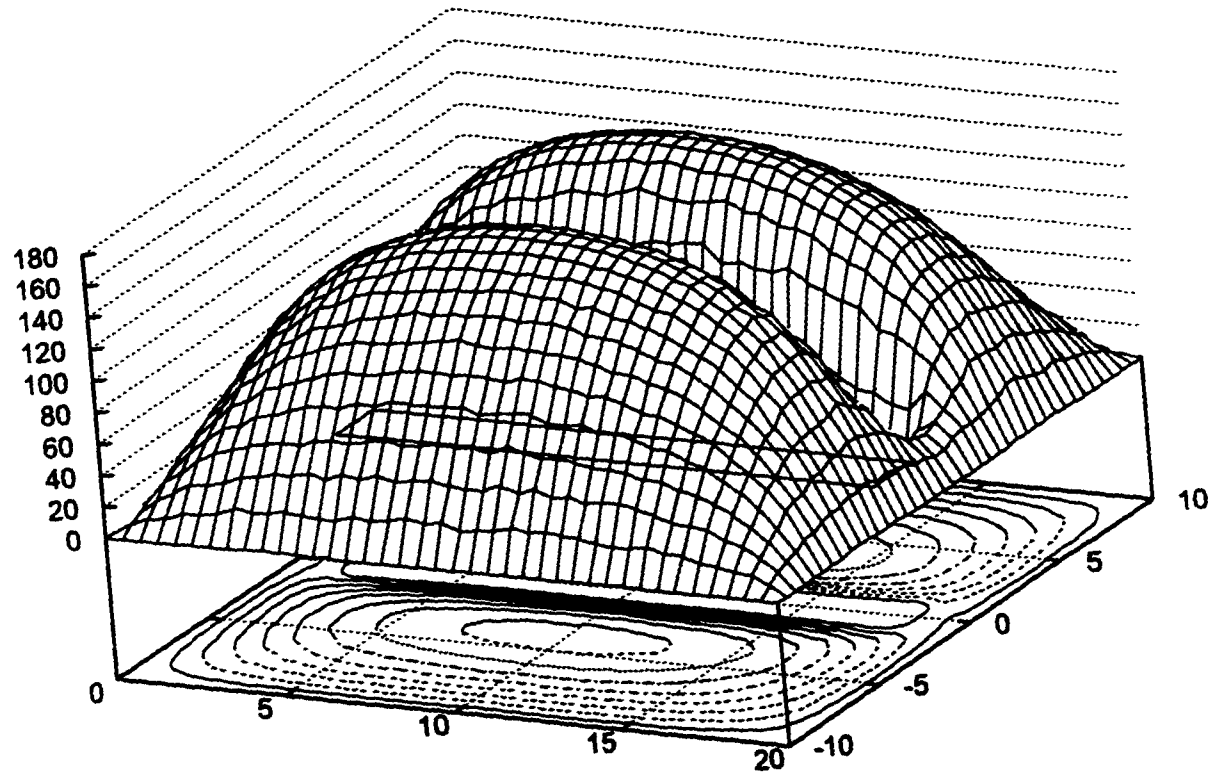

Figure 5.17: Average number of steps to reach a boundary during the MCR solution to the electrostatic lens of figure 5.13. There were $Nw = 50$ random walks started per point, and the underlying grid was 41×41.

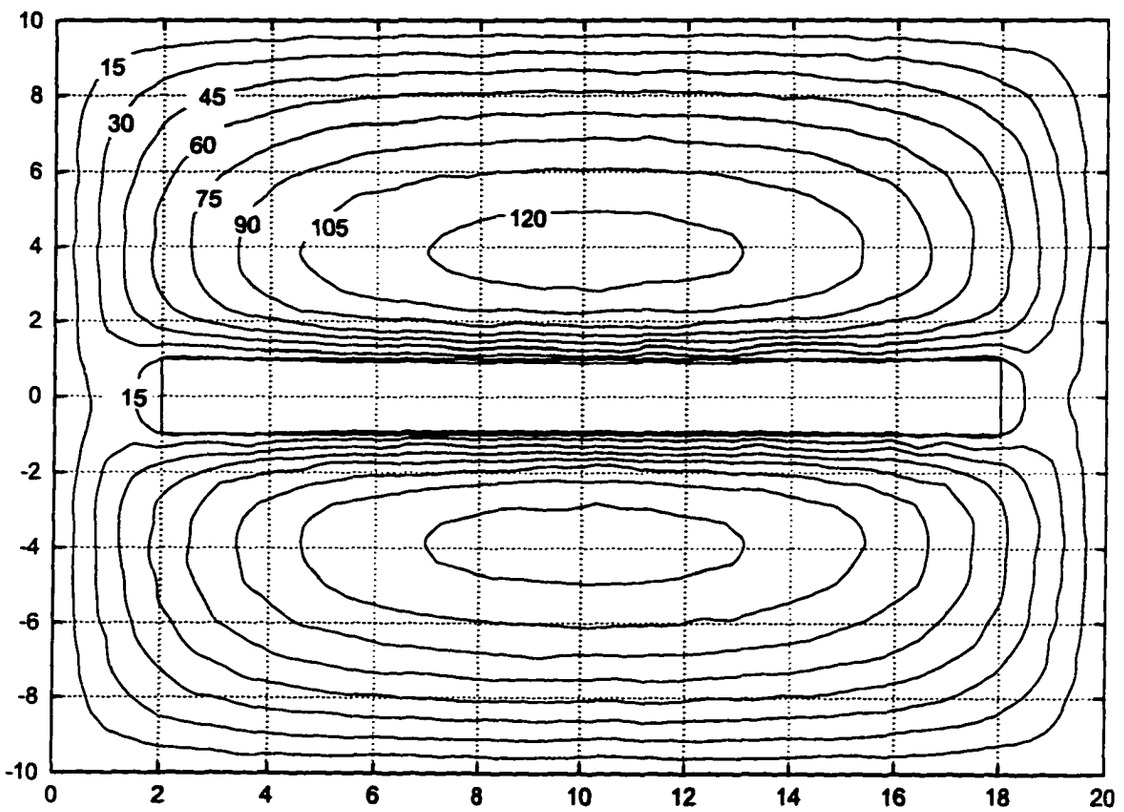

Figure 5.18: Average number of steps to reach a boundary during the MCR solution to the electrostatic lens of figure 5.13, shown as a contour plot. There were $N_W = 50$ random walks started per point, and the underlying grid was 41 × 41.

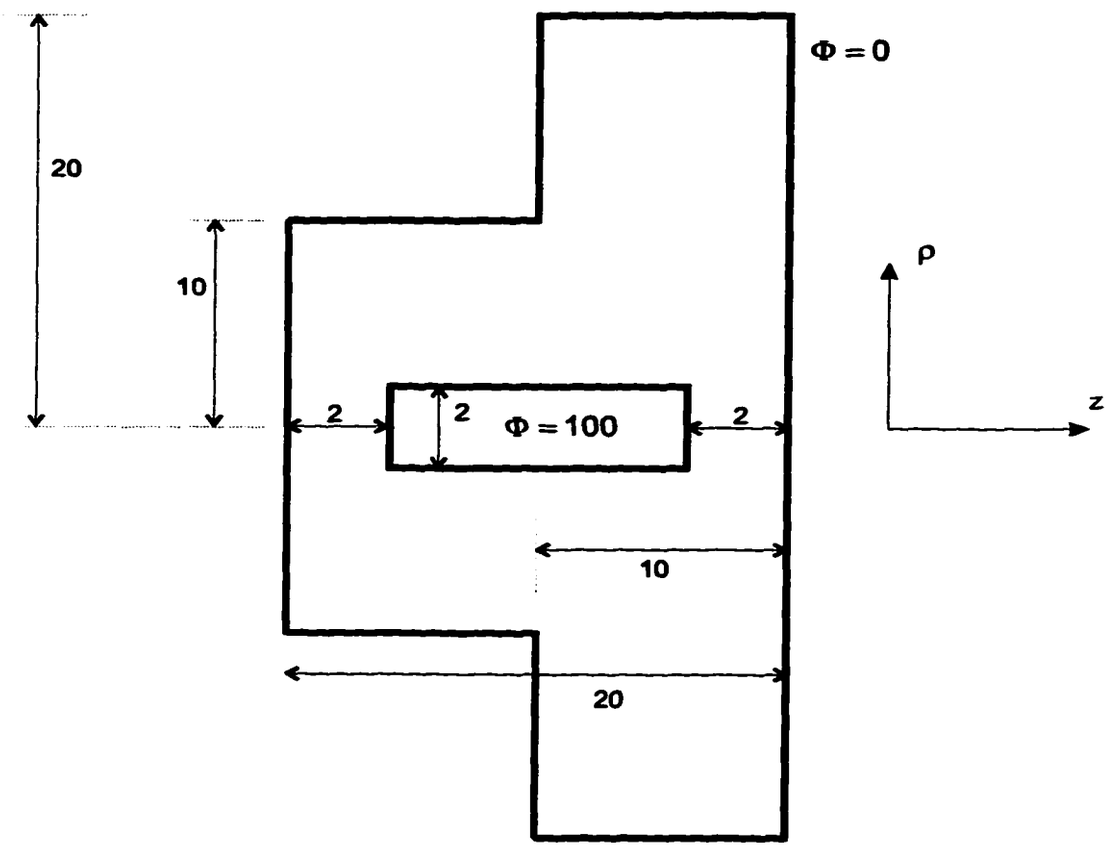

Figure 5.19: Electrostatic lens # 2 simulated in this work. A cross-section along the $z - \rho$ plane is shown. The lens is axially symmetric with respect to the z-axis and is in the form of two concentric cylinders, one with constant radius and one with a step discontinuity in the radius. The inner cylinder is maintained at 100V while the outer cylinder is grounded. After [2].

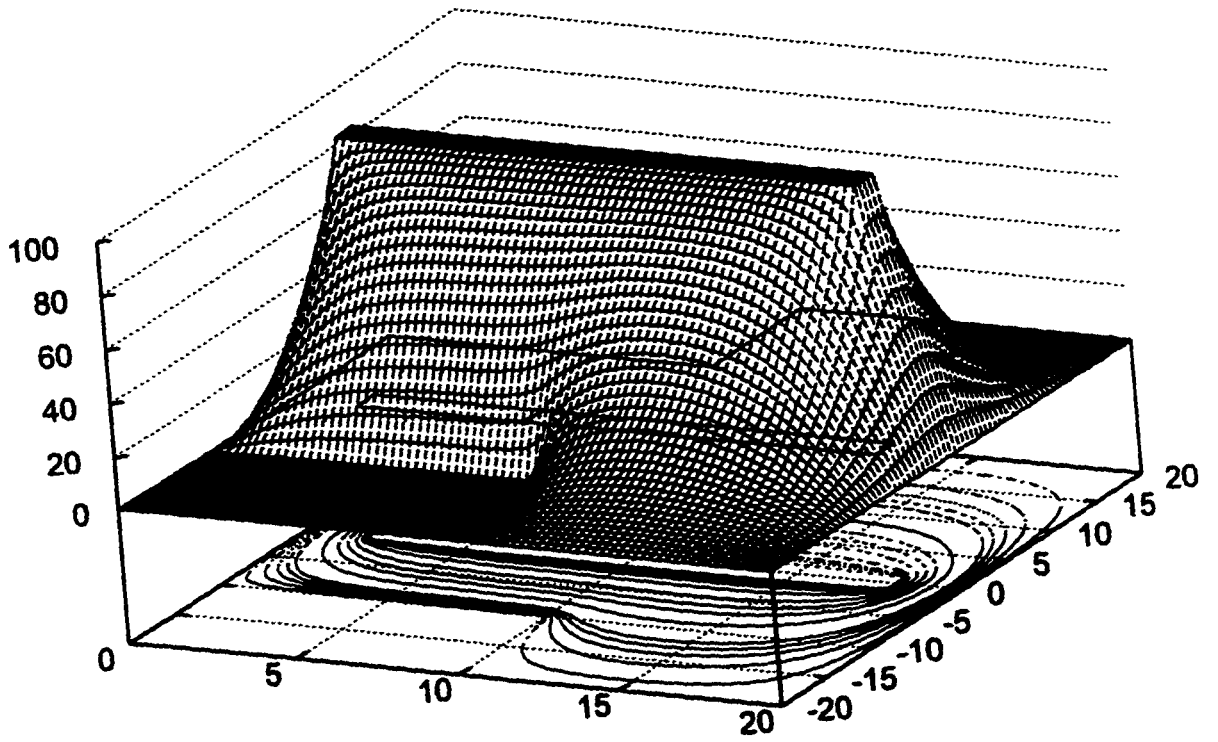

Figure 5.20: MCR solution to the electrostatic lens of figure 5.19. There were $N_W = 50$ random walks followed by 15 relaxations. The projections on the bottom of the figure are equipotential contours. The total CPU time required was 152 secs. (see text).

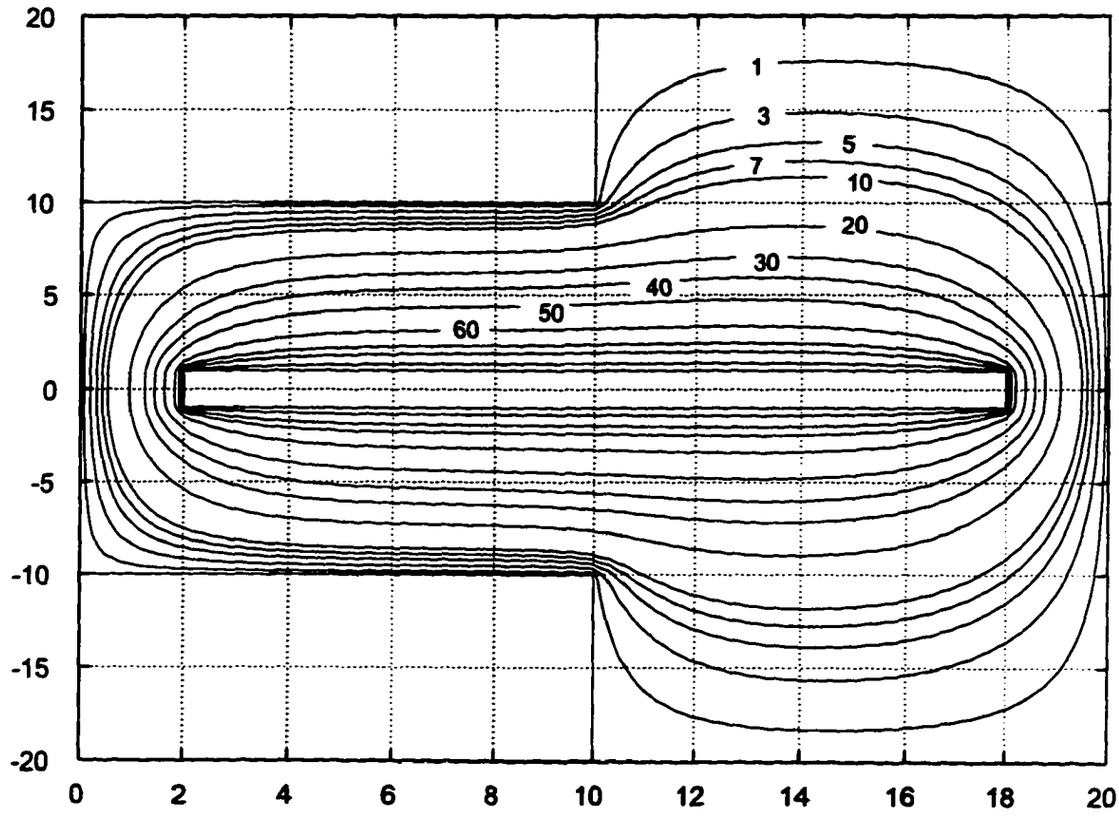

Figure 5.21: Equipotential contours of the potential depicted in figure 5.20.

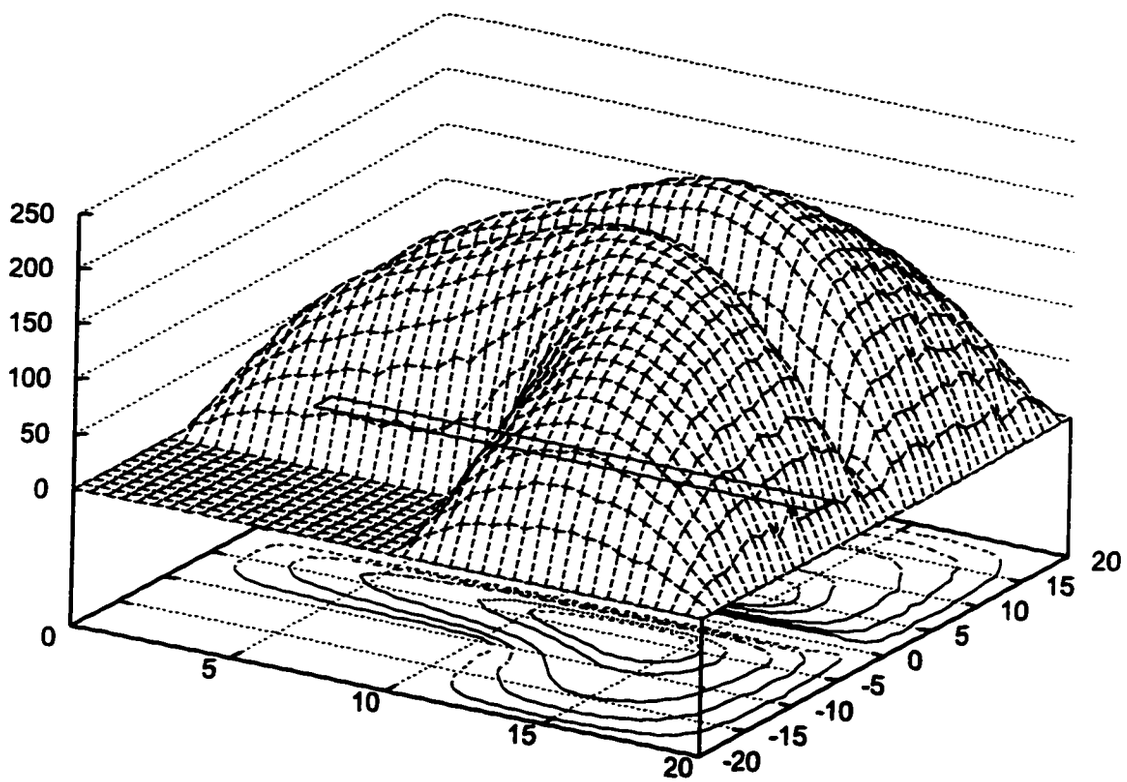

Figure 5.22: Average number of steps to reach a boundary during the MCR solution to the electrostatic lens of figure 5.19. There were $N_W = 50$ random walks started per point, and the underlying grid was 41 x 81.

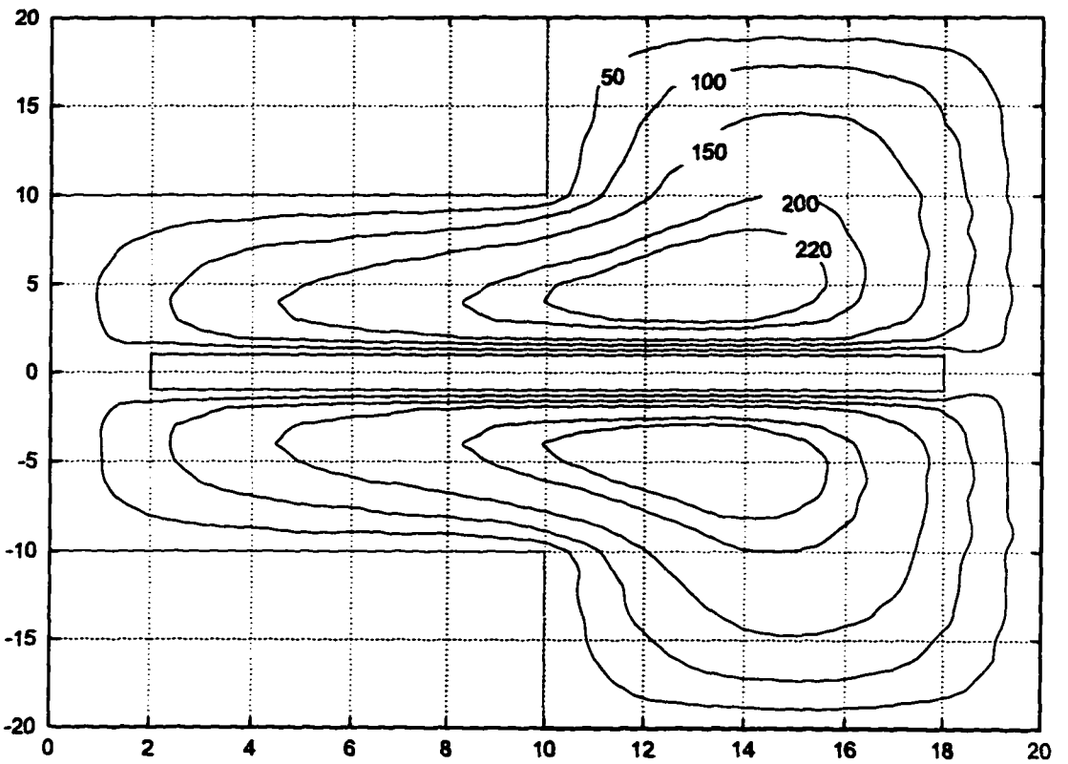

Figure 5.23: Average number of steps to reach a boundary during the MCR solution to the electrostatic lens of figure 5.19, shown as a contour plot. There were $N_W = 50$ random walks started per point, and the underlying grid was 41 × 81.

in real-world situations. We found that our MCR method provided results that match the solutions and are very competitive in terms of computational time required with the theoretical expectations or solutions obtained via other, deterministic, methods.

6 Summary and Conclusions

As we proceed to summarize our presented work, we list the disadvantages and advantages of the Monte Carlo and Monte Carlo Relaxation methods of solution of Laplace's equation as compared to traditional techniques. Apart from identifying the areas we considered strongly for optimization, this comparison puts our work in a proper perspective with regard to what we feel are its stronger points.

6.1 Disadvantages of Monte Carlo Methods

- The Monte Carlo method, as used up to now, is limited to certain types of partial differential equations of first or second order.

We showed how to alleviate part of the problem with the establishment of algorithms that solve the general elliptic problem with any desired accuracy, via random walks. Of course this approach does not address the question of how to solve parabolic or hyperbolic partial differential equations. Since some of these equations are amenable, however, to Green's functions techniques, and since, as we have seen, there is an intimate connection between the Green's functions of potential theory and transition probabilities of random walks, the author believes that there is a strong possibility of extending the theory to cover these equations as well. We consider this undertaking to be outside the scope of this work, however.

- A large number of random walks is required to obtain high accuracy, due to the diminishing speed of convergence once moderate accuracy has been obtained.

We showed how to improve the convergence and accuracy of the method by utilizing Monte Carlo simulations in the early part of the algorithm and subsequently switching to the faster converging relaxation-type simulation, when warranted. This procedure forms the core of our algorithm and has been described in detail in section 3.4.5. Several additional optimization techniques were also examined, as mentioned in section 3.4.

- During each random walk, a large number of steps is undertaken. At every step, it is essential to check whether a boundary has been reached or not, a time consuming process.

We showed how to significantly reduce the "housekeeping" type of computations by designing our algorithms to be not only efficient but intelligent as well, via the use of indexing techniques, multiple use of random numbers and various optimizations, as detailed in section 3.1.

6.2 Advantages of Monte Carlo Methods

- The potential at any given point can be calculated independently of the solution at any other point.

We consider this characteristic of Monte Carlo methods of paramount importance. It was pointed out in chapters 1 and 2 that the solution of Laplace's equation is of interest mainly near the optical axis in many systems. No other technique allows the isolation of sections of the domain from the boundary. The concentration of computational power only to the area of interest is a very important advantage of Monte Carlo methods. Parenthetically, we expect Monte Carlo methods to be applicable to field emission problems, where the potential and electric field are required only on the surface of the emitter. We have not seen any applications of stochastic methods in this area. Finally, we believe that this method lends itself to the ultimate in parallel computation: a cluster of processors can calculate the solution, one processor per point, independently of the other processors. We believe this to be an important and unique advantage of this method.

- The extension of the algorithm to higher dimensions is straightforward. Furthermore, the increase in execution time is small [114].

Although we concentrate on one- and two-dimensional problems in this work, we point out that the extension to three-dimensional geometries does not present any difficulty.

- The MCR algorithm is very simple, only slightly more complex than simple relaxation and easy to implement in a computer program with modest memory and intermediate storage requirements.

We implemented both one- and two-dimensional Laplace solvers utilizing the MCR method. Results of these programs have been presented in chapters 4 and 5.

- It is possible to simulate structures with complex geometries, irregular boundaries and discontinuities without any modifications to the algorithm.

The shapes of the electrostatic deflectors and lenses we simulate are examples of such geometries. In fact, we believe MCR is ideally suited to solve Laplace's equation in domains with fractal boundaries, a situation that appears naturally in field emission problems from semiconductor surfaces [115]. There is an extensive literature on how to construct the Laplacian operator in such domains and on fractal harmonic analysis, in general (see, for example, [116, 117, 118] and references therein).

- It is possible to solve the same problem with different boundary conditions "once and for all", with only one run of the program since what we really calculate is the Green's function and its derivative on the boundaries, which is independent of the boundary conditions.

We studied the dependence of the solution for an electrostatic deflector on the externally applied voltages. The circular geometry statistics were computed only once and the information was used to provide the potential distribution for various boundary conditions (voltages). Finally,

- MCR permits the numerical computation of Green's functions for arbitrary geometries.

Green's functions exist in tabulated form for a large number of partial differential equations and solution domains [77]. If the geometry is even slightly complex, however, analytical

expressions are not available. There are no unified approaches that allow the direct calculation of the Green's function for a particular problem that we are aware of. Monte Carlo methods on the other hand can be used for any geometry and domain, any number of dimensions and can model widely differing problems with the same algorithm. Results for a particular geometry can be obtained to arbitrary accuracy, limited only by computational time, and stored on memory or hard disk. Subsequent solution of boundary value problems for this geometry can be achieved via the reuse of these saved statistics, with minimal expenditure of CPU time.

- MC can be further optimized to reduce execution time.

In particular, the Single Point Monte Carlo method uses CPU time to perform "book-keeping" operations to automatically reduce the number of computationally expensive random walks. Furthermore it correlates errors of adjacent grid points leading to reduced error in the gradient of the potential, or electric field, which is what is generally wanted.

In concluding, we believe that MCR is a highly competitive and efficient method for the numerical solution of Laplace's equation and offers significant advantages in the simulation of electrostatic devices.

References

[1] P. Gash, "Improved numerical solutions of Laplace's equation," *Am. J. Phys.*, vol. 59, pp. 509–515, 1991.

[2] M. DiStasio and W. McHarris, "Electrostatic problems? Relax!," *Am. J. Phys.*, vol. 47, pp. 440–444, 1979.

[3] R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equation of mathematical physics," *Math. Annalen.*, vol. 32, 1928.

[4] H. Busch, "Berechnung der Bahn von Kathodenstrahlen im axialsymmetrischen electromagnetischen Felde," *Ann. Phys.*, vol. 81, pp. 974–993, 1926.

[5] C. Constantinescu and C. A., *Potential Theory on Harmonic Spaces.* Springer-Verlag, 1972.

[6] S. Axler, P. Bourdon, and W. Ramey, *Harmonic Function Theory.* Springer-Verlag, 1992.

[7] O. Kellogg, *Foundations of Potential Theory.* The Murray Printing Company, 1929.

[8] P. M. Morse and H. Feshbach, *Methods of Theoretical Physics - Part I.* McGraw-Hill, New York, 1953.

[9] P. Moon and D. Spencer, *Field Theory Handbook.* Springer-Verlag, 1961.

[10] R. Dautray and J.-L. Lions, *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 1: Physical Origins and Classical Methods.* Springer-Varlag, 1988.

[11] J. Jackson, *Classical Electrodynamics.* John Wiley & Sons, Inc., 1975.

[12] A. Malliaris and W. Brock, *Stochastic Methods in Economics and Finance.* North-Holland, 1982.

[13] A. Einstein, "Über die von molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen," *Ann. Phys.*, vol. 17, pp. 549–560, 1905.

[14] A. Einstein, "Zur Theorie der Brownschen Bewegung," *Ann. Phys.*, vol. 19, pp. 371–381, 1906.

[15] M. Smoluchowski, "Zarys teoryi kinetycznej ruchu Browna i roztworów metnych," *Ann. Phys.*, vol. 21, pp. 756–780, 1906.

[16] P. Langevin, "Sur la théorie du mouvement brownien," *Comptes Rendues*, vol. 146, pp. 530–533, 1908.

[17] P. Lévy, "Le mouvement brownien plan," *Amer. J. Math.*, vol. 62, pp. 487–550, 1940.

[18] P. Lévy, *Processus Stochastiques et Mouvement Brownien.* Gauthier-Villars, 1948.

[19] N. Wiener, "Differential space," *J. Math. and Phys.*, vol. 2, pp. 131–174, 1923.

[20] F. Spitzer, *Principles of Random Walk.* Springer-Verlag, 1976.

[21] B. Hughes, *Random Walks and Random Environments.* Oxford Science Publications, 1995.

[22] P. Révész, *Random Walk in Random and Non-Random Environments.* World Scientific, 1990.

[23] C. Domb, "Random walks and diffusion," *Am. J. Phys.*, vol. 46, pp. 49–56, 1978.

[24] P. Flory, *Statistical Mechanics of Chain Molecules.* Wiley, 1969.

[25] N. Filho, B. da Cunha, and M. Gomes, "Toy calculations to estimate critical exponents for random walks," *Am. J. Phys.*, vol. 64, pp. 305–308, 1996.

[26] N. Urakami and M. Takasu, "Multicanonical Monte Carlo simulation of a polymer with stickers," *Journal of the Physical Society of japan*, vol. 65, pp. 2694–2699, 1996.

[27] C. Tojo and P. Argyrakis, "Correlated random walk in continuous space," *Phys. Rev. E*, vol. 54, pp. 58–63, 1996.

[28] H. Stanley, *Introduction to Phase Transition and Critical Phenomena.* Oxford University Press, 1971.

[29] E. Raposo, S. de Oliveira, A. Nemirovsky, and M. Coutinho-Filho, "Random walks: A pedestrian approach to polymers, critical phenomena and field theory," *Am. J. Phys.*, vol. 59, pp. 633–645, 1991.

[30] I. Kosztin, B. Faber, and K. Schulten, "Introduction to the diffusion Monte Carlo method," *Am. J. Phys.*, vol. 64, pp. 633–644, 1996.

[31] N. Mishima, T. Petrosky, H. Minowa, and S. Goto, "Model experiment of two-dimentional Brownian motion with microcomputer," *Am. J. Phys.*, vol. 48, pp. 1050–1055, 1980.

[32] D. Amit, *Field Theory. The Renormalization Group and Critical Phenomena.* World Scientific, 1984.

[33] P. Ramond, *Field Theory. A Modern Primer.* Benjamin/Cummings, 1981.

[34] R. Sorensen, "The random walk method for dc circuit analysis," *Am. J. Phys.*, vol. 58, pp. 1056–1059, 1990.

[35] P. Doyle and J. Laurie Snell, *Random Walks and Electrical Networks.* The Mathematical Association of America, 1984.

[36] J. Snell, "Probability and martingales," *The Mathematical Intelligencer*, vol. 3, pp. 118–124, 1982.

[37] W. Nadler, T. Huang, and D. Stein, "Random walks on random partitions in one dimension," *Phys. Rev. E*, vol. 54, pp. 4037–4047, 1996.

[38] N. Lemke and I. Campbell, "Random walks in a closed space," *Physica A*, vol. 230, pp. 554–562, 1996.

[39] R. Bañuelos and T. Carroll, "Brownian motion and the fundamental frequency of a drum," *Duke Math. J.*, vol. 75, pp. 575–602, 1994.

[40] R. Bañuelos and T. Carroll, "Addendum to "Brownian motion and the fundamental frequency of a drum"," *Duke Math. J.*, vol. 82, p. 227, 1996.

[41] N. Madras and G. Slade, *The Self-Avoiding Walk*. Birkhäuser, 1993.

[42] V. Halpern, "Anti-persistent correlated random walks," *Physica A*, vol. 223, pp. 329–336, 1996.

[43] D. Poland, "Toeplitz matrices and random walks with memory," *Physica A*, vol. 223, pp. 113–124, 1996.

[44] J. Doob, *Classical Potential Theory and Its Probabilistic Counterpart*. Springer-Verlag, 1984.

[45] R. Hersch and R. Griego, "Brownian motion and potential theory," *Scient. Am.*, pp. 66–74, 1969.

[46] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.

[47] K. M. Decker, "The Monte Carlo method in science and engineering: Theory and application," *Comput. Meth. Appl. Mech. Engr.*, vol. 89, pp. 463–483, 1991.

[48] K. Binder, ed., *Monte Carlo Methods in Statistical Physics*, Springer-Verlag, 1979.

[49] K. Binder, ed., *Applications of the Monte Carlo Method in Statistical Physics*, Springer-Verlag, 1984.

[50] D. Heermann, *Computer Simulation Methods in Theoretical Physics*. Springer-Verlag, 1986.

[51] M. Kac, *Application of Statistical Methods to Differential and Integral Equations*. Lecture Notes Mass. Inst. Tech., 1949.

[52] J. H. Curtiss, "Sampling methods applied to differential and difference equations," *Proc. Sem. Sci. Comp.*, 1949.

[53] E. Yowell, "A Monte Carlo method of solving Laplace's equation," *Proc. Sem. Sci. Comp.*, 1949.

[54] M. Muller, "Some continuous monte carlo methods for the Dirichlet problem," *Ann. Math. Stats.*, vol. 27, p. 569, 1956.

[55] W. Wasow, "Random walks and the eigenvalues of elliptic difference equations," *J. Res. Nat. Bur. Stds.*, vol. 46, p. 65, 1951.

[56] J. von Plato, *Creating Modern Probability*. Cambridge University Press, 1994.

[57] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*. North-Holland, 1981.

[58] C. Steele, *Numerical Computation of Electric and Magnetic Fields*. Van Nostrand Reinhold Company, Inc., 1987.

[59] P. Zhou, *Numerical Analysis of Electromagnetics Fields*. Springer-Verlag, 1993.

[60] P. Silvester and R. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge University Press, 2nd ed., 1990.

[61] M. Chari and P. Silvester, eds., *Finite Elements in Electrical and Magnetic Field Problems*, John Wiley & Sons, 1980.

[62] K. Binns and P. Lawrenson, *Analusis and Computation of Electric Magnetic Field Problems*. The MacMillan Company, 1963.

[63] K. Binns and P. Lawrenson, *Analysis and Computation of Electric and Magnetic Field Problems*. Pergamon Press, 1973.

[64] K. Binns, P. Lawrenson, and C. Trowbridge, *The Analytical and Numerical Solution of Electric and Magnetic Fields*. John Wiley & Sons, 1993.

[65] E. Sadeh and M. Franklin, "Monte carlo solution of partial differential equations by special purpose digital computer," *IEEE Trans. Comp.*, vol. C-23, pp. 389–397, 1974.

[66] F. Hunt, J. Douglas, and J. Bernal, "Probabilistic computation of Poiseuille flow velocity fields," *J. Math. Phys.*, vol. 36, pp. 2386–2401, 1995.

[67] A. Kuznetsov, "Stochastic modeling of heating of a one-dimensional porous slab by a flow of incompressible fluid," *Acta Mechanica*, vol. 114, pp. 39–50, 1996.

[68] Y. Le Coz, R. Iverson, T.-L. Sham, H. Tiersten, and M. Shephard, "Theory of a floating random-walk algorithm for solving the steady-state heat equation in complex, materially inhomogeneous rectilinear domains," *Numerical Heat Transfer, Part B*, vol. 26, pp. 353–366, 1994.

[69] M. Szilagyi, *Electron and Ion Optics*. Plenum Press, 1988.

[70] P. Hawkes and E. Kasper, *Principles of Electron Optics. Volume 1: Basic Geometrical Optics.* Academic Press, 1989.

[71] P. Hawkes and E. Kasper, *Principles of Electron Optics. Volume 2: Applied Geometrical Optics.* Academic Press, 1989.

[72] P. Hawkes and E. Kasper, *Principles of Electron Optics. Volume 3: Wave Optics.* Academic Press, 1989.

[73] D. Gillespie, "The mathematics of Brownian motion and Johnson noise," *Am. J. Phys.*, vol. 64, pp. 235–240, 1996.

[74] T. Hida, *Brownian Motion.* Springer-Verlag, 1980.

[75] N. Tovmasyan, *Boundary Value Problems for Partial Differential Equations and Applications in Electrodynamics.* World Scientific, 1994.

[76] A. Tychonov and A. Samarski, *Partial Differential Equations of Mathematical Physics.* Holden-Day, Inc., 1964.

[77] A. Butkovskiy, *Green's Functions and Transfer Functions Handbook.* Ellis Horwood Limited, 1982.

[78] R. Durrett, *Brownian Motion and Martingales in Analysis.* Wadsworth, 1984.

[79] O. Pironneau, *Optimal Shape Design for Elliptic Systems.* Springer-Verlag, 1984.

[80] T. Booth, "Exact Monte Carlo solution of elliptic partial differential equations," *J. Comput. Phys.*, vol. 39, pp. 396–404, 1981.

[81] T. Booth, "Regional Monte Carlo solution of elliptic partial differential equations," *J. Comput. Phys.*, vol. 47, pp. 281–290, 1982.

[82] J. Vrbik, "Monte Carlo simulation of the general elliptic operator," *J. Phys. A:Math. Gen.*, vol. 20, pp. 2693–2697, 1987.

[83] O. Zienkiewicz, *The Finite Element Method in Engineering Science.* McGraw-Hill, 1971.

[84] D. Norrie and G. de Vries, *The Finite Element Method.* Academic Press, 1973.

[85] R. Harrington, *Field Computation by Moment Methods.* MacMillan, 1968.

[86] M. Jaswon and G. Symm, *Integral Equation Methods in Potential Theory and Elastostatics.* Academic Press, 1977.

[87] G. Forsythe and W. Wasow, *Finite Difference Methods for Partial Differential Equations.* Wiley, 1960.

[88] R. Varga, *Matrix Iterative Analysis.* Prentice-Hall, 1962.

[89] W. Ames, *Numerical Methods for Partial Differential Equations.* Nelson, 1969.

[90] E. Jacobs, D.H., *The State of the Art in Numerical Analysis.* Academic Press, 1977.

[91] F. Vesely, *Computational Physics: An Introduction.* Plenum, 1994.

[92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C, 2nd Ed.* Cambridge University Press, New York, 1992.

[93] D. Knuth, *The Art of Computer Programming. Vol. 2.* Addison-Wesley, 1969.

[94] H. Kobayashi, *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology.* Addison-Wesley, 1978.

[95] R. Rubinstein, *Simulation and the Monte Carlo Method.* John Wiley & Sons, 1981.

[96] P. Lewis and E. Orav, *Simulation Methodology for Statisticians, Operation Analysts, and Engineers.* Wadsworth & Brooks-Cole, 1989.

[97] R. Southwell, *Relaxation Methods in Engineering Science: A Treatise on Approximate Computation.* Oxford University Press, 1940.

[98] R. Southwell, *Relaxation Methods in Theoretical Physics, Vol.1.* Oxford at the Clarendon Press, 1946.

[99] R. Southwell, *Relaxation Methods in Theoretical Physics, Vol.2.* Oxford at the Clarendon Press, 1956.

[100] T. Crow, "Solutions to Laplace's equation using spreadsheets on a personal computer," *Am. J. Phys.*, vol. 55, pp. 817–823, 1987.

[101] W. MacDonald, "Discretization and truncation errors in a numerical solution of Laplace's equation," *Am. J. Phys.*, vol. 62, pp. 169–173, 1994.

[102] T. Henderson, *Discrete Relaxation Techniques.* Oxford University Press, 1990.

[103] A. Nachman, "A brief perspective on computational electromagnetics," *J. Comput. Phys.*, vol. 126, pp. 237–239, 1996.

[104] P. Garabedian, *Partial Differential Equations.* John Wiley and Sons, Inc., 1964.

[105] S. Wolfram, *The MATHEMATICA Book.* Cambridge University Press, 3rd ed., 1996.

[106] D. Stroock, *Probabilty Theory: An Analytic View.* Cambridge University Press, 1993.

[107] J. Patel and C. Read, *Handbook of the Normal Distribution.* Marcel Dekker, Inc., 1982.

[108] R. Elandt, "The folded normal distribution: Two methods of estimating parameters from moments," *Technometrics*, vol. 3, pp. 551–562, 1961.

[109] F. Hart, "Validating spreadsheet solutions to Laplace's equation," *Am. J. Phys.*, vol. 57, pp. 1027–1034, 1989.

[110] P. Hawkes, *Quadrupoles in Electron Lens Design.* Academic Press, 1970.

[111] M. Szilagyi and H. Cho, "Optimization of electrostatic deflectors," *J. Vac. Sci. Technol. B*, vol. 13, pp. 142–149, 1995.

[112] O. Oleinik and E. Radkevic, *Second Order Equations with Nonnegative Characteristic Form.* Plenum Press, 1973.

[113] A. Mitchell and D. Griffiths, *The Finite Difference Method in Partial Differential Equations.* John Wiley & Sons, 1980.

[114] K. Binder, "Theory and "Technical" aspects of Monte Carlo simulations," in *Monte Carlo Methods in Statistical Physics* (K. Binder, ed.), pp. 1–45, Springer-Verlag, 1979.

[115] A.-L. Barabási and H. Stanley, *Fractal Concepts in Surface Growth.* Cambridge University Press, 1995.

[116] J. Kigami, "Laplacians on self-similar sets and their spectral distributions," in *Fractal Geometry and Stochastics* (C. Bandt, S. Graf, and M. Zähle, eds.), pp. 221–238, Birkhäuser, 1995.

[117] A. Jonsson, "Brownian motion on fractals and function spaces," *Math. Z.*, vol. 222, pp. 495–504, 1996.

[118] Z.-M. Yin, "New methods for simulation of fractional Brownian Motion," *J. Comput. Phys.*, vol. 127, pp. 66–72, 1996.

[119] I. Stakgold, *Boundary Value Problems of Mathematical Physics, Vol. II.* The MacMillan Company, 1968.

Appendix A: A Formal Proof of the Main Theorem

In this appendix we present a formal proof of the Main Theorem. Ours is modeled along the lines of the proof presented in reference [66] but we have modified it to account for a non-zero boundary condition (non-homogeneous Dirichlet problem) and to solve a general Poisson's equation rather than Laplace's equation.

The proof proceeds in two parts: first the deterministic equation to be solved is cast as a stochastic random walk problem and, subsequently, the derived problem is solved.

For generality, we will consider the case of a d-dimensional space R^d. The random walk takes place on a uniform grid of points in R^d with spacing $h > 0$. Each grid point has then $2d$ nearest neighbors. Each vector $\mathbf{x} = (x_1, x_2, \cdots, x_d)$ whose end rests on a grid point, can then be expressed as:

$$\mathbf{x} = \sum_{i=1}^{d} x_i \mathbf{e}_i, \tag{A.1}$$

where $\mathbf{e}_i$ is the unit vector along the i-th axis and the coordinates x_i are all integral multiples of the spacing, $x_i = k_i h$, $k_i \in Z$. These relations completely specify the lattice embedded in R^d. We will call the set of all vectors with end points in the lattice H^d.

We now seek stochastic approximations to the solution of the equation:

$$\nabla^2 u(\mathbf{x}) = -f(\mathbf{x}), \tag{A.2}$$

where $\mathbf{x} \in \bar{\Omega} = (\Omega \cup \partial\Omega) \subset R^d$, $f(\mathbf{x})$ is a known function in Ω and u takes known values on the boundary $\partial\Omega$:

$$u(\mathbf{x}) = g(\mathbf{x}), \qquad \mathbf{x} \in \partial\Omega. \tag{A.3}$$

We assume that a uniform lattice such as described previously has been defined. The discrete approximation of the Laplacian operator in H^d is then:

$$\nabla^2 u(\mathbf{x}) \equiv \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2} u(\mathbf{x}) \simeq \sum_{\substack{k=-d \\ k\neq 0}}^{d} \frac{u(\mathbf{x}+h\mathbf{e}_k) - (2d)u(\mathbf{x})}{h^2}, \qquad \mathbf{x} \in H^d. \tag{A.4}$$

The summation here runs over all the $2d$ nearest neighbors of $\mathbf{x}$, excluding $\mathbf{x}$ itself.

Consider now a simple, symmetric random walk that proceeds in H^d and starts at $\mathbf{x}$. The probability that after one step the walk is at $\mathbf{y}$, is:

$$P_1(\mathbf{x},\mathbf{y}) = \begin{cases} 1/(2d) & \text{if } \mathbf{y} = \mathbf{x} + h\mathbf{e}_k \text{ for some } k = 1 \cdots d, \\ 0 & \text{otherwise.} \end{cases} \tag{A.5}$$

Consider now any real function $q(\mathbf{x})$ defined on H^d. We define an operator P_1 on the set of such functions as:

$$P_1 q(\mathbf{x}) = \mathbf{E}_{\mathbf{x}}\left[q(\mathbf{x}_1)\right], \tag{A.6}$$

where $\mathbf{x}_1$ is the position of the random walk after the first step and $\mathbf{E}_{\mathbf{x}}$ is the expectation value along all possible ending points after the 1st step. Because of the Markovian property of the random walk, (A.6) can be cast as:

$$P_1 q(\mathbf{x}) = \sum_{\mathbf{y}\in H^d} P_1(\mathbf{x},\mathbf{y}) q(\mathbf{y}), \tag{A.7}$$

and, with the help of (A.5):

$$P_1 q(\mathbf{x}) = \left(\frac{1}{2d}\right) \sum_{\substack{k=-d \\ k\neq 0}}^{d} q(\mathbf{x}+h\mathbf{e}_k). \tag{A.8}$$

With the help of (A.8), the discrete approximation of the Laplacian, (A.4), becomes:

$$\nabla^2 u(\mathbf{x}) \simeq \left(\frac{2d}{h^2}\right)(P_1 - I)\, u(\mathbf{x}), \tag{A.9}$$

where I is the identity operator.

Finally, combining (A.9) and (A.2), the problem is reduced to finding $u(\mathbf{x})$ such that it satisfies:

$$(P_1 - I)\, u(\mathbf{x}) = -\left(\frac{h^2}{2d}\right) f(\mathbf{x}), \qquad \mathbf{x} \in H^d. \tag{A.10}$$

We have thus reduced the deterministic equation (A.2) with solutions in R^d to a stochastic random walk problem (A.10) with solutions in H^d. We now proceed to solve this equation.

First, we will generalize the definition of the operator P_1, equation (A.6):

$$P_n q(\mathbf{x}) = \mathbf{E}_{\mathbf{x}}\left[q(\mathbf{x}_n)\right], \tag{A.11}$$

where $\mathbf{x}_n$ is the position of the walk after the n-th step, having started at $\mathbf{x}$. P_n gives the mathematical expectation value of the function $q(\mathbf{x})$ over all possible paths of the random walk in n steps. Consider now the alternative form of P_n, equivalent to (A.7):

$$P_n q(\mathbf{x}) = \sum_{\mathbf{y} \in H^d} P_n(\mathbf{x}, \mathbf{y}) q(\mathbf{y}), \tag{A.12}$$

where now $P_n(\mathbf{x}, \mathbf{y})$ is the probability that the random walk starts at $\mathbf{x}$ and after n steps arrives at $\mathbf{y}$. The Markovian property of the random walk can now be expressed in terms of the properties of P_n:

$$P_{n+1} q(\mathbf{x}) = P \cdot P_n q(\mathbf{x}), \tag{A.13}$$

In other words, the expectation value of $q(\mathbf{x})$ along the random walk is the same whether we perform the walk in $n+1$ steps, or in n steps, stop and then perform a new walk of only 1 step. The proof of (A.13) follows from the statistical independence of the steps of the random walk (the Markovian property, in other words):

$$P_{n+1}(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{z}\in H^d} P_1(\mathbf{x},\mathbf{z})P_n(\mathbf{z},\mathbf{y}). \tag{A.14}$$

Now, we consider a random walk in H^d that starts at $\mathbf{x}$. Since the original domain of the problem, Ω, is finite by assumption, so is the lattice H^d embedded in Ω. Therefore the random walk will reach the boundary $\partial\Omega$ in a finite time, which we call τ, and the random walk will terminate.

Let now q be an arbitrary, bounded real valued function: $u : H^d \to R$. We define the family of functions ϕ_α for $0 < \alpha < 1$ by:

$$(I - \alpha P_1)\, q = \phi_\alpha. \tag{A.15}$$

Equation (A.15) can be solved for q as:

$$q(\mathbf{x}) = (I - \alpha P_1)^{-1}\phi_\alpha = \sum_{n=0}^{\infty} \alpha^n P_1^n \phi_\alpha(\mathbf{x}). \tag{A.16}$$

But by virtue of (A.13), $P_1^n = P_n$. Hence (A.16) becomes:

$$q(\mathbf{x}) = \sum_{n=0}^{\infty} \alpha^n P_n \phi_\alpha(\mathbf{x}) = \mathbf{E}_{\mathbf{x}}\left[\sum_{n=0}^{\infty} \alpha^n \phi_\alpha(\mathbf{x}_n)\right]. \tag{A.17}$$

Now, if τ is the time (or step) at which the random walk reaches the boundary, we can rewrite this as:

$$q(\mathbf{x}) = \mathbf{E}_{\mathbf{x}}\left[\sum_{n=0}^{\infty} \alpha^n \phi_\alpha(\mathbf{x}_n)\right] = \mathbf{E}_{\mathbf{x}}\left[\sum_{n=0}^{\tau-1} \alpha^n \phi_\alpha(\mathbf{x}_n)\right] + \mathbf{E}_{\mathbf{x}}\left[\sum_{n=\tau}^{\infty} \alpha^n \phi_\alpha(\mathbf{x}_n)\right]. \tag{A.18}$$

But,

$$\mathbf{E}_{\mathbf{x}}\left[\sum_{n=\tau}^{\infty}\alpha^{n}\phi_{\alpha}(\mathbf{x}_{n})\right]=\mathbf{E}_{\mathbf{x}}\left[\alpha^{\tau}\sum_{k=0}^{\infty}\alpha^{k}\phi_{\alpha}(\mathbf{x}_{\tau+k})\right], \tag{A.19}$$

and

$$\mathbf{E}_{\mathbf{x}}\left[\phi_{\alpha}(\mathbf{x}_{\tau+k})\right]=P_{k+\tau}\phi_{\alpha}(\mathbf{x})=P_{k}P_{\tau}\phi_{\alpha}(\mathbf{x})=\mathbf{E}_{\mathbf{x}}\left[\mathbf{E}_{\mathbf{x}_{\tau}}\left[\phi_{\alpha}(\mathbf{x}_{k})\right]\right]. \tag{A.20}$$

This is, once more, a statement of the Markovian property of the random walk: the expectation value of the function $\phi_{\alpha}(\mathbf{x})$ along the path of the walk at time $\tau+k$ is the same whether we consider the walk to proceed continuously from time 0 to $\tau+k$, or to stop at time τ, when the boundary is reached and, then a new walk of duration k is run.

Combining (A.18), (A.19) and (A.20), we write:

$$q(\mathbf{x})=\mathbf{E}_{\mathbf{x}}\left[\sum_{n=0}^{\tau-1}\alpha^{n}\phi_{\alpha}(\mathbf{x}_{n})\right]+\mathbf{E}_{\mathbf{x}}\left[\alpha^{\tau}\mathbf{E}_{\mathbf{x}_{\tau}}\left[\sum_{k=0}^{\infty}\alpha^{k}\phi_{\alpha}(\mathbf{x}_{k})\right]\right]. \tag{A.21}$$

Using (A.17), this finally becomes:

$$q(\mathbf{x})=\mathbf{E}_{\mathbf{x}}\left[\sum_{n=0}^{\tau-1}\alpha^{n}\phi_{\alpha}(\mathbf{x}_{n})\right]+\mathbf{E}_{\mathbf{x}}\left[\alpha^{\tau}q(\mathbf{x}_{\tau})\right]. \tag{A.22}$$

Taking the limit $\alpha\rightarrow 1$ and using (A.15), we have:

$$q(\mathbf{x})=\mathbf{E}_{\mathbf{x}}\left[\sum_{n=0}^{\tau-1}(I-P_{1})\,q(\mathbf{x}_{n})\right]+\mathbf{E}_{\mathbf{x}}\left[q(\mathbf{x}_{\tau})\right]. \tag{A.23}$$

This result is true for any discrete function in H^{d}, provided that the random walk terminates after a finite number of steps, i.e. the exit time is finite.

Now, the solution to the original problem, equation (A.2), subject to the boundary condition (A.3) is given as the sum of two terms: the homogeneous solution, $u_{h}(\mathbf{x})$, which

satisfies the equation:

$$\nabla^2 u_h(\mathbf{x}) = 0, \qquad u_h(\mathbf{x}) = g(\mathbf{x}) \qquad \text{for} \qquad \mathbf{x} \in \partial\Omega, \tag{A.24}$$

and a particular solution, $u_p(\mathbf{x})$, satisfying:

$$\nabla^2 u_p(\mathbf{x}) = -f(\mathbf{x}), \qquad u_p(\mathbf{x}) = 0 \qquad \text{for} \qquad \mathbf{x} \in \partial\Omega. \tag{A.25}$$

Applying equation (A.23) to the functions $u_h(\mathbf{x})$ and $u_p(\mathbf{x})$, we finally obtain:

$$u(\mathbf{x}) = u_h(\mathbf{x}) + u_p(\mathbf{x}) = \left(\frac{h^2}{2d}\right) \mathbf{E}_{\mathbf{x}} \left[\sum_{n=0}^{\tau-1} f(\mathbf{x}_n)\right] + \mathbf{E}_{\mathbf{x}}\left[g(\mathbf{x}_\tau)\right]. \tag{A.26}$$

We can obtain a useful result by setting $f = 2d/h^2$ and $g(\mathbf{x}) = 0$ for $\mathbf{x} \in \partial\Omega$:

$$u(\mathbf{x}) = \mathbf{E}_{\mathbf{x}} \left[\sum_{n=0}^{\tau-1} (1)\right]. \tag{A.27}$$

This remarkable result states that the expected number of steps for a random walk starting at $\mathbf{x}$ to reach the boundary is given by the solution to:

$$\nabla^2 u(\mathbf{x}) = -\frac{2d}{h^2}, \qquad u(\mathbf{x}) = 0 \quad \text{for} \quad \mathbf{x} \in \partial\Omega. \tag{A.28}$$

We will make extensive use of this fact. Finally, comparing equation (A.26) with the corresponding deterministic solution of the interior Dirichlet problem [119]:

$$u(\mathbf{x}) = \int_\Omega G_D(\mathbf{x}|\mathbf{x}') f(\mathbf{x}') d\mathbf{x}' - \int_{\partial\Omega} g(\mathbf{x}) \frac{\partial G_D(\mathbf{x}'|\mathbf{x}')}{\partial n} d\mathbf{x}', \tag{A.29}$$

where $G_D(\mathbf{x}|\mathbf{x}')$ is the Green's function for the Dirichlet problem (A.2) and $\partial/\partial n$ denotes the directional derivative along the boundary $\partial\Omega$, we see that we can immediately make the following identifications:

- The fraction of random walks that start at point $\mathbf{x}$ and end at point $\mathbf{x}'$ on the boundary is equal to $-\partial G_D(\mathbf{x}'|\mathbf{x}')/\partial n$.

- The fraction of random walks that start at point $\mathbf{x}$ and pass through point $\mathbf{x}'$, where $\mathbf{x}'$ now is in the interior of the domain, is simply $(2d/h^2)G_D(\mathbf{x}|\mathbf{x}')$.

Appendix B: Extension of the Main Theorem

We seek stochastic approximations to the solution of the equation:

$$L[u(\mathbf{x})] = -f(\mathbf{x}), \tag{B.1}$$

where $\mathbf{x} \in \bar{\Omega} = (\Omega \cup \partial\Omega) \subset R^d$, $f(\mathbf{x})$ is a known function in Ω and u takes known values on the boundary $\partial\Omega$:

$$u(\mathbf{x}) = g(\mathbf{x}), \qquad \mathbf{x} \in \partial\Omega. \tag{B.2}$$

We assume that a uniform lattice such as described previously has been defined.The discrete approximation of the general elliptic operator in H^d is then (from equation 5.18):

$$L[u(\mathbf{x})] \simeq \alpha_0(\mathbf{x})[\sum_{\substack{k=-d \\ k\neq 0}}^{d} \lambda_k(\mathbf{x})u(\mathbf{x}+h\mathbf{e}_k) - u(\mathbf{x})], \qquad \mathbf{x} \in H^d. \tag{B.3}$$

The summation here runs aver all the $2d$ nearest neighbors of $\mathbf{x}$, excluding $\mathbf{x}$ itself, and the dependence of the coefficients in the discretization, $\alpha_0(\mathbf{x})$ and $\lambda_k(\mathbf{x})$ on the starting position, $\mathbf{x}$, has been shown explicidy. In the two-dimensional case, equation 5.18, $\alpha_0(\mathbf{x})$ corresponds to $\alpha_{i,j}$ and $\lambda_k(\mathbf{x})$ to $\lambda_{i-1,j}$, $\lambda_{i+1,j}$ $\lambda_{i,j-1}$ or $\lambda_{i,j+1}$, depending on the value of k.

Consider now a non-simple, non-symmetric random walk that proceeds in H^d and starts at $\mathbf{x}$. The probability that after one step the walk is at $\mathbf{y}$, is:

$$P_1(\mathbf{x},\mathbf{y}) = \begin{cases} \lambda_k & \text{if } \mathbf{y} = \mathbf{x} + h\mathbf{e}_k \text{ for some } k = 1 \cdots d, \\ 0 & \text{otherwise.} \end{cases} \tag{B.4}$$

Proceeding in an entirely analogous manner as in appendix A, we obtain the generalization of (A.8):

$$P_1 q(\mathbf{x}) = \sum_{\substack{k=-d \\ k\neq 0}}^{d} \lambda_k q(\mathbf{x} + h\mathbf{e}_k). \tag{B.5}$$

With the help of (B.5), the discrete approximation of L, (B.3), becomes:

$$L[u(\mathbf{x})] \simeq \alpha_0(\mathbf{x})(P_1 - I)u(\mathbf{x}), \tag{B.6}$$

which is the generalization of (A.9).

Finally, combining (B.5) and (B.1), the problem is reduced to finding $u(\mathbf{x})$ such that it satisfies:

$$(P_1 - I)\, u(\mathbf{x}) = -\left(\frac{1}{\alpha_0(\mathbf{x})}\right) f(\mathbf{x}), \qquad \mathbf{x} \in H^d. \tag{B.7}$$

Again, by identical arguments as those following (A.10), it is straightforward to write the solution to the general elliptic problem:

$$u(\mathbf{x}) = \left(\frac{1}{\alpha_0(\mathbf{x})}\right) \mathbf{E}_{\mathbf{x}} \left[\sum_{n=0}^{\tau-1} f(\mathbf{x}_n)\right] + \mathbf{E}_{\mathbf{x}} \left[g(\mathbf{x}_\tau)\right]. \tag{B.8}$$

This is the generalization of (A.26) to the general elliptic problem.

Finally, from this it follows that the number of steps required for a random walk starting at $\mathbf{x}$ to reach the boundary is given by the solution to:

$$L[u(\mathbf{x})] = -\alpha_0(\mathbf{x}), \qquad u(\mathbf{x}) = 0 \quad \text{for} \quad \mathbf{x} \in \partial\Omega. \tag{B.9}$$

Vita

Elias Kougianos received his Bachelor of Science degree in Electrical Engineering from the University of Patras, Patras, Greece, in 1985. He then joined the Electrical and Computer Engineering department at Louisiana State University, receiving his Master of Science degree in Electrical Engineering in 1987 and his Master of Science degree in Physics in 1988. After a short interlude where he was employed as a Software Engineer, he joined Texas Instruments, Inc., in Houston, Texas, as an Electrical Design Engineer. His main responsibilities included the modeling and simulation of Flash EEPROM cell designs. He returned to Louisiana State University in 1991 and since then has been enrolled in the doctoral program in the Department of Electrical and Computer Engineering. The degree of Doctor of Philosophy will be conferred in December, 1997. His research interests include the application of computers to large-scale simulations of processes in semiconductors, mathematical effective mass theory, numerical electromagnetics and the application of stochastic methods for the solution of deterministic problems. Outside academia, he enjoys reading and listening to music.

DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Elias Kougianos

Major Field: Electrical Engineering

Title of Dissertation: Stochastic Techniques for the Solution of Electrostatic Problems with Applications to Electron Optics

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

July 21, 1997