

UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

**Application of Quasi-Random Sequences
to Monte Carlo Methods
(Ph.D. Thesis)**

Bradley Scott Moskowitz

September 1993

CAM Report 93-35

**Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555**

UNIVERSITY OF CALIFORNIA

Los Angeles

Application of Quasi-Random Sequences to Monte Carlo Methods

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Mathematics

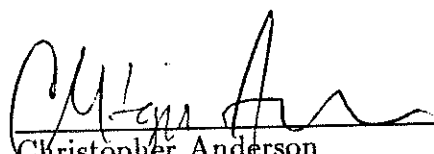
by

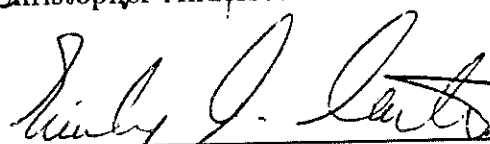
Bradley Scott Moskowitz

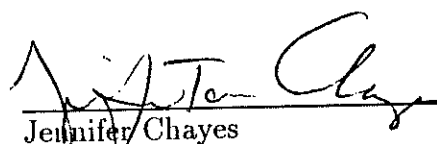
1993

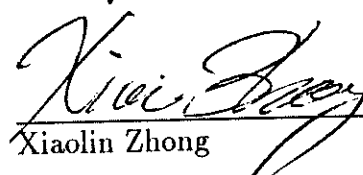
© Copyright by
Bradley Scott Moskowitz
1993

The dissertation of Bradley Scott Moskowitz is approved.


Christopher Anderson


Emily Carter


Jennifer Chayes


Xiaolin Zhong


Russel Caflisch, Committee Chair

University of California, Los Angeles

1993

DEDICATION

I dedicate this to

- my parents Bert and Eileen Moskowitz
- my brother Stephen Moskowitz

for all the love and support they have given me.

TABLE OF CONTENTS

I	Monte Carlo Integration and Quasi-Random Sequences	1
1	Introduction	2
2	Monte Carlo Integration	5
2.1	Statistical Basics	5
2.2	Statistical Error Measures and Convergence Rates	7
2.2.1	Expectation and Variance of Sums	8
2.3	Crude Monte Carlo Estimation	9
2.4	Simple Integration Example	11
2.4.1	Discussion of Computations	11
3	Quasi-Random Sequences for Monte Carlo Integration	17
3.1	Discrepancy	17
3.2	Koksma-Hlawka Inequality and Convergence	19
3.3	The Halton Sequence	23
3.4	Other Quasi-Random Sequences	25
4	Variance Reduction Using Importance Functions	26
4.1	Importance Sampling	27

4.1.1	The Rejection Method	29
4.2	Weighted Uniform Sampling	30
4.3	Statistical Convergence Results	32
II	Applying Quasi-Random Sequences	37
5	Discontinuity Avoidance	38
5.1	Reduction of Convergence Rate with a Discontinuous Integrand . .	38
5.2	Example: Integration of Characteristic Functions	41
5.3	Application to Variance Reduction with Importance Functions . .	48
5.3.1	Discontinuous Nature of the Rejection Method	49
5.3.2	Weighted Uniform Sampling as a Continuous Alternative .	51
5.3.3	The Smoothed Rejection Method	52
5.4	Computational Examples and Results	55
5.4.1	Computations for Variance Reduction Examples	57
5.4.2	Description of Examples, Computational Results	61
5.5	Discussion of Results, Conclusions	75
6	Reduction of Dimensions	77
6.1	Ineffectiveness of Quasi-Random Sequences for High Dimensions .	77
6.2	Computational Examples: Reordered Dimensions	80
6.3	Stochastic Process Simulation: Feynman-Kac Formula	87
6.3.1	Stochastic Basics, Wiener Process	87

6.3.2	Stochastic Differential Equations, and the Ito Formula . . .	94
6.3.3	Proof of Feynman-Kac Formula, Conversion to Backwards Time	96
6.3.4	Time Discretization, Monte Carlo Estimates	100
6.3.5	Application of Quasi-Random Sequences, Reformulation . .	103
6.4	Computational Example for the Feynman-Kac Formula	110
III	Advanced Applications : Stochastic Processes	120
7	Quantum Monte Carlo Methods	121
7.1	Diffusion Monte Carlo - Underlying Theory, Convergence Result .	123
7.2	Trial Function Reformulation	129
7.3	Theoretical Ground State Energy Approximations	132
7.4	Connections to Stochastic Differential Equations	135
7.5	RMS Convergence, Time Discrete Stochastic Simulations	142
7.6	Diffusion Monte Carlo Estimates	150
7.7	Continuation and Empirical Transformation Methods	155
7.7.1	Empirical Transformation Method	162
7.7.2	Correlated Correlation	166
7.8	Diffusion Monte Carlo Examples	168
8	Conclusion	183

A Moments of Sums of Random Variables	185
Bibliography	189

LIST OF FIGURES

2.1	Log-Log Plots for Example 1.	15
5.1	2D Versions of Functions used in Example 2.	43
5.2	Log-Log Plot for Example 2.	45
5.3	Log-Log Plot vs $t(\text{sec})$ for Example 2.	46
5.4	Log-Log Plot for Example 3.	69
5.5	Log-Log Plot vs $t(\text{sec})$ for Example 3.	70
5.6	Log-Log Plot for Example 4.	71
5.7	Log-Log Plot vs $t(\text{sec})$ for Example 4.	72
5.8	Log-Log Plot for Example 5.	73
5.9	Log-Log Plot vs $t(\text{sec})$ for Example 5.	74
6.1	Convergence Results and Log-Log Plot for Example 6, 100 trials. .	83
6.2	Log-Log Plot for Example 7	84
6.3	Log-Log Plot for Example 8, 75 trials	85
6.4	Graphical Comparison of Discretization Schemes.	109
6.5	Log-Log Plot for Example 9, $T = 0.02$, $m = 8$	114
6.6	Log-Log Plot vs. $t(\text{sec})$ for Example 9, $T = 0.02$, $m = 8$	115
6.7	Log-Log Plot for Example 9, $T = 0.04$, $m = 16$	116

6.8	Log-Log Plot vs $t(\text{sec})$ for Example 9, $T = 0.04$, $m = 16$	117
6.9	Log-Log Plot for Example 9, $T = 0.08$, $m = 32$	118
6.10	Log-Log Plot vs $t(\text{sec})$ for Example 9, $T = 0.08$, $m = 32$	119
7.1	Results for Example 10, 100 trials.	160
7.2	Log-Log Plot for Example 11.	178
7.3	Log-Log Plot vs $t(\text{sec})$ for Example 11.	179
7.4	Log-Log Plot for Example 12.	181
7.5	Log-Log Plot vs $t(\text{sec})$ for Example 12.	182

LIST OF TABLES

2.1	Results for Example 1.	16
3.1	Discrepancy Comparison, $d =$ dimensions.	19
5.1	Results for Example 2, 75 trials.	47
5.2	Example 3, Crude Monte Carlo.	63
5.3	Example 3, Weighted Uniform Sampling.	63
5.4	Example 3, Importance Sampling with the Rejection Method. . . .	64
5.5	Example 3, Importance Sampling with Smoothed Rejection.	64
5.6	Example 4, Crude Monte Carlo.	65
5.7	Example 4, Weighted Uniform Sampling.	65
5.8	Example 4, Importance Sampling with the Rejection Method. . . .	66
5.9	Example 4, Importance Sampling with Smoothed Rejection.	66
5.10	Example 5, Crude Monte Carlo.	67
5.11	Example 5, Weighted Uniform Sampling.	67
5.12	Example 5, Importance Sampling with the Rejection Method. . . .	68
5.13	Example 5, Importance Sampling with Smoothed Rejection.	68
6.1	Results for Example 7, 75 trials	82
6.2	Results for Example 8, 75 trials	84

6.3	Importance of Dimensions, Standard discretization	104
6.4	Importance of Dimensions, Comparison for $m = 8$	108
6.5	Results for Example 9, 75 trials	113
7.1	Results for Example 11, 70 trials.	177
7.2	Results for Example 12, 60 trials.	180

ACKNOWLEDGMENTS

I would like to thank Russel Caffisch for his encouragement and help throughout my work on this dissertation.

This work has been supported by grant #AFOSR 89-0003 from the Air Force Office of Scientific Research.

VITA

February 17, 1966	Born, Flushing, New York.
1984	Valedictorian, Plainview-Old Bethpage High School, New York.
1988	B.S., Applied Mathematics Brown University Providence, Rhode Island.
1990	M.S., Mathematics (Applied) University of California, Los Angeles Los Angeles, California.

PRESENTATIONS

August 6, 1993 *Application of Quasi-Random Sequences to Green's Function Monte Carlo*, Presented at the Society for Industrial and Applied Mathematics, Conference on Simulation and Monte Carlo Methods.

ABSTRACT OF THE DISSERTATION

Application of Quasi-Random Sequences to Monte Carlo Methods

by

Bradley Scott Moskowitz

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 1993

Professor Russel Caflisch, Chair

I have examined the reasons that quasi-random sequences sometimes fail to improve Monte Carlo estimates, which include discontinuities in the integrand and high numbers of dimensions. With these factors in mind, I have found ways to successfully apply quasi-random sequences to several model problems for which initial attempts to apply them were met with failure. In doing so, I have obtained accelerated convergence rates for these examples and improved accuracy in a given amount of cpu time. These model problems can serve as a guide to the successful application of quasi-random sequences to a variety of practical problems in order to obtain similar improvements.

Unlike pseudorandom sequences, quasi-random sequences are deliberately not intended to behave like independent random uniformly distributed points. Instead,

points in a quasi-random sequences are allowed to be correlated in such a way as to improve the accuracy of Monte Carlo estimates. Such estimates have been successfully applied to several problems in physics, chemistry, and reliability analysis, among others.

For a much larger set of problems though, quasi-random sequences have not been as successful, and their use has not become widespread. However, there are a great many problems for which quasi-random Monte Carlo estimates could indeed provide significant improvements over pseudorandom estimates, as long as one is careful about the particular algorithms used and the way in which the quasi-random sequences are applied.

In the case of variance reduction, I found that the use of weighted uniform sampling instead of importance sampling can improve quasi-random results significantly.

For applications involving iterations or successive time steps, such as Diffusion Monte Carlo, which also involves very high dimensional integrals, continuation methods for combining pseudorandom and quasi-random sequences were devised and successfully applied to a few simple quantum mechanical systems.

Both of these cases, and a few similar examples, provide evidence that quasi-random sequences could be useful for a far wider range of applications than perhaps previously believed.

Part I

Monte Carlo Integration and Quasi-Random Sequences

CHAPTER 1

Introduction

Monte Carlo methods are statistically based methods in which solutions are obtained by accumulating a large number of samples or trial runs and then averaging the results. Although the theory is based on using purely independent random points to generate samples, in practice sequences of points called pseudorandom are used. These are designed to perform as closely as possible to uniformly distributed independent random points while also being repeatable and quickly produced on a computer. There are several tests for randomness which pseudorandom sequences are designed to pass, including Chi-Square Tests, the Kolmogorov-Smirnov Test, and the Spectral Test. Good pseudorandom generators, such as ULTRA by Zaman and Marsaglia, are readily available. Monte Carlo methods using sequences of random or pseudorandom points have an inherent statistical convergence rate of order $O(N^{-1/2})$, where N is the number of points used, or equivalently the length of the sequence of points.

In stark contrast with pseudorandom sequences are another variety of sequences commonly known as quasi-random. Unlike pseudorandom sequences, quasi-random sequences are **not** intended to act like independent random points. They badly fail most statistical tests for randomness since there are large corre-

lations between the points in a quasi-random sequence. These correlations are a result of the design of quasi-random sequences, and they force the points of these sequences to spread out more uniformly than pseudorandom or random points over the region of interest by eliminating the “clumping” which is characteristic of pseudorandom and random points. On the other hand, quasi-random sequences are designed to be distributed according to the same uniform distribution as pseudorandom points. This allows them to be used in place of pseudorandom points for several Monte Carlo methods, without introducing any systematic error and in some cases improving the convergence to rates unobtainable using pseudorandom or random points.

Using quasi-random sequences, one can theoretically improve convergence rates to as much as order $O(N^{-1}(\log N)^d)$, in d dimensions. This can be a substantial improvement. For simple low dimensional integrations, such improvements are clearly observed on the computer. However, in many more complicated problems, if one simply substitutes quasi-random points for pseudorandom points in a particular Monte Carlo simulation, the results often show no improvement or worse. There are several reasons for this, which will be a major topic of this paper. The result of such difficulties has been a general lack of faith in the usefulness of quasi-random sequences, and a reluctance to use them in many situations where a few minor alterations to existing algorithms would allow for significant gains in convergence rates and hence reduced computer time usage.

For this thesis, I have studied quasi-random sequences and identified the reasons

for their failure to improve convergence rates in various applications. Subsequently, I have used this knowledge to modify traditional Monte Carlo algorithms in subtle yet important ways so that with quasi-random sequences the kind of improvement in convergence rate that is hoped for is attained. In many cases, the resulting acceleration of convergence rates and reduced computer time more than makes up for the small cost of the modifications involved and a faster method is the end product.

CHAPTER 2

Monte Carlo Integration

Integration, and especially multidimensional integration, is often the setting in which Monte Carlo methods are employed. In fact, a great many Monte Carlo methods can be accurately thought of as being nothing more or less than estimating integrals. The underlying theory of such estimates is straightforward and will be briefly reviewed.

2.1 Statistical Basics

Given random vector $\mathbf{x} \in \mathbb{R}^d$ with probability density function $g(\mathbf{x})$ over region $D \subset \mathbb{R}^d$ and scalar function $f(\mathbf{x})$ defined on D , we define the following statistical properties of f :

$$\text{Expected Value: } E_g f(\mathbf{x}) = \int_D f(\mathbf{x})g(\mathbf{x})d\mathbf{x} \quad (2.1)$$

$$\text{Variance: } var_g(f(\mathbf{x})) = \int_D (f(\mathbf{x}) - E_g f(\mathbf{x}))^2 g(\mathbf{x})d\mathbf{x} \quad (2.2)$$

$$\begin{aligned} &= E_g[(f(\mathbf{x}) - E_g f(\mathbf{x}))^2] \\ &= E_g[(f(\mathbf{x}))^2 - 2f(\mathbf{x})E_g f(\mathbf{x}) + (E_g f(\mathbf{x}))^2] \\ &= E_g[(f(\mathbf{x}))^2] - (E_g f(\mathbf{x}))^2 \end{aligned} \quad (2.3)$$

$$\text{Standard Deviation: } \sigma_g(f(\mathbf{x})) = \sqrt{\text{var}_g(f(\mathbf{x}))} \quad (2.4)$$

Say A is some quantity estimated by $\hat{A}(\mathbf{y})$, a function of a set of N random samples, $\mathbf{y} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$, with joint probability density function $g(\mathbf{y})$. Let J denote the domain of \mathbf{y} in \mathbb{R}^{Nd} . Then, the following quantities characterize the statistical quality of the estimate:

$$\begin{aligned} \text{Mean Square Error: } mse(\hat{A}) &= E_g((\hat{A} - A)^2) \\ &= \int_J (\hat{A}(\mathbf{y}) - A)^2 g(\mathbf{y}) d\mathbf{y} \end{aligned} \quad (2.5)$$

$$\text{Bias: } bias(\hat{A}) = E_g \hat{A} - A \quad (2.6)$$

$$\begin{aligned} \text{Variance: } var(\hat{A}) &= E_g((\hat{A} - E_g \hat{A})^2) \\ &= \int_J (\hat{A}(\mathbf{y}) - E_g \hat{A})^2 g(\mathbf{y}) d\mathbf{y} \end{aligned} \quad (2.7)$$

These quantities are simply related as demonstrated below:

$$\begin{aligned} mse(\hat{A}) &= \int_J (\hat{A}(\mathbf{y}) - A)^2 g(\mathbf{y}) d\mathbf{y} \\ &= \int_J (\hat{A}(\mathbf{y}) - E\hat{A} - A + E\hat{A})^2 g(\mathbf{y}) d\mathbf{y} \\ &= \int_J [(\hat{A}(\mathbf{y}) - E\hat{A})^2 + (A - E\hat{A})^2 - 2(\hat{A}(\mathbf{y}) - E\hat{A})(A - E\hat{A})] g(\mathbf{y}) d\mathbf{y} \\ &= var(\hat{A}) + (bias(\hat{A}))^2 - 2 \int_J (\hat{A}(\mathbf{y}) - E\hat{A})(A - E\hat{A}) g(\mathbf{y}) d\mathbf{y} \\ &= var(\hat{A}) + (bias(\hat{A}))^2 - 2 \int_J (\hat{A}A - \hat{A}E\hat{A} - AE\hat{A} + (E\hat{A})^2) g(\mathbf{y}) d\mathbf{y} \\ &= var(\hat{A}) + (bias(\hat{A}))^2 - 2[AE\hat{A} - (E\hat{A})^2 - AE\hat{A} + (E\hat{A})^2] \\ mse(\hat{A}) &= var(\hat{A}) + (bias(\hat{A}))^2 \end{aligned} \quad (2.8)$$

We will be concerned with estimators of the form \hat{A}_N where as N increases

$\hat{A}_N \rightarrow A$, or more precisely $\lim_{N \rightarrow \infty} mse(\hat{A}_N) = 0$. Such estimates are called **consistent**.

2.2 Statistical Error Measures and Convergence Rates

In comparisons with deterministic error, two useful measures of the statistical error of an estimator are as follows:

$$\begin{aligned} \text{Root Mean Square Error: } rmse(\hat{A}) &= \sqrt{mse(\hat{A})} \\ &= \sqrt{\int_J (\hat{A}(\mathbf{y}) - A)^2 g(\mathbf{y}) d\mathbf{y}} \quad (2.9) \end{aligned}$$

$$\begin{aligned} \text{Mean Absolute Error: } mae(\hat{A}) &= E(|\hat{A} - A|) \\ &= \int_J |\hat{A}(\mathbf{y}) - A| g(\mathbf{y}) d\mathbf{y} \quad (2.10) \end{aligned}$$

If the error is a fixed non-random amount, ϵ , then we have $rmse = \sqrt{\epsilon^2} = |\epsilon|$ and $mae = |\epsilon|$. On the other hand, when the error is a random quantity, either of the above quantities serves as a measure of its average magnitude over repeated trials, and hence a measure of the quality of the estimator in general. The root mean square error is generally preferred because of some nice statistical properties that it has. Also, the mean absolute error is always smaller, by Jensen's Inequality which states: $E(|x|) \leq \sqrt{E(x^2)}$ for any random variable. I will use root mean square error throughout this dissertation. Note that for unbiased estimators, the root mean square error is exactly equal to the standard deviation of the estimator.

In the case of an estimator of the form \hat{A}_N with N increasing, one says that

the estimator has a statistical convergence rate of order $O(N^{-a})$ if the following is satisfied:

$$\lim_{N \rightarrow \infty} \frac{rmse(A_N)}{N^a} = \text{const} < \infty$$

2.2.1 Expectation and Variance of Sums

Before going on to consider Monte Carlo estimation, it is useful to derive the expectation and variance of sums of random variables. Let \mathbf{x}_1 and \mathbf{x}_2 be random variables over $D_1 \times D_2$ with joint probability density function $g(\mathbf{x}_1, \mathbf{x}_2)$, and marginal density functions $g_1(\mathbf{x}_1) = \int_{D_2} g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2$ and $g_2(\mathbf{x}_2) = \int_{D_1} g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1$. Then,

$$\begin{aligned} E_g(a\mathbf{x}_1 + b\mathbf{x}_2) &= \int_{D_2} \int_{D_1} (a\mathbf{x}_1 + b\mathbf{x}_2) g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \\ &= \int_{D_1} a\mathbf{x}_1 \left(\int_{D_2} g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 \right) d\mathbf{x}_1 + \int_{D_2} b\mathbf{x}_2 \left(\int_{D_1} g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 \right) d\mathbf{x}_2 \\ &= \int_{D_1} a\mathbf{x}_1 g_1(\mathbf{x}_1) d\mathbf{x}_1 + \int_{D_2} b\mathbf{x}_2 g_2(\mathbf{x}_2) d\mathbf{x}_2 \\ &= aE_{g_1}\mathbf{x}_1 + bE_{g_2}\mathbf{x}_2 \end{aligned} \tag{2.11}$$

$$\begin{aligned} var_g(a\mathbf{x}_1 + b\mathbf{x}_2) &= \int_{D_2} \int_{D_1} \left((a\mathbf{x}_1 + b\mathbf{x}_2) - (aE_g\mathbf{x}_1 + bE_g\mathbf{x}_2) \right)^2 g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \\ &= \int_{D_2} \int_{D_1} \left(a(\mathbf{x}_1 - E_g\mathbf{x}_1) + b(\mathbf{x}_2 - E_g\mathbf{x}_2) \right)^2 g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \\ &= a^2 \int_{D_1} (\mathbf{x}_1 - E_g\mathbf{x}_1)^2 \left(\int_{D_2} g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 \right) d\mathbf{x}_1 \\ &\quad + b^2 \int_{D_2} (\mathbf{x}_2 - E_g\mathbf{x}_2)^2 \left(\int_{D_1} g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 \right) d\mathbf{x}_2 \\ &\quad + 2ab \int_{D_2} \int_{D_1} (\mathbf{x}_1 - E_g\mathbf{x}_1)(\mathbf{x}_2 - E_g\mathbf{x}_2) g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \\ &= a^2 \int_{D_1} (\mathbf{x}_1 - E_g\mathbf{x}_1)^2 g_1(\mathbf{x}_1) d\mathbf{x}_1 \end{aligned}$$

$$\begin{aligned}
& + b^2 \int_{D_2} (\mathbf{x}_2 - E_g \mathbf{x}_2)^2 g_2(\mathbf{x}_2) d\mathbf{x}_2 \\
& + 2ab \int_{D_2} \int_{D_1} (\mathbf{x}_1 - E_g \mathbf{x}_1)(\mathbf{x}_2 - E_g \mathbf{x}_2) g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \\
& = a^2 \text{var}_{g_1}(\mathbf{x}_1) + b^2 \text{var}_{g_2}(\mathbf{x}_2) + 2ab \text{cov}_g(\mathbf{x}_1, \mathbf{x}_2)
\end{aligned} \tag{2.12}$$

Note: $\text{cov}_g(\mathbf{x}_1, \mathbf{x}_2) = \int_{D_2} \int_{D_1} (\mathbf{x}_1 - E_g \mathbf{x}_1)(\mathbf{x}_2 - E_g \mathbf{x}_2) g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2$.

2.3 Crude Monte Carlo Estimation

Let $A = \int_D f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}$ be an unknown quantity to be estimated, with $D \subset \mathbb{R}^d$, g strictly positive, and $\eta = \int_D g(\mathbf{x}) d\mathbf{x}$, the known normalization constant for $g(\mathbf{x})$. (Note: Often $D = [0, 1]^d$ which is denoted I^d , and $g(\mathbf{x}) = 1$ for all \mathbf{x} , which is represented by $\mathbf{x} \in U([0, 1]^d)$. Then $\eta = 1$.)

Take \mathbf{x}_1 to be distributed according to $g(\mathbf{x})$, i.e. $\frac{1}{\eta} g(\mathbf{x})$ is the probability density function for \mathbf{x}_1 . Let $\hat{A}_1 = \eta f(\mathbf{x}_1)$. Then,

$$\begin{aligned}
E(\hat{A}_1) &= \int_D \eta f(\mathbf{x}) \frac{1}{\eta} g(\mathbf{x}) d\mathbf{x} = A \\
\text{var}(\hat{A}_1) &= \eta^2 \text{var}(f)
\end{aligned}$$

Therefore \hat{A}_1 is an unbiased estimator for A , with $\sigma(\hat{A}_1) = \eta \sigma(f)$.

For simplicity, I will assume for the rest of this section that $g(\mathbf{x})$ is normalized so that $\eta = 1$. There is no loss in generality since η could easily be reinserted. This gives us $\hat{A}_1 = f(\mathbf{x}_1)$ with zero bias and standard deviation $\sigma_g(f)$.

This estimate is improved by using an independent and identically distributed (i.i.d.) sample of points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ each from $g(\mathbf{x})$.

Let, the Crude Monte Carlo estimate be defined as follows:

$$\hat{A}_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)$$

Then,

$$E(\hat{A}_N) = E\left(\frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)\right) \quad (2.13)$$

$$= \frac{1}{N} \sum_{i=1}^N E_g(f(\mathbf{x}_i)) \quad , \quad \text{using Eqn 2.11}$$

$$= \frac{1}{N} \cdot N \cdot A = A \quad (2.14)$$

$$\begin{aligned} \text{var}(\hat{A}_N) &= \text{var}\left(\frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)\right) \\ &= \frac{1}{N^2} \sum_{i=1}^N \text{var}_g(f(\mathbf{x}_i)) \quad , \quad \text{using Eqn 2.12} \quad , \quad \text{cov}(\mathbf{x}_i, \mathbf{x}_j) = 0, \forall i \neq j \\ &= \frac{1}{N^2} \cdot N \cdot \text{var}_g(f) = \frac{1}{N} \text{var}_g(f) \end{aligned} \quad (2.15)$$

We see then that \hat{A}_N is an unbiased estimator of A with $\sigma(\hat{A}_N) = \frac{1}{\sqrt{N}} \sigma_g(f)$. Therefore $\text{rsmc}(\hat{A}_N) = \frac{1}{\sqrt{N}} \sigma_g(f)$. Then, as long as $\text{var}_g(f)$ is finite, which we will assume, the Crude Monte Carlo estimate is consistent, with a convergence rate of order $O(N^{-1/2})$. If the variance is infinite, it is often possible to use variance reduction methods such as those discussed in Chapter 4 to make it finite. The above result is the basis for the statement that Monte Carlo methods have a statistical convergence rate of order $O(N^{-1/2})$.

2.4 Simple Integration Example

Before studying quasi-random sequences in depth in the next chapter, here is an eight dimensional integration example which will demonstrate the significant gains which are made possible by utilizing a quasi-random sequence of points in place of a pseudorandom sequence. This example will also introduce the methods of data analysis which will be used throughout this study. The pseudorandom points are taken from a non-linear additive feedback generator in the standard C library, which is known to pass many of the standard statistical tests for random-like behavior, while the quasi-random points are from a particular quasi-random sequence known as the Halton Sequence, which will be introduced in Chapter 3.

2.4.1 Discussion of Computations

Let $\mathbf{x}_i = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ represent a single d -dimensional pseudorandom or quasi-random point. Also let $\mathbf{y} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) \in \mathbb{R}^{Nd}$ represent a full sample of N points (i.e. a particular subsequence of length N from either a pseudorandom or quasi-random sequence.). Then, the Crude Monte Carlo estimate of the integral of a function $f(\mathbf{x})$ can be written as follows: $\hat{A}_N(\mathbf{y}) = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)$.

In the example, this estimate is recomputed M times using a new sample each time in order to approximate the root mean square error (*rmse*) of the estimate

as follows:

$$\begin{aligned} rmse(\hat{A}_N) &= \sqrt{E((\hat{A}_N(\mathbf{y}) - A)^2)} \\ &\approx \sqrt{\frac{1}{M} \sum_{k=1}^M (\hat{A}_N(\mathbf{y}_k) - A)^2} \end{aligned}$$

where A is the exact value of the integral being estimated, $\{\mathbf{y}_k\}_{k=1}^M$ is a set of M pseudorandom or quasi-random samples, and $\hat{A}_N(\mathbf{y}_k)$ are the associated estimates.

Notice that knowledge of the exact solution is used here. In general, the exact solution is (obviously) unknown. Therefore, another way to approximate the *rmse* is needed. To do this it is assumed that each estimate is independent. This is a fair assumption when using pseudorandom sequences. However, it could be questionable when using quasi-random sequences. For the examples I have studied this assumption has not given me any problems, but it certainly is a matter which could be studied further. Another assumption is that the estimate \hat{A}_N is unbiased, or at least that the bias is an order of magnitude lower than any statistical errors. This will be true for all of the examples. The *rmse* is then estimated by computing the sample variance of the M values of $\hat{A}_N(\mathbf{y})$ and then taking the square root. This can be written as follows:

$$rmse(\hat{A}_N) \approx \sqrt{\frac{1}{M-1} \sum_{k=1}^M \left(\hat{A}_N(\mathbf{y}_k) - \overline{\hat{A}_N} \right)^2}$$

where $\overline{\hat{A}_N} = \frac{1}{M} \sum_{k=1}^M \hat{A}_N(\mathbf{y}_k)$.

This sample variance will be a good estimate as long as M is sufficiently high and the assumptions above hold.

Once $rmse(\hat{A}_N)$ has been approximated for several different values of N using either method above, the convergence rate can then be approximated by using the following linear relationship:

$$rmse(\hat{A}_N) = cN^a \quad (2.16)$$

$$\Rightarrow \log(rmse(\hat{A}_N)) = \log(cN^a)$$

$$\Rightarrow \log(rmse(\hat{A}_N)) = \log(c) + a \log(N) \quad (2.17)$$

Now we can find the approximate convergence rate by computing the least squares line for $\log(rmse(\hat{A}_N))$ against $\log(N)$ and taking its slope a . The convergence rate is then of order $O(N^a)$, where a should be less than zero. For pseudorandom sequences, we expect a to be near $-\frac{1}{2}$, while for quasi-random sequences the expectation is that a lower value of a , near -1 , will be observed. The same computations using $\log(rmse(\hat{A}_N))$ against the logarithm of t_N , the average cpu time to compute \hat{A}_N , allow estimates of convergence relative to cpu time to be made. This is done to provide a more fair comparison by taking into account any extra computation time needed when using quasi-random sequences.

Example 1 *The following integral:*

$$I = \int_{I^8} \sin(x_1 + x_2 + \cdots + x_8) dx_1 \dots dx_8$$

is approximated using crude Monte Carlo estimates of the form $\hat{A}_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)$ where each $\mathbf{x}_i = (x_1, \dots, x_8)$ is an 8 dimensional point from either a pseudorandom

or Halton quasi-random sequence, and $f(\mathbf{x}) = \sin(x_1 + x_2 + \cdots + x_8)$. Defining $\mathbf{y} = (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \mathbb{R}^{8N}$ as a full sample of N points and $\hat{A}_N(\mathbf{y})$ as the associated estimate, the root mean square error of \hat{A}_N is approximated by the following:

$$rmse(\hat{A}_N) \approx \sqrt{\frac{1}{M} \sum_{k=1}^M (\hat{A}_N(\mathbf{y}_k) - A)^2}$$

using the exact solution $A = \frac{75}{288}$, and $M = 75$ trials.

Some numerical results are given in Table 2.1, which shows the computed *rmse* and average *cpu* times for selected values of N , as well as the convergence rate estimates. Figure 2.1 contains plots of $\log(rmse)$ versus $\log(N)$ and $\log(rmse)$ versus $\log(t_N)$ on which one can graphically observe differences in convergence rates by comparing slopes. As expected, the quasi-random sequences provide a significant improvement for this example, as is evident from both the table and plots.

An examination of the results for Example 1 in Table 2.1 and Figure 2.1 indicates that while the quasi-random points take longer to compute, the improved convergence more than makes up for the difference. This is typical of a successful application of quasi-random points to a Monte Carlo method. There is some overhead and extra computation time involved, but the gains in convergence will more than make up for it. Of course, this is not always the case, especially when dealing with Monte Carlo methods which produce discontinuous or very high dimensional integrands, as will be discussed in Chapters 5 and 6. However, I intend to demonstrate that in several important applications improved convergence rates using quasi-random sequences are possible and allow one to obtain more accurate

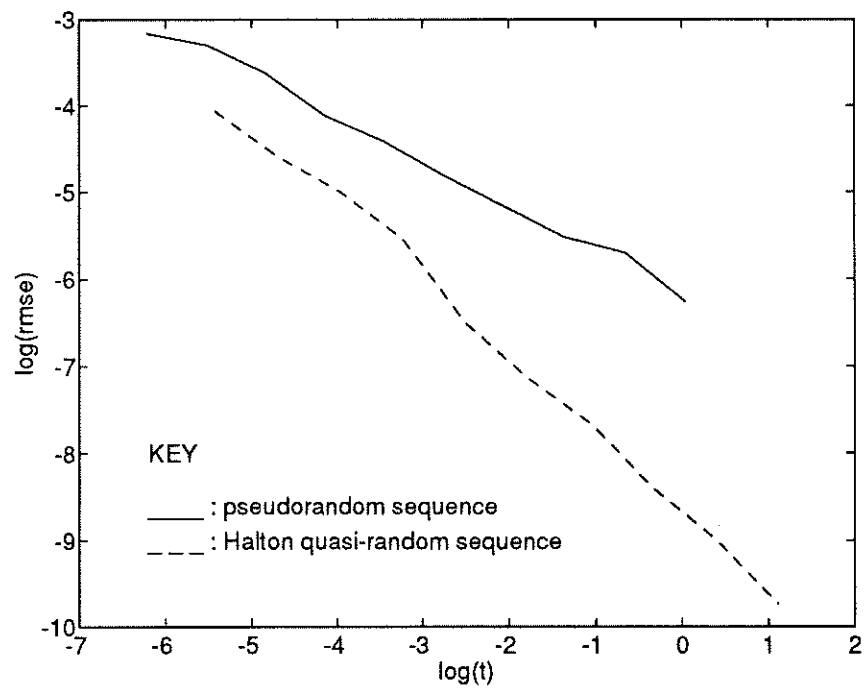
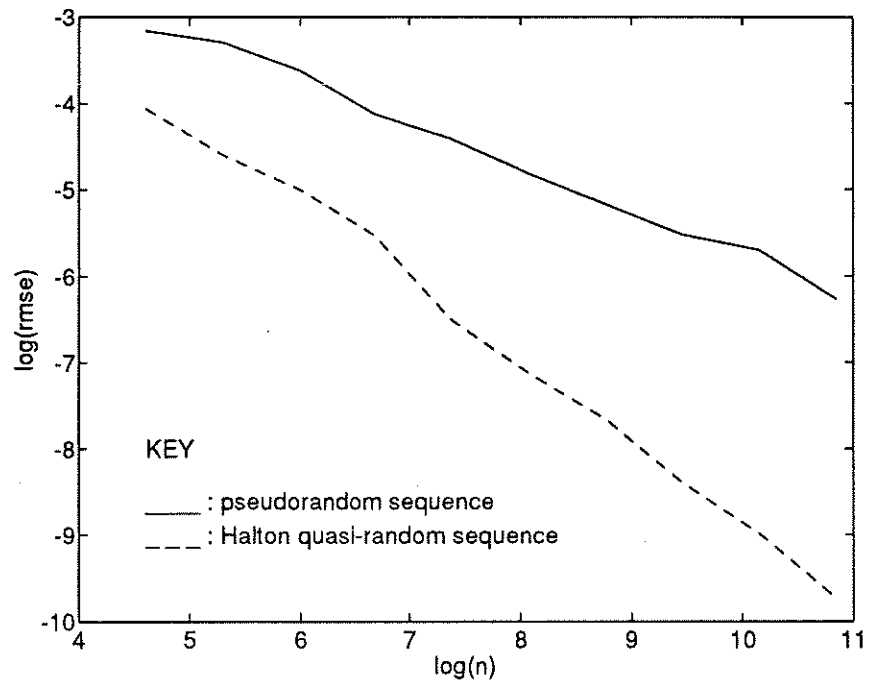


Figure 2.1: Log-Log Plots for Example 1.

Integration Example, Crude Monte Carlo

N	Pseudorandom		Halton Quasi-Random	
	$rmse(\hat{A}_N)$	t_N	$rmse(\hat{A}_N)$	t_N
200	0.037070	0.00400	0.010109	0.00920
800	0.016191	0.01600	0.003970	0.03933
3200	0.008167	0.06400	0.000806	0.16560
12800	0.004000	0.25773	0.000226	0.70733
51200	0.001907	1.03853	0.000059	3.02893

Root Mean Square Error, $rmse$, Convergence Rates

	Pseudorandom		Halton Quasi-Random	
	vs N	vs t_N	vs N	vs t_N
	-0.505	-0.504	-0.925	-0.884

Table 2.1: Results for Example 1.

results in a shorter amount of time, as they do here in this simple example.

CHAPTER 3

Quasi-Random Sequences for Monte Carlo Integration

3.1 Discrepancy

Quasi-random sequences are specifically designed to spread out more evenly than random or pseudorandom sequences. A measure of this property, developed by number theorists is known as **discrepancy** (see Niederreiter [42]). In fact, an alternate name for quasi-random sequences is low-discrepancy sequences. In general terms, the lower the discrepancy of a sequence, the closer it is to being perfectly uniformly distributed.

One might think that independent random points from a uniform distribution would be ideal, but while they are not bad, there are better sequences in terms of discrepancy. The problem with random or pseudorandom sequences is that they are ‘random,’ so while one expects them to spread out uniformly, they will not always do so in an efficient way. One might next think that points on an equally spaced grid would obviously be best. In one dimension this is correct, but in more than one dimension, and increasingly so as the number of dimensions increases, such grid points are actually quite poor. A simple argument is that, for example

in three dimensions, among N grid points only $N^{1/3}$ unique values along each coordinate are used.

As discrepancy measurements will show, for a moderate number of dimensions, d , points from quasi-random sequences spread out more uniformly than either pseudorandom points or equally spaced grid points. This will turn out to allow Monte Carlo methods using quasi-random sequences to achieve faster convergence rates.

Definition 1 (Local Discrepancy) *Given a sequence of points, $\{x_i\}_{i=1}^N$, in I^d , and a subset $J = \prod_{i=1}^d [0, j_i) \subset I^d$, the **local discrepancy over J** is defined by*

$$D(J) = \left| \frac{A(J)}{N} - \text{vol}(J) \right|$$

where $A(J) = \#$ of points contained in J

Definition 2 (Discrepancy) *Given a sequence of points, $\{x_i\}_{i=1}^N$, in I^d , the **discrepancy, D** , is given by the maximum local discrepancy:*

$$D = \sup_{J=\prod [0, j_i)} D(J)$$

A classical result of Roth gives a lower bound on D as $D \geq O(N^{-1}(\log N)^{d/2})$ for any sequence in I^d as $N \rightarrow \infty$. However, the best quasi-random sequences found to date have discrepancies of order $O(N^{-1}(\log N)^d)$. This is an area of active research (see for example Niederreiter [44]). On the other hand, for pseudorandom

sequence type	asymptotic discrepancy	observation
pseudo-random	$O(N^{-1/2})$	independent of dimension
quasi-random	$O(N^{-1}(\log N)^d)$	weak dimensional dependence
grid-based	$O(N^{-1/d})$	strong dimensional dependence

Table 3.1: Discrepancy Comparison, d = dimensions.

sequences the discrepancy is of order $O(N^{-1/2})$, while for grid-based points (e.g. $\mathbf{x}_{j+k*N_x} = (\frac{j}{N_x}, \frac{k}{N_x})$) the discrepancy is of order $O(N^{-1/d})$. Table 3.1 summarizes these results.

For more information about the computation of discrepancies, see Halton [21] for the Halton sequence, Niederreiter [42, pg.1018] for a typical pseudorandom sequence calculation, and Fox [13] for the equally spaced grid.

3.2 Koksma-Hlawka Inequality and Convergence

The usefulness of discrepancy comes from the following theorem which states that whenever a sequence of points is used to approximate a definite integral by averaging function values (such as in Crude Monte Carlo), the absolute value of the error can be bounded from above by a product of the discrepancy of the sequence and a constant depending on the integrand known as its variation. This theorem along with the discrepancy results from the previous section provide proof that

Monte Carlo integration using a quasi-random sequence of points does in fact converge to the correct result, and importantly that the convergence rate is of the order $O(N^{-1}(\log N)^d)$ for large N .

First we must define the variation of a function. There are several different measures of variation which are useful for various purposes. The one we require is known as the Hardy-Krause variation. It is defined as follows:

Definition 3 (Hardy-Krause Variation) *For function f on I^d and subinterval*

$J = [a_1^{(1)}, a_2^{(1)}] \times [a_1^{(2)}, a_2^{(2)}] \times \cdots \times [a_1^{(d)}, a_2^{(d)}] \subseteq I^d$, *let,*

$$\Delta(f; J) = \sum_{\epsilon_1=1}^2 \sum_{\epsilon_2=1}^2 \cdots \sum_{\epsilon_d=1}^2 (-1)^{\epsilon_1 + \cdots + \epsilon_d} f(a_{\epsilon_1}^{(1)}, a_{\epsilon_2}^{(2)}, \dots, a_{\epsilon_d}^{(d)})$$

In order to define a partition \mathcal{P} of I^d , first select a set of d finite sequences with points in $[0, 1]$:

$$0 = \nu_0^{(j)} < \nu_1^{(j)} < \cdots < \nu_{m_j}^{(j)} = 1 \quad , \quad (j = 1, 2, \dots, d)$$

Then, the partition consists of the following subintervals:

$$[\nu_{i_1}^{(1)}, \nu_{i_1+1}^{(1)}] \times [\nu_{i_2}^{(2)}, \nu_{i_2+1}^{(2)}] \times \cdots \times [\nu_{i_d}^{(d)}, \nu_{i_d+1}^{(d)}]$$

over all sets of i_j with $0 \leq i_j < m_j$, ($j = 1, 2, \dots, d$).

Now, define the Vitali variation of a function, f , in d -dimensions as follows:

$$V_v^{(d)}(f) = \sup_{\mathcal{P}} \sum_{J \in \mathcal{P}} |\Delta(f; J)|$$

Let $V_v^{(k)}(f; i_1, \dots, i_k)$ denote the Vitali variation in k -dimensions of the function f restricted to the k -dimensional subspace $(x_{i_1}, x_{i_2}, \dots, x_{i_k})$ with all the unused

components set equal to 1. For example, if f is five dimensional we might have $V_v^{(3)}(f; 2, 4, 5)$, which would equal $V_v^{(3)}(f(1, x_2, 1, x_4, x_5))$.

Finally, the **Hardy-Krause Variation** is defined as follows:

$$V(f) = \sum_{k=1}^d \sum_{1 \leq i_1 < \dots < i_k \leq d} V_v^{(k)}(f; i_1, i_2, \dots, i_k)$$

Functions for which the Hardy-Krause variation is finite are said to be of bounded variation (in the Hardy-Krause sense).

Now the Koksma-Hlawka Inequality can be stated as follows:

Theorem 1 (Koksma-Hlawka Inequality) *For any sequence, $\{x_i\}_{i=1}^n$, in I^d and any function, f , of bounded variation (in the Hardy-Krause sense), over I^d ,*

$$\left| \frac{1}{n} \sum_{i=1}^n f(x_i) - \int_{I^d} f(x) dx \right| \leq V(f)D$$

where $V(f)$ is the Hardy-Krause variation of f .

The proof of this theorem is omitted but can be found for example in Kuipers and Niederreiter [35] or Keng [39].

Using the Koksma-Hlawka inequality and the asymptotic discrepancy results in Table 3.1, one gets the following error bounds:

$$\text{pseudorandom : } |err| \leq O(N^{-1/2}) \quad (3.1)$$

$$\text{grid : } |err| \leq O(N^{-1/d}) \quad (3.2)$$

$$\text{quasi-random : } |err| \leq O(N^{-1} \log(N)^d) \quad (3.3)$$

Note that the errors considered here are definite errors of deterministic sequences rather than statistical errors, although the bounds are only asymptotic

in nature. Therefore, in computational studies, I will be using repeated trial runs with new quasi-random or pseudorandom sequences for each trial to estimate the root mean square errors of Monte Carlo estimates, which encompasses both deterministic errors and statistical errors. This is discussed further in Example 1. The convergence rate for pseudorandom sequences here corresponds well with the expected statistical convergence rate. In the case of truly random sequences of points, the discrepancy is itself a random quantity, but its expected value is near order $O(N^{-1/2})$, which is what one would expect (Niederreiter [42]).

The results above provide evidence of the relative merits of pseudorandom, quasi-random, and grid-based sequences for integration. For very high dimensional problems, pseudorandom sequences appear to be the best choice, while for low dimensional problems, grid-based points – in the traditional form of finite difference methods or Newton-Cotes integration formulas are best. However, in the intermediate range of roughly 4 to 30 dimensions, quasi-random sequences, *when used properly*, are often the best choice, which is of course one of the things which I am setting out to demonstrate.

One should not however place too much importance on the actual bounds themselves, since they are not sharp and are also only asymptotically correct. In many cases there is no good rule as to how high N must be before the asymptotic results dominate, and often it is simply a matter of observing computational results to decide whether or not N is high enough. For example, William Morokoff (Morokoff [41]) found evidence that as d becomes higher, the value of N at

which quasi-random sequences start to exhibit their asymptotic behavior becomes exponentially higher.

On the other hand, discrepancy is important in that it does provide a theoretical justification for the application of quasi-random sequences to Monte Carlo methods through the Koksma-Hlawka Inequality. In addition, the Koksma-Hlawka Inequality provides clues to the most important factors affecting the success of quasi-random sequences. As will be discussed, these are the continuity of the integrand and the dimensionality of the integrand.

Most discontinuities will have an adverse effect on the success of quasi-random sequences. This is suggested by the Koksma-Hlawka Inequality since its hypothesis of bounded variation fails for discontinuous integrands in more than one dimension except in special cases.

High dimensions will also adversely affect the success of quasi-random sequences, as is suggested by the $(\log N)^d$ factor in their discrepancy bounds. Computational results in Chapters 5 and 6 will confirm the importance of these two factors, and this knowledge will serve as a guide to obtaining improved results with quasi-random sequences throughout this thesis.

3.3 The Halton Sequence

The Halton sequence is a particular quasi-random sequence with a relatively simple construction. I will be using the Halton sequence whenever a quasi-random

sequence is needed. For the Halton sequence, as for other quasi-random sequences, in d dimensions the discrepancy is of the order $O(N^{-1}(\log N)^d)$.

The Halton sequence is constructed as follows :

To compute the particular Halton point, \mathbf{x}_k , in d dimensions,

1. Expand k in powers of each of the first d primes,

i.e. find $a_k^{(i,j)}$, $i = 1, \dots, d$, $j = 0, \dots$ such that

$$a_k^{(1,0)} + a_k^{(1,1)}2 + a_k^{(1,2)}2^2 + \dots = k$$

$$a_k^{(2,0)} + a_k^{(2,1)}3 + a_k^{(2,2)}3^2 + \dots = k$$

$$\vdots$$

$$a_k^{(d,0)} + a_k^{(d,1)}p_d + a_k^{(d,2)}p_d^2 + \dots = k$$

where p_d is the d th lowest prime.

2. Invert the expansions, defining the components of \mathbf{x}_k by

$$x_k^{(1)} = a_k^{(1,0)}\frac{1}{2} + a_k^{(1,1)}\frac{1}{2^2} + \dots$$

$$x_k^{(2)} = a_k^{(2,0)}\frac{1}{3} + a_k^{(2,1)}\frac{1}{3^2} + \dots$$

$$\vdots$$

$$x_k^{(d)} = a_k^{(d,0)}p_d^{-1} + a_k^{(d,1)}p_d^{-2} + \dots$$

Thus for example in the 3D case,

$$\mathbf{x}_1 = (1/2, 1/3, 1/5) , \mathbf{x}_2 = (1/4, 2/3, 2/5) , \mathbf{x}_3 = (3/4, 1/9, 3/5) ,$$

$$\mathbf{x}_4 = (1/8, 4/9, 4/5) , \mathbf{x}_5 = (5/8, 7/9, 1/25) , \mathbf{x}_6 = (3/8, 2/9, 6/25) , \dots$$

Normally one starts with $k \gg 1$ to avoid the initial closeness of the high dimensional components to zero.

3.4 Other Quasi-Random Sequences

I have used the Halton sequence as a representative of all the various quasi-random sequences since it is the most straightforward to understand and program. Theoretical results indicate that some of the other quasi-random sequences have lower proportionality constants for large N and d (they all have the same asymptotic order with respect to N), which has led some to prefer these sequences (see for example Fox [13]). However, I have found in most of my work that the Halton sequence performed just as well, if not better, than some of the other quasi-random sequences. Others have reported similar findings (Morokoff [41], Harrison [24]).

For the sake of completeness, I have done some computations using another particular quasi-random sequence known as the Sobol sequence (Sobol' [55]). These results were comparable to those found using the Halton sequence and will not be reported here.

Other important quasi-random sequences include the Faure sequence and a collection of sequences developed by Niederreiter (Niederreiter [43]) which he terms (t, s) -nets.

CHAPTER 4

Variance Reduction Using Importance Functions

Variance reduction refers to a number of techniques which have been developed to improve Monte Carlo methods. Recalling the result from Section 2.3, for Crude Monte Carlo integration of $A = \int_D f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}$ using \mathbf{x} distributed according to $g(\mathbf{x})$, which is assumed normalized, we have the following:

$$\hat{A}_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad , \quad \mathbf{x}_i \in g(\mathbf{x}) \quad (4.1)$$

$$rmse(\hat{A}_N) = \frac{1}{\sqrt{N}} \sigma_g(f) \quad (4.2)$$

Variance reduction methods have the effect of lowering the constant factor $\sigma_g(f)$. While this does not affect the underlying convergence rate, it can greatly improve the results when the constant factor is significantly reduced. These methods are widespread, and indeed Monte Carlo would probably be worthless in most practical applications without some use of variance reduction.

Some of the more common variance reduction techniques include Importance Sampling, Control Variates, Antithetic Variates, and Stratified Sampling. For many of these methods, the use of quasi-random sequences in place of pseudorandom sequences is straightforward. However, in some instances there are significant issues that arise when quasi-random sequences are used which are not present

when pseudorandom sequences are used. An example, which will be studied in some detail is the case of Importance Sampling using the Rejection Method. I will show that a similar, but less well-known method known as Weighted Uniform Sampling is often significantly better than Importance Sampling when quasi-random sequences are used even though both methods are often about equal when pseudorandom sequences are used.

In this Chapter, I will introduce Importance Sampling, the Rejection Method, and Weighted Uniform Sampling from a purely statistical point of view. Later, in Chapter 5, the application of quasi-random sequences will be examined.

4.1 Importance Sampling

Say one wishes to estimate $A = \int_D f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}$, where g is normalized as mentioned above, and also g is strictly positive. Then, when f is single-signed, or nearly single-signed, or the domain, D , can be easily divided up into regions where f is single-signed, importance sampling provides a way to greatly reduce the variance.

Importance Sampling involves finding a positive definite “importance function,” $h(\mathbf{x})$, which mimics the behavior of $|fg|$ over D but is also either integrable analytically or easily integrated numerically to a high degree of accuracy (for example it could be a product of one-dimensional functions). The sampling procedure is then altered to generate points distributed according to $h(\mathbf{x})$, which is normalized,

instead of $g(\mathbf{x})$. Then instead of evaluating $f(\mathbf{x})$ at each sample point, one evaluates $\frac{f(\mathbf{x})g(\mathbf{x})}{h(\mathbf{x})}$. The effect is one of sampling more often where $|fg|$ is high and less often where it is low. This improves the efficiency of the sampling. Also if h mimics $|fg|$ well, then $\frac{fg}{h}$ will be nearly constant modulo sign changes, which reduces the variance significantly when fg is mostly single-signed over D .

The Importance Sampled Monte Carlo estimate can be written as follows:

$$\tilde{A}_N = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i) g(\mathbf{x}_i)}{h(\mathbf{x}_i)} \quad , \quad \mathbf{x}_i \in h(\mathbf{x}) \quad (4.3)$$

Now consider the following computations:

$$\begin{aligned} E(\tilde{A}_N) &= E\left(\frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i) g(\mathbf{x}_i)}{h(\mathbf{x}_i)}\right) \\ &= \frac{1}{N} \sum_{i=1}^N E_h\left(\frac{f(\mathbf{x}_i) g(\mathbf{x}_i)}{h(\mathbf{x}_i)}\right) \quad , \quad \text{using Eqn 2.11} \\ &= \frac{1}{N} N E_h\left(\frac{fg}{h}\right) = \int_D \frac{f(\mathbf{x}) g(\mathbf{x})}{h(\mathbf{x})} h(\mathbf{x}) d\mathbf{x} \quad , \quad \text{using Eqn 2.1} \\ &= A \\ \text{var}(\tilde{A}_N) &= \text{var}\left(\frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i) g(\mathbf{x}_i)}{h(\mathbf{x}_i)}\right) = \frac{1}{N^2} \cdot N \cdot \text{var}_h\left(\frac{fg}{h}\right) \\ &= \frac{1}{N} \text{var}_h\left(\frac{fg}{h}\right) \\ &= \frac{1}{N} \int_D \left(\frac{f(\mathbf{x}) g(\mathbf{x})}{h(\mathbf{x})} - E_h\left(\frac{f(\mathbf{x}) g(\mathbf{x})}{h(\mathbf{x})}\right)\right)^2 h(\mathbf{x}) d\mathbf{x} \end{aligned}$$

This shows that the estimate is unbiased, and that its variance would be zero if h were exactly proportional to $|fg|$ and fg were strictly single-signed. In practice these conditions cannot be met exactly, especially the first, but when they are nearly satisfied the variance is very small.

4.1.1 The Rejection Method

The process of generating sample points distributed according to $h(\mathbf{x})$ can be difficult. In some special cases a direct transformation can be used to convert uniformly distributed points to those distributed like h , or if g is the natural distribution of sample points a direct transformation from points distributed like g to points distributed like h may be possible. However, it is often the case that neither of these transformations is readily available. It can also happen that an exact transformation is known but it is extremely costly to execute on the computer.

A general technique for transforming sample points from $g(\mathbf{x})$ to $h(\mathbf{x})$ where both are normalized is the **Rejection Method**. Here is the basic algorithm for the **Rejection Method**:

Repeat until N points have been accepted:

1. Sample $\mathbf{x}_t \in g(\mathbf{x})$ and $y \in U([0, 1])$.
2. If $y < \frac{1}{\gamma} \frac{h(\mathbf{x}_t)}{g(\mathbf{x}_t)}$, where $\gamma \geq \sup_{\mathbf{x} \in D, g(\mathbf{x}) \neq 0} \frac{h(\mathbf{x})}{g(\mathbf{x})}$, then accept \mathbf{x}_t .
3. Otherwise, reject \mathbf{x}_t .

The justification of this method is demonstrated as follows:

Denote the probability density of points selected in step (1) as $f_{select}(\mathbf{x})$ and the probability of accepted points produced by the Rejection Method as $f_{accept}(\mathbf{x})$.

Then,

$$\begin{aligned}
f_{\text{accept}}(\mathbf{x}) &= \frac{f_{\text{select}}(\mathbf{x}) \text{Prob}\{\text{accept } \mathbf{x}\}}{\int_D f_{\text{select}}(\boldsymbol{\xi}) \text{Prob}\{\text{accept } \boldsymbol{\xi}\} d\boldsymbol{\xi}} \\
&= \frac{g(\mathbf{x}) \cdot \text{Prob}\{y < \frac{1}{\gamma} \frac{h(\mathbf{x})}{g(\mathbf{x})}\}}{\int_D \left[g(\boldsymbol{\xi}) \cdot \text{Prob}\{y < \frac{1}{\gamma} \frac{h(\boldsymbol{\xi})}{g(\boldsymbol{\xi})}\} \right] d\boldsymbol{\xi}} \\
&= \frac{\frac{h(\mathbf{x})}{\gamma}}{\int_D \frac{h(\boldsymbol{\xi})}{\gamma} d\boldsymbol{\xi}} \\
f_{\text{accept}}(\mathbf{x}) &= h(\mathbf{x})
\end{aligned}$$

This shows that the Rejection Method correctly samples the probability density $h(\mathbf{x})$. (Note: When the original samples are uniformly distributed over I^d , the unit cube in d dimensions, then $g(\mathbf{x}) = 1$.)

4.2 Weighted Uniform Sampling

Weighted sampling refers to methods which involve assigning each sample point an associated weight, and then using weighted averages to compute Monte Carlo estimates. These methods can be used in cases with any initial sample distribution $g(\mathbf{x})$, but I will be dealing primarily with the case where the original sample is uniform, in particular with sequences of pseudorandom or quasi-random points, hence the name Weighted Uniform Sampling. This type of method is described in Powell and Swann [46]. They present a convergence proof, but without providing estimates of the bias and variance as I do here. My treatment is based on material in Hansen, Hurwitz, and Madow [23], in which they discuss properties of ‘ratios

of estimators' in general. However, I believe that my use of their results to obtain bias and variance estimates for Weighted Uniform Sampling is new.

Weighted sampling can be used in place of Importance Sampling by using the importance function $h(\mathbf{x})$ to assign weights instead of transforming the sampling distribution. For example one could obtain such a method from the Rejection Method by replacing the acceptance-rejection step with the simple assignment of a weight, $\frac{1}{\gamma} \frac{h(\mathbf{x}_i)}{g(\mathbf{x}_i)}$, according to the acceptance probability. This eliminates the need for an additional random variable y . The resultant estimate can be written as follows:

$$\begin{aligned}\check{A}_N &= \frac{\sum_{i=1}^N \frac{f(\mathbf{x}_i)g(\mathbf{x}_i)}{h(\mathbf{x}_i)} \left(\frac{1}{\gamma} \frac{h(\mathbf{x}_i)}{g(\mathbf{x}_i)} \right)}{\sum_{i=1}^N \frac{1}{\gamma} \frac{h(\mathbf{x}_i)}{g(\mathbf{x}_i)}} \quad , \quad \mathbf{x}_i \in g(\mathbf{x}) \\ &= \frac{\sum_{i=1}^N f(\mathbf{x}_i)}{\sum_{i=1}^N \frac{h(\mathbf{x}_i)}{g(\mathbf{x}_i)}} \quad , \quad \mathbf{x}_i \in g(\mathbf{x})\end{aligned}\tag{4.4}$$

In the next section I will show that this method is biased, but that the bias is smaller than statistical error levels. Later in Chapter 5, we will see that this method is comparable to Importance Sampling when using pseudorandom sequences, as one might expect given the similarities. However, when quasi-random sequences are used Weighted Uniform Sampling is often significantly better.

4.3 Statistical Convergence Results

We have found three Monte Carlo estimates for integral $A = \int_D f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}$, with $g(\mathbf{x})$ normalized, as follows:

$$\text{Crude MC: } \hat{A}_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad , \quad \mathbf{x}_i \in g(\mathbf{x}) \quad (4.5)$$

$$\text{Importance Sample MC: } \tilde{A}_N = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i) g(\mathbf{x}_i)}{h(\mathbf{x}_i)} \quad , \quad \mathbf{x}_i \in h(\mathbf{x}) \quad (4.6)$$

$$\text{Weighted Sample MC: } \check{A}_N = \frac{\sum_{i=1}^N f(\mathbf{x}_i)}{\sum_{i=1}^N \frac{h(\mathbf{x}_i)}{g(\mathbf{x}_i)}} \quad , \quad \mathbf{x}_i \in g(\mathbf{x}) \quad (4.7)$$

In the simple case where $D = [0, 1]^d$ and $g = 1$, uniform, we have:

$$\text{Crude MC: } \hat{A}_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad , \quad \mathbf{x}_i \in U(D) \quad (4.8)$$

$$\text{Importance Sample MC: } \tilde{A}_N = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i)}{h(\mathbf{x}_i)} \quad , \quad \mathbf{x}_i \in h(\mathbf{x}) \quad (4.9)$$

$$\text{Weighted Sample MC: } \check{A}_N = \frac{\sum_{i=1}^N f(\mathbf{x}_i)}{\sum_{i=1}^N h(\mathbf{x}_i)} \quad , \quad \mathbf{x}_i \in U(D) \quad (4.10)$$

For the crude estimate we have already found the following result:

$$rmse(\hat{A}_N) = \frac{1}{\sqrt{N}} \sigma_g(f) \quad (4.11)$$

Also, the importance sampling estimate is unbiased, as shown in Section 4.1, and has a variance given by $\frac{1}{N} var_h \left(\frac{fg}{h} \right)$. Therefore it satisfies the following:

$$rmse(\tilde{A}_N) = \frac{1}{\sqrt{N}} \sigma_h \left(\frac{fg}{h} \right) \quad (4.12)$$

The computation for \check{A}_N will require a bit more work. I will consider only the simple case where $D = [0, 1]^d$ and $g = 1$, uniform, but the general case is similar. I

will assume that the absolute value of the solution, $|A|$, is bounded away from zero by $\epsilon > 0$. Recall that an importance function is only useful when the integrand is nearly always the same sign. I will also assume that $|f|$ is bounded by some constant, M_1 , and that the importance function, h , is normalized, positive-definite, and bounded by constant M_2 . These assumption are not all necessary, but they will make the analysis simpler, and cover the examples I will study.

Let,

$$\begin{aligned} F &= \sum_{i=1}^N f(x_i) \quad , \quad H = \sum_{i=1}^N h(x_i) \\ \Delta F &= \frac{F - EF}{EF} \quad , \quad \Delta H = \frac{H - EH}{EH} \end{aligned}$$

Then,

$$\begin{aligned} EF &= N A \\ EH &= N \\ E(\Delta F) &= E\left(\frac{F - EF}{EF}\right) = \frac{EF - EF}{EF} = 0 \\ E(\Delta H) &= E\left(\frac{H - EH}{EH}\right) = \frac{EH - EH}{EH} = 0 \\ E((\Delta F)^2) &= E\left(\frac{(F - EF)^2}{(EF)^2}\right) = \frac{\text{var}(F)}{(EF)^2} = \frac{\text{var}(F)}{N^2 A^2} \\ E((\Delta H)^2) &= E\left(\frac{(H - EH)^2}{(EH)^2}\right) = \frac{\text{var}(H)}{(EH)^2} = \frac{\text{var}(H)}{N^2} \\ E(\Delta F \cdot \Delta H) &= E\left(\frac{F - EF}{EF} \cdot \frac{H - EH}{EH}\right) = \frac{\text{cov}(F, H)}{N^2 A} \end{aligned}$$

Now consider the following:

$$E(\check{A}_N) = E\left(\frac{F}{H}\right)$$

$$\begin{aligned}
&= E \left(\frac{EF(1 + \Delta F)}{EH(1 + \Delta H)} \right) = \frac{EF}{EH} E \left(\frac{1 + \Delta F}{1 + \Delta H} \right) = A E \left(\frac{1 + \Delta F}{1 + \Delta H} \right) \\
&= A E [(1 + \Delta F) (1 - \Delta H + (\Delta H)^2 - (\Delta H)^3 + (\Delta H)^4 + \dots)] \\
&= A E (1 + \Delta F - \Delta H - \Delta F \Delta H + (\Delta H)^2 + \Delta F (\Delta H)^2 + \dots) \\
&= A (1 - E(\Delta F \Delta H) + E((\Delta H)^2) + E(\Delta F (\Delta H)^2) + \dots) \\
&= A - \frac{cov(F, H)}{N^2} + \frac{A var(H)}{N^2} + \dots
\end{aligned}$$

If the points are independent identically distributed (i.i.d.) random points, then we have the following:

$$\begin{aligned}
var(H) &= N var(h) \quad , \quad \text{by Eqn 2.12} \\
cov(F, G) &= E((F - EF) \cdot (H - EH)) \\
&= E \left[\left(\sum_{i=1}^N f(\mathbf{x}_i) - N Ef \right) \left(\sum_{i=1}^N h(\mathbf{x}_i) - N Eh \right) \right] \\
&= E \left(\sum_{i=1}^N (f(\mathbf{x}_i) - Ef) \cdot \sum_{i=1}^N (H(\mathbf{x}_i) - Eh) \right) \\
&= \sum_{i=1}^N \sum_{j=1}^N cov(f(\mathbf{x}_i), h(\mathbf{x}_j)) \\
&= \sum_{i=1}^N cov(f(\mathbf{x}_i), g(\mathbf{x}_i)) \quad , \quad cov(f(\mathbf{x}_i), g(\mathbf{x}_j)) = 0 \quad \forall i \neq j \\
cov(F, G) &= N \cdot cov(f, g)
\end{aligned}$$

It is also shown in the appendix that the following are true:

$$\begin{aligned}
E((\Delta H)^i) &= O(N^{-i/2}) \\
E(\Delta F (\Delta H)^i) &= O(N^{-(i+1)/2})
\end{aligned}$$

This leads to the following result:

$$E(\check{A}_N) = A + \frac{A \operatorname{var}(h)}{N} - \frac{\operatorname{cov}(f, g)}{N} + O(N^{-3/2})$$

Therefore the weighted uniform sampling estimate is biased, but the bias decreases like $1/N$.

The bias can be expressed as follows:

$$\operatorname{bias}(\check{A}_N) = \frac{A \operatorname{var}(H)}{N^2} - \frac{\operatorname{cov}(F, H)}{N^2} + \dots, \quad \text{in general} \quad (4.13)$$

$$\operatorname{bias}(\check{A}_N) = \frac{A \operatorname{var}(h)}{N} - \frac{\operatorname{cov}(f, g)}{N} + O(N^{-3/2}), \quad \text{i.i.d. samples} \quad (4.14)$$

For the variance, consider the following first:

$$\begin{aligned} E(\check{A}_N^2) &= E\left(\frac{F^2}{H^2}\right) \\ &= E\left(\frac{(EF)^2(1+\Delta F)^2}{(EH)^2(1+\Delta H)^2}\right) = A^2 E\left(\frac{(1+\Delta F)^2}{(1+\Delta H)^2}\right) \\ &= A^2 E[(1+\Delta F)^2(1-2\Delta H+3(\Delta H)^2+\dots)] \\ &= A^2 E[1+2\Delta F+(\Delta F)^2-2\Delta H-4\Delta F\Delta H+3(\Delta H)^2+\dots] \\ &= A^2 + \frac{\operatorname{var}(F)}{N^2} - \frac{4A \operatorname{cov}(F, H)}{N^2} + \frac{3A^2 \operatorname{var}(H)}{N^2} + \dots \end{aligned}$$

Then we use the basic rule $\operatorname{var}(x) = E(x^2) - (Ex)^2$, and a similar argument to the one above to obtain the results:

$$\operatorname{var}(\check{A}_N) = \frac{\operatorname{var}(F)}{N^2} + \frac{A^2 \operatorname{var}(H)}{N^2} - \frac{2A \operatorname{cov}(F, H)}{N^2} + \dots, \quad \text{in general} \quad (4.15)$$

$$\operatorname{var}(\check{A}_N) = \frac{\operatorname{var}(f)}{N} + \frac{A^2 \operatorname{var}(h)}{N} - \frac{2A \operatorname{cov}(f, h)}{N} + O(N^{-3/2}), \quad \text{i.i.d.} \quad (4.16)$$

Finally, using Equation 2.8, we have the following results:

$$\operatorname{mse}(\check{A}_N) = \operatorname{var}(\check{A}_N) + (\operatorname{bias}(\check{A}_N))^2$$

$$\begin{aligned}
&= \frac{\text{var}(f)}{N} + \frac{A^2 \text{var}(h)}{N} - \frac{2 A \text{cov}(f, h)}{N} + O(N^{-3/2}) \\
\text{rmse}(\check{A}_N) &= \frac{\sqrt{\text{var}(f) + A^2 \text{var}(h) - 2 A \text{cov}(f, h)}}{\sqrt{N}} + O(N^{-3/4}) \quad (4.17)
\end{aligned}$$

Recalling Equations 4.11 and 4.12, we see then that all three methods have a convergence rate of order $O(N^{-1/2})$ with random points, which is to be expected.

Variance reduction as a result of using Weighted Uniform Sampling is not immediately apparent. However, it is present in a way which is similar to Importance Sampling though not identical. Consider the following hypothetical situation:

Say f were strictly positive over D and we already knew the exact solution, A . Then we could use $h(\mathbf{x}) = \frac{f(\mathbf{x})}{A}$ as the importance function. In the case of importance sampling, this would give us $\frac{f(\mathbf{x})}{h(\mathbf{x})} \equiv A$ with zero variance. On the other hand, in the case of weighted uniform sampling, we would have $\text{var}(h) = \frac{\text{var}(f)}{A^2}$ and $\text{cov}(f, h) = \frac{\text{var}(f)}{A}$ which by Equation 4.14 leads again to a zero variance condition (in fact all of the higher order terms would cancel as well).

In practical applications A is, of course, unknown and f may not be single-signed. But if h is a good importance function which is nearly proportional to f , and f is single-signed or nearly so, then the positive correlation between f and h will tend to reduce the variance of the weighted estimate to much smaller values than the crude estimate.

Part II

Applying Quasi-Random Sequences

CHAPTER 5

Discontinuity Avoidance

5.1 Reduction of Convergence Rate with a Discontinuous Integrand

The use of quasi-random sequences to accelerate the convergence of Monte Carlo estimates is, in general, significantly more successful when the computed estimates vary continuously as a function of the quasi-random points used to compute them. This means in particular that when estimating an integral, the integrand should be continuous and any transformations or other operations involved in the process of evaluating the integrand should be continuous as well.

The sensitivity of quasi-random sequences to discontinuities is suggested by the Koksma-Hlawka Inequality, which fails whenever discontinuities cause the variation of the integrand to become infinite. Computational results, some of which will be presented here, also support the conclusion that discontinuities should be avoided when using quasi-random sequences.

We can therefore greatly improve the effectiveness of quasi-random sequences in cases where traditional Monte Carlo methods inherently produce discontinuous integrands by modifying these methods so that they instead produce continuous

integrands. It is important to realize that in complex situations the integrand is often not given explicitly but depends on the entire process of generating samples and evaluating functions. The term *effective integrand* will refer to this entire process acting on the initial samples of pseudorandom or quasi-random points. A discontinuity anywhere in the process can hurt the performance of quasi-random sequences. Thus, when one is able to successfully eliminate all the discontinuities, the acceleration of convergence is often significantly improved, revealing the full benefits of quasi-random sequences. In the subsequent sections we shall examine the integration of characteristic functions and the use of the Rejection Method in this respect.

First, let us examine the relationship between continuity and quasi-random convergence more closely. The Koksma-Hlawka Inequality provides an upper bound on convergence as $|err| \leq V(f)D$, where D is the discrepancy of the sequence and $V(f)$ is the Hardy-Krause variation of the integrand. Consider the following integrand on $I^2 = [0, 1]^2 \subset \mathbb{R}^2$:

$$f(x, y) = \begin{cases} 0 & \text{if } x \leq y \\ 1 & \text{otherwise} \end{cases}$$

The Hardy-Krause variation of this function is unbounded because for any partition of the domain, at each x_i of the partition, ($i = 1, \dots, m_1$), there must be at least one subinterval, $J = [x_{i-1}, x_i] \times [y_{j-1}, y_j]$, for which $\Delta(f; J)$ is equal to one. (Note: See Section 3.2 for definitions of these terms.) Therefore, as the partition becomes finer and $m_1 \rightarrow \infty$, the Hardy-Krause variation also must become infinite.

An infinite variation (in the Hardy-Krause sense) means that the Koksma-Hlawka Inequality fails to hold for this function. This suggests that an attempt to integrate the above function using quasi-random sequences might not even converge, and it puts into doubt the accelerated convergence expected when using quasi-random sequences. By a similar argument one can show that the variation is always infinite when a discontinuity occurs along any direction which is not parallel to a coordinate axes. It may seem strange that the direction should matter, but quasi-random sequences are designed to minimize the discrepancy which is itself tied to the cartesian coordinate system. Therefore, one should not be surprised to find that the directions can be a factor. Of course in the case above one could simply rotate the coordinates to get the discontinuity along an axial direction, but in more general cases the discontinuities are seldom strictly along straight lines.

The Koksma-Hlawka Inequality suggests an all or nothing situation with regard to convergence using quasi-random sequences. However, computational results suggest a more graduated situation. When there are no discontinuities at all, convergence rates are of order $O(N^{-1}(\log N)^d)$, and when there are numerous large discontinuities in non-coordinate directions, convergence rates are actually very near to order $O(N^{-1/2})$. With fewer or smaller discontinuities, the results are typically between these two extremes. A heuristic argument for this is that near jumps, the positioning of a particular point just to one side or the other is basically 'random' at such an infinitesimal scale. Hence, convergence rates of order $O(N^{-1/2})$ associated with random points are observed, more so the more jumps

there are and the larger they are. Unfortunately, aside from the Koksma-Hlawka Inequality, there is little theory with regards to this issue since it is such a relatively young topic. However, the computational results found by myself and others have been quite clear and consistent (see for example Press and Teukolsky [47] and Morokoff [41]). In computational studies it is most interesting to notice how eliminating the discontinuities in an integrand has almost no effect on convergence with pseudorandom sequences while it greatly improves convergence with quasi-random sequences.

5.2 Example: Integration of Characteristic Functions

In order to explore the effectiveness of avoiding discontinuities when applying quasi-random sequences to Monte Carlo integration, I have studied the integration of characteristic functions. Such functions, denoted by the symbol χ , are defined as having the value 1 over a specified set, and the value 0 over its complement.

$$\chi_E(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in E \subset \mathbb{R}^d \\ 0 & \text{otherwise} \end{cases}$$

Such a function is clearly discontinuous along the set's boundary, where its value jumps from 0 to 1, and unless the boundary is completely parallel to coordinate axes the variation is infinite. However, since the integral is simply the volume of the set, E , in d dimensions, it is often possible to replace the characteristic function with a continuous function which has its edges 'smoothed out' without changing

the value of the integral (see Figure 5.1). In such a case, the crude Monte Carlo estimates of each of the above equal integrals have nearly the same error levels when pseudorandom points are used, but when quasi-random points are used, the smoothed version converges more rapidly and the estimates for it are more accurate than those for the original integral. This is to be expected based on the discussion in the preceding section.

Example 2 *Comparison of crude Monte Carlo estimates using pseudorandom and quasi-random (Halton) sequences of points for the following two 3 dimensional integrands over I^3 :*

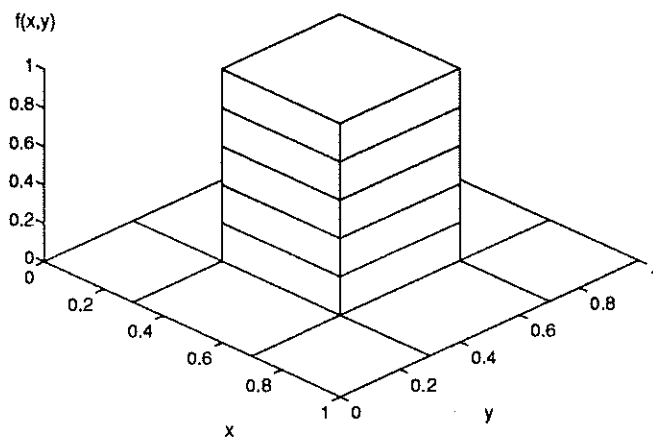
Let,

$$\begin{aligned} u_1 &= \frac{\sqrt{2}x_1 + x_2 + x_3}{2 + \sqrt{2}} - \frac{1}{2} \in \left[-\frac{1}{2}, \frac{1}{2}\right] \\ u_2 &= \frac{\sqrt{2}(1 - x_1) + x_2 + x_3}{2 + \sqrt{2}} - \frac{1}{2} \in \left[-\frac{1}{2}, \frac{1}{2}\right] \\ u_3 &= \frac{-x_2 + x_3}{2} \in \left[-\frac{1}{2}, \frac{1}{2}\right] \end{aligned}$$

$$\begin{aligned} f_A(x_1, x_2, x_3) &= \prod_{k=1}^3 \begin{cases} 1 & \text{if } |u_k| \leq .2 \\ 0 & \text{otherwise} \end{cases} \\ f_B(x_1, x_2, x_3) &= \prod_{k=1}^3 \begin{cases} 1 & \text{if } |u_k| \leq .2 - \delta \\ 0 & \text{if } |u_k| \geq .2 + \delta \\ \frac{\delta - (|u_k| - .2)}{2\delta} & \text{otherwise} \end{cases} \end{aligned}$$

Then f_A is the characteristic function of a rotated cube, while f_B is a continuous

Original, discontinuous function :



Revised, continuous function :

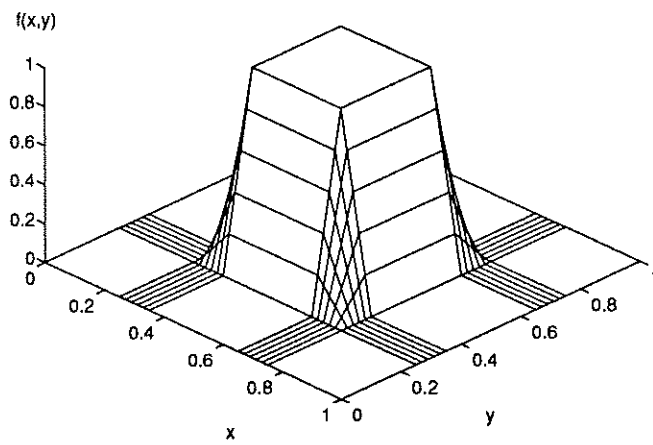


Figure 5.1: 2D Versions of Functions used in Example 2.

function which has the same integral over I^3 :

$$\begin{aligned}
\int_0^1 \int_0^1 \int_0^1 f_A(x_1, x_2, x_3) dx_1 dx_2 dx_3 &= \int_{-.5}^{.5} \int_{-.5}^{.5} \int_{-.5}^{.5} f_A \left| \frac{d(u_1, u_2, u_3)}{d(x_1, x_2, x_3)} \right| du_1 du_2 du_3 \\
&= \int_{-.2}^{.2} \int_{-.2}^{.2} \int_{-.2}^{.2} 1.0 du_1 du_2 du_3 = (.4)^3 = .064 \\
\int_0^1 \int_0^1 \int_0^1 f_B(x_1, x_2, x_3) dx_1 dx_2 dx_3 &= \int_{-.5}^{.5} \int_{-.5}^{.5} \int_{-.5}^{.5} f_B \left| \frac{d(u_1, u_2, u_3)}{d(x_1, x_2, x_3)} \right| du_1 du_2 du_3 \\
&= \left\{ \int_{-.2+\delta}^{.2-\delta} 1.0 du + \int_{-.2-\delta}^{-.2+\delta} \frac{\delta - (-u - .2)}{2\delta} du \right. \\
&\quad \left. + \int_{.2-\delta}^{.2+\delta} \frac{\delta - (u - .2)}{2\delta} du \right\}^3 \\
&= (.4 - 2\delta + \delta + \delta)^3 = .064
\end{aligned}$$

The results of crude Monte Carlo estimation in three different cases are compared: First, the discontinuous case; second, the continuous case with weak smoothing ($\delta = .025$); and third, the continuous case with strong smoothing ($\delta = .1$). See Example 1 in Section 2.3 for further explanation of the results. Here, as in Example 1, 75 repetitions are used to obtain estimates. Results for selected values of N are shown in Table 5.1 along with convergence estimates. Full results are plotted in Figure 5.2.

The results for this example show that the quasi-random sequences are much more sensitive to the continuity of the integrand than pseudorandom sequences. The slight decreases in the error seen with pseudorandom sequences are explained by a slight variance reduction as a result of the smoothing. However, the much larger decreases for quasi-random sequences cannot be explained by this alone.

It should be obvious by now that significant reductions of error are possible

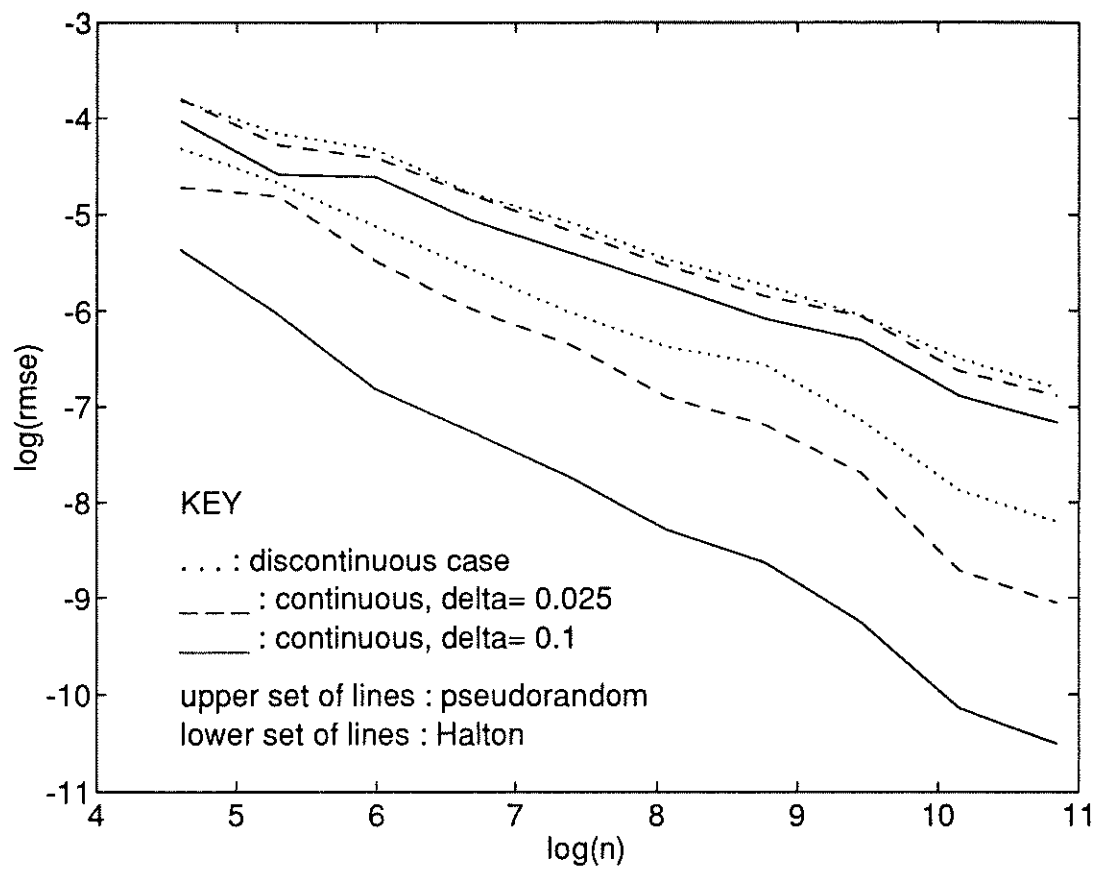


Figure 5.2: Log-Log Plot for Example 2.

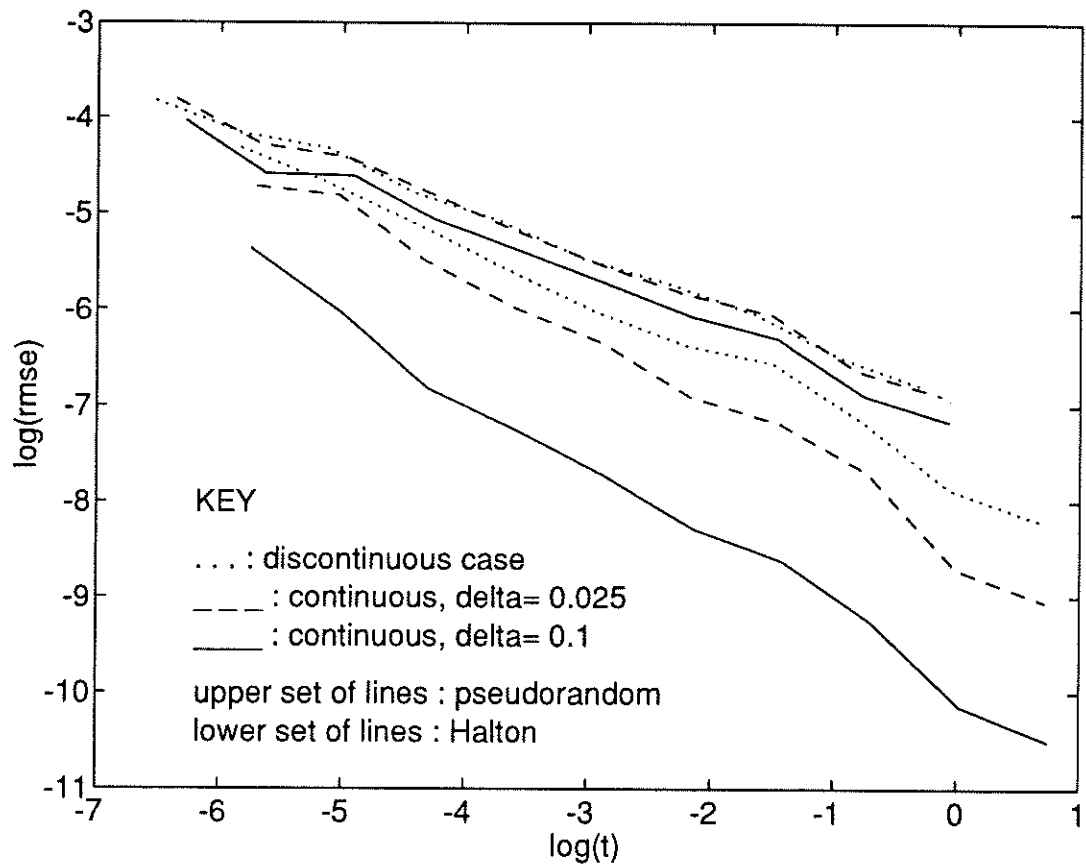


Figure 5.3: Log-Log Plot vs $t(\text{sec})$ for Example 2.

Pseudorandom Sequences

	Discontinuous		Continuous, $\delta = .025$		Continuous, $\delta = .1$	
N	$rmse(\hat{A}_N)$	t_N	$rmse(\hat{A}_N)$	t_N	$rmse(\hat{A}_N)$	t_N
200	0.015541	0.00293	0.013824	0.00347	0.010166	0.00360
800	0.008365	0.01200	0.008272	0.01387	0.006334	0.01413
3200	0.004185	0.04800	0.003935	0.05507	0.003230	0.05813
12800	0.002366	0.19227	0.002341	0.22133	0.001831	0.23227
51200	0.001110	0.76613	0.001022	0.87520	0.000775	0.93227

Halton Quasi-Random Sequences

	Discontinuous		Continuous, $\delta = .025$		Continuous, $\delta = .1$	
N	$rmse(\hat{A}_N)$	t_N	$rmse(\hat{A}_N)$	t_N	$rmse(\hat{A}_N)$	t_N
200	0.009324	0.00587	0.008139	0.00653	0.002381	0.00667
800	0.003800	0.02507	0.002522	0.02720	0.000704	0.28000
3200	0.001710	0.10933	0.001003	0.11440	0.000252	0.11787
12800	0.000789	0.45440	0.000456	0.48187	0.000096	0.49467
51200	0.000275	1.93453	0.000117	2.03747	0.000027	2.09507

Root Mean Square Error, $rmse$, Convergence Rates

	Discontinuous		Continuous, $\delta = .025$		Continuous, $\delta = .1$	
SeqType	vs N	vs t_N	vs N	vs t_N	vs N	vs t_N
pseudo	-0.481	-0.479	-0.488	-0.489	-0.490	-0.491
Halton	-0.618	-0.593	-0.713	-0.690	-0.802	-0.772

Table 5.1: Results for Example 2, 75 trials.

when quasi-random sequences are used in place of pseudorandom. In this example, even in the discontinuous case, the quasi-random sequences outperformed pseudorandom. However, in more complex situations, this is not always the case. It sometimes appears that quasi-random sequences will not be useful because there is no appreciable gain in convergence when they are first plugged into an existing method which is inherently discontinuous. However, in many of these cases, if one finds a way to modify the method so that the estimates are a continuous function of the sample points, then quasi-random sequences will indeed improve the convergence, thereby justifying their use.

5.3 Application to Variance Reduction with Importance Functions

The concept of avoiding discontinuities becomes a practical consideration when quasi-random sequences are applied to Monte Carlo methods which are inherently discontinuous. There are many instances of this, and when quasi-random sequences are directly substituted for pseudorandom sequences in such cases, the results are often, not surprisingly, quite poor. This has led some, I believe, to prematurely decide that quasi-random sequences are ineffective for such situations. However, I have found that when one is able to modify such methods so as to avoid discontinuities, dramatic improvements are possible and convergence rates can indeed be improved using quasi-random sequences.

One such instance, which will now be examined, is when an importance function

is used as a variance reduction technique. This is very common. Often, an Importance Sampled Monte Carlo estimate is computed using the Rejection Method to transform a sequence of uniformly distributed points to the required distribution for importance sampling. There is another, less widely used alternative, perhaps because it is biased, which is Weighted Uniform Sampling. It involves assigning weights to the sample points instead of accepting or rejecting them. Both of these methods are discussed and analyzed from a statistical point of view in Chapter 4. In the next few sections I will show that the Rejection Method leads to discontinuous integrands and therefore should be avoided when one intends to use quasi-random points, while Weighted Uniform Sampling is well-suited for quasi-random points. For cases in which the Rejection Method is strongly preferred, I present a smoothed version which will work better than the Rejection Method with quasi-random points, but usually not quite as well as Weighted Uniform Sampling. Finally, some computational results will be presented.

5.3.1 Discontinuous Nature of the Rejection Method

The Rejection Method is a method for transforming samples from an initial distribution $g(\mathbf{x})$ to a desired new distribution $h(\mathbf{x})$. The algorithm is presented and justified in Section 4.1.1. Here, I will treat the case when the initial distribution is uniform over the unit d -dimensional cube, $I^d = [0, 1]^d$. The simplified algorithm is then as follows for the **Rejection Method**:

Repeat until N points have been accepted:

1. Sample (\mathbf{x}_t, y_t) from $U([0, 1]^{d+1})$.
2. If $y_t < \frac{h(\mathbf{x}_t)}{\gamma}$, where $\gamma \geq \sup_{\mathbf{x} \in I^d} h(\mathbf{x})$, then accept \mathbf{x}_t .
3. Otherwise, reject \mathbf{x}_t .

The discontinuous nature of the Rejection Method is apparent when one expresses an Importance Sampled Monte Carlo estimate obtained using the Rejection Method explicitly in terms of the original uniform sample. This is done by using a characteristic function to represent the acceptance/rejection step as follows:

$$\tilde{w}(\mathbf{x}_t, y) = \chi \left\{ y < \frac{h(\mathbf{x}_t)}{\gamma} \right\}$$

where $\tilde{w}(\mathbf{x}_t, t)$ is the acceptance weight of the trial point \mathbf{x}_t , given the value of y .

This only takes on the values 0 or 1.

Then we have the following:

$$\hat{A}_N = \frac{1}{N} \sum_{i=1}^{N^*} \frac{f(\mathbf{x}_i^*)}{h(\mathbf{x}_i^*)} \chi \left\{ y_i < \frac{h(\mathbf{x}_i^*)}{\gamma} \right\} \quad , \quad (\mathbf{x}_i^*, y_i) \in U([0, 1]^{d+1})$$

where N^* is the number of trial points needed to get N acceptances, and (\mathbf{x}_i^*, y_i) represents the i 'th sample point from the original uniform sample.

Written in this form, it is apparent that the effective integrand given by $\frac{f}{h} \tilde{w}$, with a weight of \tilde{w} , over I^{d+1} , is discontinuous, even when f and h are themselves continuous functions. The discontinuity comes along the boundary between acceptance and rejection, which in general is not parallel to any coordinate axis.

Therefore, the quasi-random convergence limits from the Koksma-Hlawka fail to hold, and as discussed earlier, one should not expect significantly improved convergence rates. This appears to be exactly the case in computational studies. An example will be presented shortly.

5.3.2 Weighted Uniform Sampling as a Continuous Alternative

The Rejection Method is discontinuous as a result of the inherently discontinuous nature of the binary decision: accept or reject. Therefore to eliminate the discontinuity, this decision process must be either smoothed out or eliminated entirely. This can be accomplished by allowing for a weighted sample and associating with each point a weight according to its acceptance rate. Weighted Uniform Sampling is the result when one eliminates the acceptance/rejection step completely and simply assigns each sample point, \mathbf{x}_i , a weight according to $\frac{h(\mathbf{x}_i)}{\gamma}$.

The new estimate is obtained as follows:

$$\begin{aligned}\check{A}_N &= \frac{\sum_{i=1}^N \frac{f(\mathbf{x}_i)}{h(\mathbf{x}_i)} \cdot \frac{h(\mathbf{x}_i)}{\gamma}}{\sum_{i=1}^N \frac{h(\mathbf{x}_i)}{\gamma}}, \quad \mathbf{x}_i \in U([0, 1]^d) \\ &= \frac{\sum_{i=1}^N \frac{f(\mathbf{x}_i)}{\gamma}}{\sum_{i=1}^N \frac{h(\mathbf{x}_i)}{\gamma}} \\ &= \frac{\sum_{i=1}^N f(\mathbf{x}_i)}{\sum_{i=1}^N h(\mathbf{x}_i)}, \quad \text{Eqn 4.10}\end{aligned}$$

Therefore, there is a very close connection between importance sampling using the Rejection Method and Weighted Uniform Sampling. However, in the latter case we now have what is essentially the ratio of two Crude Monte Carlo integration

estimates. An estimate of the original function, f , in the numerator, and an estimate of the importance function, h , in the denominator. It may seem odd to do this, and the resulting estimate is biased. But, as explained in Section 4.3, the bias is extremely small, while the correlations between numerator and denominator can significantly reduce the variance. Importantly, as long as both f and h are continuous, the weighted uniform sampling estimate is continuous, unlike the Rejection Method. This leads to much better quasi-random results. In addition, the extra variable y is eliminated, so the number of dimensions is d instead of $d + 1$. This saves some computation time, and, as discussed in Chapter 6, also helps the performance of quasi-random sequences. Therefore, Weighted Uniform Sampling should produce significantly better results with quasi-random sequences than the Rejection Method. The computational examples will demonstrate this.

5.3.3 The Smoothed Rejection Method

Although Weighted Uniform Sampling discussed above is an excellent alternative to the Rejection Method, there are cases when one would still rather use the Rejection Method. An example would be a situation in which the integrand has regions of very low importance weight and evaluation of the function f is costly. In this case the Rejection Method allows one to reject most of the points in such regions, saving time and effort. On the other hand, Weighted Uniform Sampling requires one to evaluate f at all of the sample points, but simply count some of

them less than others using weights. This can lead to extra work being expended on relatively unimportant data points, and is usually reflected in higher variance using Weighted Uniform Sampling than using Importance Sampling with the Rejection Method (recall that though both methods reduce variance, the reduction is not identical). For such occasions, an alternative is the following slightly modified version of the Rejection Method, dubbed “Smoothed Rejection.” The algorithm for **Smoothed Rejection** follows:

Repeat until weight of accepted points exceeds N :

1. Sample (\mathbf{x}_t, y_t) from $U([0, 1]^{d+1})$.
2. If $y_t < \frac{h(\mathbf{x}_t)}{\gamma} - \frac{1}{2}\delta$, where $\gamma \geq \sup_{\mathbf{x} \in I^d} h(\mathbf{x})$ and $\delta \ll 1$, then $w = 1$.
3. Else if $y_t > \frac{h(\mathbf{x}_t)}{\gamma} + \frac{1}{2}\delta$, then $w = 0$.
4. Else, accept \mathbf{x}_t with its weight given by $w = \frac{\left(\frac{h(\mathbf{x}_t)}{\gamma} + \frac{1}{2}\delta\right) - y_t}{\delta}$

Representing the acceptance weight given \mathbf{x} and y by $w(\mathbf{x}, y)$, one can show that this method generates samples from the correct distribution as follows:

$$\begin{aligned}
 f_{\text{accept}}(\mathbf{x}) &= \frac{f_{\text{select}}(\mathbf{x}) E(w(\mathbf{x}))}{\int_I^d f_{\text{select}}(\boldsymbol{\xi}) E(w(\boldsymbol{\xi})) d\boldsymbol{\xi}} \\
 &= \frac{1 \cdot \int_0^1 w(\mathbf{x}, y) dy}{\int_I^d [1 \cdot \int_0^1 w(\boldsymbol{\xi}, y) dy] d\boldsymbol{\xi}} \\
 &= \frac{\frac{h(\mathbf{x})}{\gamma}}{\int_I^d \frac{h(\boldsymbol{\xi})}{\gamma} d\boldsymbol{\xi}} \\
 &= \frac{h(\mathbf{x})/\gamma}{1/\gamma} \\
 f_{\text{accept}}(\mathbf{x}) &= h(\mathbf{x})
 \end{aligned}$$

There is an additional amount of work required by Smoothed Rejection as compared to ordinary Rejection for two reasons. First, there is the extra work associated with assigning each sample point a weight and using those weights later on. This is often a very small amount of extra work. Second, although points with weight 0 can be thrown out completely, those points with non-zero weights less than 1 require just as much work as points with weight 1. Therefore, for a sample of total acceptance weight N , there will likely be more than N points with which subsequent computations must be performed. This can be significant. However, in most cases by setting the fixed constant δ to be sufficiently small one can ensure that the vast majority of points will be assigned weights of 0 or 1. This minimizes the amount of extra work required. On the other hand, if δ is taken too small, one is effectively back to the original Rejection Method.

The continuous nature of the Smoothed Rejection Method is apparent when one expresses the estimate obtained using the Smoothed Rejection Method in terms of the original uniform sample points. Let $w(\mathbf{x}_i, y_i)$ represent the acceptance weight. Then, we have the following:

$$\hat{A}_N = \frac{1}{N} \sum_{i=1}^{N^*} \frac{f(\mathbf{x}_i^*)}{h(\mathbf{x}_i^*)} w(\mathbf{x}_i^*, y_i) \quad , \quad (\mathbf{x}_i^*, y_i) \in U([0, 1]^{d+1})$$

where N^* is the number of trial points needed to get a total acceptance weight of at least N , and (\mathbf{x}_i^*, y_i) represents the i 'th sample point from a pseudorandom or quasi-random sequence of $d + 1$ dimensions.

Since $w(\mathbf{x}, y)$ is a continuous function, the estimate is continuous as long as f

and h are continuous functions. (Note: This is not precisely true since a slight weight increase in one point may put the total weight over N a point earlier or later, however this slight discontinuity is clearly insignificant.)

With Smoothed Rejection, as with the Rejection Method, many of the points in ‘low importance regions’ will be rejected, so f will not have to be evaluated as often in these regions as when Weighted Uniform Sampling is used. That is why Smoothed Rejection could sometimes be preferable to Weighted Uniform Sampling.

However, my experience has been that with quasi-random sequences, Smoothed Rejection, while better than the Rejection Method, is usually not as effective as Weighted Uniform Sampling. Although it is basically continuous, there is still a relatively sharp slope in acceptance weights in the transition from acceptance to rejection regions. This leads to higher variation and poorer quasi-random performance, in general, than for Weighted Uniform Sampling, where this boundary is completely eliminated. Also, Smoothed Rejection requires $d + 1$ dimensional sample points, while Weighted Uniform Sampling only requires d dimensions.

5.4 Computational Examples and Results

In order to compare Crude Monte Carlo, Weighted Uniform Sampling, Importance Sampling using the Rejection Method, and Importance Sampling using Smoothed Rejection, I have considered several multidimensional integration problems. They have been chosen to be somewhat representative of the kinds of inte-

grals for which Monte Carlo methods might be applied. As we will see, the results support the claim that Weighted Uniform Sampling can produce results which are far superior to those possible using the Rejection Method or Smoothed Rejection when quasi-random sequences are used. Also, the results confirm that Smoothed Rejection is better than the Rejection Method with quasi-random sequences.

As discussed in the first example in Chapter 2, when the exact solution is unknown, an effective estimate of the error can often be obtained by using repeated samples of size N and then computing the sample variance of the estimates computed. This depends on an assumption that the error in each successive estimate is essentially independent of the others, which should clearly be true of pseudorandom estimates, but may be questionable for quasi-random estimates. This is an open area for future study.

For the examples here, I computed direct estimates of the root mean square error, as well as the estimates discussed above in order to compare them and get a sense of the validity of the assumption above. In all of these examples, I found few differences between these two estimates of the root mean square error.

To check the formulas for the bias and variance of Weighted Uniform Sampling derived in Chapter 4, I compared estimates based on those formulas with direct estimates of the bias and variance over repeated trials. The formulas checked out well since there were few differences, except when the bias became too small to measure beneath the statistical errors.

5.4.1 Computations for Variance Reduction Examples

We are, in general, estimating the following integral:

$$A = \int_{I^d} f(\mathbf{x}) d\mathbf{x}$$

where I^d is the unit cube in d dimensions.

The estimates are as follows:

$$\text{Crude MC: } \hat{A}_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad , \quad \mathbf{x}_i \in U(I^d)$$

$$\text{W.U.S. MC: } \check{A}_N = \frac{\sum_{i=1}^N f(\mathbf{x}_i)}{\sum_{i=1}^N h(\mathbf{x}_i)} \quad , \quad \mathbf{x}_i \in U(I^d)$$

$$\text{Rej Meth MC: } \tilde{A}_N = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i)}{h(\mathbf{x}_i)} \quad , \quad \mathbf{x}_i \text{ an accepted point.}$$

$$\text{Smooth Rej MC: } \acute{A}_N = \frac{1}{N} \sum_{i=1}^{N^*} w_i \frac{f(\mathbf{x}_i)}{h(\mathbf{x}_i)} \quad , \quad w_i = \text{acceptance weight, } N^* \geq N$$

where for the last estimate, N^* is chosen so that the sum of the acceptance weights is within one unit of N .

We will be examining the root mean square error as a function of N for each of these estimators and either pseudorandom or quasi-random sequences, which is approximated directly as follows:

$$rmse(N) \approx \sqrt{\frac{1}{R} \sum_{k=1}^R (\acute{A}_N^{(k)} - A)^2} \quad (5.1)$$

where $\acute{A}_N^{(k)}$ represents the k -th computed estimate out of R total trials using new samples of size N for each trial and one of the above estimators.

This can be compared with the standard deviation of the set of R estimates

which is given as follows:

$$sdr(N) \approx \sqrt{\frac{1}{R-1} \sum_{k=1}^R (\dot{A}_N^{(k)} - \overline{A_N})^2} \quad (5.2)$$

where $\overline{A_N}$ is the average of the R estimates.

In the examples, the values of *rmse* are computed using the best available estimates of the exact solutions, A , in Equation 5.1. These were obtained either using Gaussian cubature methods or Monte Carlo estimates using extremely high values of N . On the other hand, the values of *sdr* computed using Equation 5.2 do not require any prior knowledge of the solution, but they also estimate the root mean square error (Under the assumption that the errors from repeated trials are nearly independent, as discussed above and in Example 1).

One can also compare the computed estimates of the root mean square error above with the expected root mean square error for independent random points given by the product of $N^{-1/2}$ and the standard deviation of f (or the reduced standard deviation when variance reduction is used). For Crude Monte Carlo and the Rejection Method, this can be estimated using the average sample variance as follows:

$$sd(N) \approx \frac{1}{\sqrt{N}} \sqrt{\frac{1}{R} \sum_{k=1}^R \frac{N}{N-1} \left[\frac{1}{N} \sum_{i=1}^N f^*(\mathbf{x}_i)^2 - \left(\frac{1}{N} \sum_{i=1}^N f^*(\mathbf{x}_i) \right)^2 \right]}$$

where f^* is either f for Crude Monte Carlo, or $\frac{f}{h}$ for the Rejection Method.

Similarly, for smoothed rejection, with the weight included, as follows:

$$sd(N) \approx \frac{1}{\sqrt{N}} \sqrt{\frac{1}{R} \sum_{k=1}^R \frac{N}{N-1} \left[\frac{1}{N} \sum_{i=1}^{N^*} w_i f^*(\mathbf{x}_i)^2 - \left(\frac{1}{N} \sum_{i=1}^{N^*} w_i f^*(\mathbf{x}_i) \right)^2 \right]}$$

where f^* is again equal to $\frac{f}{h}$.

For weighted uniform sampling, The root mean square error for random points is given by Equation 4.17, which gives us the following estimate:

$$\begin{aligned} sd(N)^2 \approx & \frac{1}{N} \frac{1}{R} \sum_{k=1}^R \left\{ \frac{N}{N-1} \left[\frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)^2 - \left(\frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \right)^2 \right] \right. \\ & + \left(\overline{A_N} \right)^2 \frac{N}{N-1} \left[\frac{1}{N} \sum_{i=1}^N h(\mathbf{x}_i)^2 - \left(\frac{1}{N} \sum_{i=1}^N h(\mathbf{x}_i) \right)^2 \right] \\ & \left. - 2 \overline{A_N} \frac{N}{N-1} \left[\frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) h(\mathbf{x}_i) - \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \frac{1}{N} \sum_{i=1}^N h(\mathbf{x}_i) \right] \right\} \end{aligned}$$

We expect $sd(N)$, in general, to be close to $rmse(N)$ and $sdr(N)$ whenever pseudorandom sequences are used, but when quasi-random sequences are used we expect $rmse(N)$ and $sdr(N)$ to be significantly lower than $sd(N)$, indicating the improvement obtained from quasi-random sequences.

Finally for Weighted Uniform Sampling, the bias and variance are estimated using Equations 4.13 and 4.15, derived in Chapter 4 (ignoring the higher terms), which gives us estimates that should be valid for both pseudorandom and quasi-random sequences. In the examples, the computations using the formulas from Chapter 4 are given as follows:

$$\begin{aligned} b^*(N) &= \overline{A_N} \frac{1}{R-1} \sum_{k=1}^R (H_N^{(k)} - \overline{H_N})^2 - \frac{1}{R-1} \sum_{k=1}^R (F_N^{(k)} H_N^{(k)} - \overline{F_N H_N})^2 \\ sdr^*(N)^2 &= \frac{1}{R-1} \sum_{k=1}^R (F_N^{(k)} - \overline{F_N})^2 + \left(\overline{A_N} \right)^2 \frac{1}{R-1} \sum_{k=1}^R (H_N^{(k)} - \overline{H_N})^2 \\ &\quad - 2 \overline{A_N} \frac{1}{R-1} \sum_{k=1}^R (F_N^{(k)} H_N^{(k)} - \overline{F_N H_N})^2 \end{aligned}$$

where $F_N^{(k)}$, $H_N^{(k)}$ are the averages of f and h respectively over the k -th trial, and bars indicate averages over all the trials.

We expect $sdr^*(N)$ to be close to $sdr(N)$, and consequently $rmse(N)$, since these all estimate the same quantity.

The value of $b^*(N)$ can be checked by computing $\overline{A_N}$ over enough trials so that the statistical error in $\overline{A_N}$ is small compared to the bias. Then the bias is approximated by $\overline{A_N} - A$, when the exact solution is known. However, this often requires an extremely large number of trials because the bias decreases so rapidly as N increases. A close examination of a few selected cases, which will not be presented here, has confirmed that $b^*(N)$ is indeed a good estimate of the bias in these examples.

(Technical note: The plots for this example and most of those throughout the rest of this thesis have the y-axis labeled $\log(rmse)$ which means “the logarithm base e of the root mean square error.” However, in order to be more realistic, I have actually used the sample variance, referred to above as sdr , instead of the direct estimate of the rms error, referred to above as $rmse$, when drawing up the plots. Therefore, a more precise label for the y-axis might be $\log(sdr)$, but I felt that using the notation $\log(rmse)$ would make it more immediately obvious that the y-axes were showing a measurement of the accuracy of the estimates. In fact, since sdr is itself an estimate of the root mean square error, the notation may be justified in that sense. In most cases, as the tables indicate, the computed values of sdr and $rmse$ were very close, as expected, so the distinction is largely a notational one.)

One final observation is that in the examples, the bias of Weighted Uniform

Sampling is significantly smaller when using quasi-random sequences than pseudo-random sequences. The reason goes back to the difference between Equations 4.13 and 4.14. As a result, it appears to be safe to ignore the bias for most moderate or higher values of N , even when quasi-random sequences are used.

5.4.2 Description of Examples, Computational Results

Example 3 *Monte Carlo integration over $I^5 = [0, 1]^5$ of the function:*

$$f(\mathbf{x}) = \exp \left(\sum_{i=1}^5 a_i x_i^2 \frac{2 + \sin \left(\sum_{j=1, j \neq i}^5 x_j \right)}{2} \right)$$

Using the positive definite importance function:

$$h(\mathbf{x}) = \frac{1}{\eta} \exp \left(\sum_{i=1}^5 a_i x_i^2 \right)$$

where $\mathbf{a} = (1, \frac{1}{2}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$ and $\eta = \int_{I^5} \exp \left(\sum_{i=1}^5 a_i x_i^2 \right) d\mathbf{x}$ so that h is normalized (i.e. its integral is one). (Note: η is easily computed with high accuracy as the product of five one-dimensional integrals, using for example Gaussian quadrature.)

The results are presented in Tables 5.2-5.5 and graphically in Figure 5.3.

Example 4 *Monte Carlo integration over $I^3 = [0, 1]^3$ of the function:*

$$f(x, y, z) = \sin \left(\frac{\pi(x + \sqrt{xyz} + y^2)}{6} \right)$$

Using the positive definite importance function:

$$h(x, y, z) = \frac{54}{61} (x + \sqrt{xyz} + y^2)$$

where h is again normalized so that $\int_{I^3} h(x, y, z) dx dy dz = 1$.

The results are presented in Tables 5.6-5.9 and graphically in Figure 5.4.

Example 5 Monte Carlo integration over $I^7 = [0, 1]^7$ of the function:

$$f(x_1, \dots, x_7) = e^{1 - (\sin^2(\frac{\pi}{2}x_1) + \sin^2(\frac{\pi}{2}x_2) + \sin^2(\frac{\pi}{2}x_3))} \arcsin(\sin(1) + \frac{x_1 + \dots + x_7}{200})$$

Using the positive definite importance function:

$$h(x_1, \dots, x_7) = \frac{1}{\eta} e^{1 - (\sin^2(\frac{\pi}{2}x_1) + \sin^2(\frac{\pi}{2}x_2) + \sin^2(\frac{\pi}{2}x_3))}$$

where η is computed by expressing h as follows:

$$h(x_1, \dots, x_7) = e e^{-\sin^2(\frac{\pi}{2}x_1)} e^{-\sin^2(\frac{\pi}{2}x_2)} e^{-\sin^2(\frac{\pi}{2}x_3)}$$

Then,

$$\eta = \int_{I^7} h(x_1, \dots, x_7) dx_1 \dots dx_7 = e \cdot \left(\int_0^1 e^{-\sin^2(\frac{\pi}{2}x)} dx \right)^3$$

which is easily approximated to high accuracy as a one-dimensional integral.

The results are presented in Table 5.10-5.13 and graphically in Figure 5.5.

Crude Monte Carlo (variance= 3.16 E -2)

	Pseudorandom, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	4.44 E -3	4.538 E -3	4.571 E -3	0.044
6400	2.22 E -3	2.064 E -3	2.065 E -3	0.174
25600	1.11 E -3	1.189 E -3	1.190 E -3	0.697
102400	5.56 E -4	5.723 E -4	5.775 E -4	2.791
409600	2.78 E -4	2.873 E -4	2.876 E -4	11.161
convergence	sdr vs. N : -0.480 , sdr vs. t : -0.481			
	Halton Quasi-Random, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	4.45 E -3	1.727 E -4	1.727 E -4	0.081
6400	2.22 E -3	4.142 E -5	4.144 E -5	0.342
25600	1.11 E -3	1.229 E -5	1.229 E -5	1.439
102400	5.56 E -4	2.977 E -6	2.977 E -6	6.002
409600	2.78 E -4	8.086 E -7	8.086 E -7	25.168
convergence	sdr vs. N : -0.980 , sdr vs. t : -0.946			

Table 5.2: Example 3, Crude Monte Carlo.

Weighted Uniform Sampling, (variance= 2.61 E -3)

	Pseudorandom, 75 trials					
N	$sd(N)$	$sdr(N)$	$sdr^*(N)$	$b^*(N)$	$rmse(N)$	$t(N)$
1600	1.28 E -3	1.344 E -3	1.343 E -3	-9.11 E -6	1.363 E -3	0.056
6400	6.37 E -4	7.050 E -4	7.050 E -4	-1.85 E -6	7.063 E -3	0.223
25600	3.19 E -4	3.625 E -4	3.623 E -4	-6.11 E -7	3.625 E -4	0.891
102400	1.60 E -4	1.755 E -4	1.754 E -4	-1.42 E -7	1.756 E -4	3.566
409600	7.98 E -5	8.624 E -5	8.624 E -5	-3.35 E -8	8.624 E -5	14.268
convergence		sdr vs. N : -0.484 , sdr vs. t : -0.484				

	Halton Quasi-Random, 75 trials					
N	$sd(N)$	$sdr(N)$	$sdr^*(N)$	$b^*(N)$	$rmse(N)$	$t(N)$
1600	1.28 E -3	7.259 E -5	7.259 E -5	-9.72 E -9	7.259 E -5	0.094
6400	6.38 E -4	2.106 E -5	2.106 E -5	-5.76 E -10	2.106 E -5	0.391
25600	3.19 E -4	4.525 E -6	4.525 E -6	-4.37 E -11	4.525 E -6	1.633
102400	1.60 E -4	1.297 E -6	1.297 E -6	-2.93 E -12	1.297 E -6	6.777
409600	7.98 E -5	3.491 E -7	3.492 E -7	-2.35 E -13	3.492 E -7	28.272
convergence		sdr vs. N : -0.970 , sdr vs. t : -0.939				

Table 5.3: Example 3, Weighted Uniform Sampling.

Importance Sampling / Rej Meth, (variance= 2.57 E -3)				
	Pseudorandom, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.27 E -3	1.378 E -3	1.378 E -3	0.085
6400	6.34 E -4	6.788 E -4	6.899 E -4	0.340
25600	3.18 E -4	3.112 E -4	3.175 E -4	1.358
102400	1.59 E -4	1.503 E -4	1.510 E -4	5.433
409600	7.93 E -5	7.995 E -5	8.090 E -5	21.728
convergence	sdr vs. N : -0.498 , sdr vs. t : -0.499			
	Halton Quasi-Random, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.27 E -3	3.675 E -4	3.692 E -4	0.167
6400	6.34 E -4	1.695 E -4	1.695 E -4	0.705
25600	3.17 E -4	1.134 E -4	1.134 E -4	2.972
102400	1.59 E -4	1.963 E -5	1.968 E -5	12.446
409600	7.93 E -5	1.176 E -5	1.176 E -5	52.157
convergence	sdr vs. N : -0.645 , sdr vs. t : -0.622			

Table 5.4: Example 3, Importance Sampling with the Rejection Method.

Impo Sampl /	Smooth Rej, $\delta = .2$, (variance= 2.57 E -3)			
	Pseudorandom, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.27 E -3	1.282 E -3	1.282 E -3	0.100
6400	6.34 E -4	6.482 E -4	6.459 E -4	0.400
25600	3.17 E -4	2.842 E -4	2.913 E -4	1.597
102400	1.58 E -4	1.495 E -4	1.502 E -4	6.397
409600	7.93 E -5	7.732 E -5	7.876 E -5	25.583
convergence	sdr vs. N : -0.504 , sdr vs. t : -0.504			

	Halton Quasi-Random, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.27 E -3	2.552 E -4	2.552 E -4	0.184
6400	6.34 E -4	5.810 E -5	5.811 E -5	0.781
25600	3.17 E -4	1.875 E -5	1.876 E -5	3.247
102400	1.59 E -4	5.668 E -6	5.669 E -6	13.479
409600	7.93 E -5	2.396 E -6	2.396 E -6	56.286
convergence	sdr vs. N : -0.852 , sdr vs. t : -0.826			

Table 5.5: Example 3, Importance Sampling with Smoothed Rejection.

Crude Monte Carlo, (variance= 5.41 E -2)

	Pseudorandom, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	5.82 E -3	5.553 E -3	5.561 E -3	0.021
6400	2.90 E -3	3.260 E -3	3.263 E -3	0.083
25600	1.45 E -3	1.526 E -3	1.534 E -3	0.327
102400	7.27 E -4	7.536 E -4	7.673 E -4	1.317
409600	3.63 E -4	4.121 E -4	4.150 E -4	5.274
convergence	sdr vs. N : -0.492 , sdr vs. t : -0.494			
	Halton Quasi-Random, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	5.82 E -3	2.205 E -4	2.205 E -5	0.049
6400	2.91 E -3	5.097 E -5	5.101 E -5	0.205
25600	1.45 E -3	1.662 E -5	1.662 E -5	0.892
102400	7.27 E -4	4.154 E -6	4.154 E -6	3.731
409600	3.63 E -4	9.937 E -7	9.939 E -7	15.792
convergence	sdr vs. N : -0.971 , sdr vs. t : -0.927			

Table 5.6: Example 4, Crude Monte Carlo.

Weighted Uniform Sampling, (variance= 2.55 E -3)

	Pseudorandom					
N	$sd(N)$	$sdr(N)$	$sdr^*(N)$	$b^*(N)$	$rmse(N)$	$t(N)$
1600	1.26 E -3	1.233 E -3	1.231 E -3	1.11 E -5	1.233 E -3	0.029
6400	6.31 E -4	7.186 E -4	7.188 E -4	3.88 E -6	7.214 E -4	0.115
25600	3.16 E -4	3.117 E -4	3.117 E -4	7.41 E -7	3.123 E -4	0.463
102400	1.58 E -4	1.773 E -4	1.775 E -4	2.50 E -7	1.784 E -4	1.892
409600	7.89 E -5	8.165 E -5	8.164 E -5	5.28 E -8	8.248 E -5	7.546
convergence		sdr vs. N : -0.502 , sdr vs. t : -0.501				

	Halton Quasi-Random					
N	$sd(N)$	$sdr(N)$	$sdr^*(N)$	$b^*(N)$	$rmse(N)$	$t(N)$
1600	1.26 E -3	5.753 E -5	5.753 E -5	1.58 E -8	5.753 E -5	0.059
6400	6.32 E -4	9.936 E -6	9.936 E -6	1.50 E -10	9.938 E -6	0.244
25600	3.16 E -4	4.896 E -6	4.896 E -6	1.41 E -10	4.896 E -6	1.055
102400	1.58 E -4	1.076 E -6	1.076 E -6	7.34 E -12	1.076 E -6	4.339
409600	7.89 E -5	2.530 E -7	2.535 E -7	3.75 E -13	2.531 E -7	18.138
convergence		sdr vs. N : -0.978 , sdr vs. t : -0.947				

Table 5.7: Example 4, Weighted Uniform Sampling.

Importance Sampling / Rej Meth, (variance= 1.65 E -3)				
Pseudorandom				
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.02 E -3	9.828 E -4	9.843 E -4	0.064
6400	5.08 E -4	5.384 E -4	5.417 E -4	0.277
25600	2.54 E -4	2.108 E -4	2.192 E -4	1.077
102400	1.27 E -4	1.239 E -4	1.250 E -4	4.348
409600	6.35 E -5	6.366 E -5	6.387 E -5	17.251
convergence	sdr vs. N : -0.500 , sdr vs. t : -0.498			
Halton Quasi-Random				
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.02 E -3	2.194 E -4	2.194 E -4	0.163
6400	5.08 E -4	9.337 E -5	9.342 E -5	0.701
25600	2.54 E -4	3.774 E -5	3.791 E -5	2.959
102400	1.27 E -4	1.882 E -5	1.891 E -5	12.259
409600	6.35 E -5	6.727 E -6	6.730 E -6	52.028
convergence	sdr vs. N : -0.622 , sdr vs. t : -0.597			

Table 5.8: Example 4, Importance Sampling with the Rejection Method.

Impo Sampl / Smooth Rej, $\delta = .2$, (variance= 1.65 E -3)				
Pseudorandom				
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.02 E -3	9.377 E -4	9.394 E -4	0.082
6400	5.08 E -4	4.879 E -4	4.894 E -4	0.338
25600	2.54 E -4	2.000 E -4	2.076 E -4	1.321
102400	1.27 E -4	1.188 E -4	1.201 E -4	5.276
409600	6.35 E -5	6.152 E -5	6.166 E -5	21.058
convergence	sdr vs. N : -0.495 , sdr vs. t : -0.496			
Halton Quasi-Random				
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.02 E -3	9.518 E -5	9.520 E -5	0.181
6400	5.08 E -4	2.558 E -5	2.559 E -5	0.761
25600	2.54 E -4	7.863 E -6	7.869 E -6	3.190
102400	1.27 E -4	2.094 E -6	2.094 E -6	13.282
409600	6.35 E -5	6.282 E -7	6.284 E -7	56.037
convergence	sdr vs. N : -0.906 , sdr vs. t : -0.871			

Table 5.9: Example 4, Importance Sampling with Smoothed Rejection.

Crude Monte Carlo, (variance= 2.24 E -1)

	Pseudorandom, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.18 E -2	1.251 E -2	1.251 E -2	0.049
6400	5.92 E -3	5.873 E -3	5.876 E -3	0.200
25600	2.96 E -3	2.944 E -3	2.951 E -3	0.774
102400	1.48 E -3	1.344 E -3	1.348 E -3	3.107
409600	7.39 E -4	6.851 E -4	6.885 E -4	11.496
convergence	sdr vs. N : -0.518 , sdr vs. t : -0.518			
	Halton Quasi-Random, 75 trials			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.18 E -2	5.274 E -4	5.274 E -4	0.092
6400	5.92 E -3	1.428 E -4	1.428 E -4	0.397
25600	2.96 E -3	3.190 E -5	3.190 E -5	1.673
102400	1.48 E -3	9.729 E -6	9.730 E -6	6.981
409600	7.39 E -4	2.236 E -6	2.236 E -6	29.317
convergence	sdr vs. N : -0.981 , sdr vs. t : -0.937			

Table 5.10: Example 5, Crude Monte Carlo.

Weighted Uniform Sampling, (variance= 4.06 E -5)

Pseudorandom						
N	$sd(N)$	$sdr(N)$	$sdr^*(N)$	$b^*(N)$	$rmse(N)$	$t(N)$
1600	1.58 E -4	1.763 E -4	1.758 E -4	2.21 E -6	1.773 E -4	0.063
6400	7.98 E -5	8.202 E -5	8.203 E -5	4.44 E -7	8.203 E -5	0.253
25600	3.99 E -5	4.308 E -5	4.308 E -5	1.10 E -7	4.311 E -5	1.001
102400	1.99 E -5	1.988 E -5	1.988 E -5	2.40 E -8	2.015 E -5	4.042
409600	9.96 E -6	9.879 E -6	9.879 E -6	4.49 E -9	9.912 E -6	16.091
convergence		sdr vs. N : -0.489 , sdr vs. t : -0.490				

Halton Quasi-Random						
N	$sd(N)$	$sdr(N)$	$sdr^*(N)$	$b^*(N)$	$rmse(N)$	$t(N)$
1600	1.59 E -4	1.292 E -5	1.291 E -5	4.48 E -9	1.292 E -5	0.094
6400	7.97 E -5	3.747 E -6	3.747 E -6	3.02 E -10	3.748 E -6	0.392
25600	3.98 E -5	1.084 E -6	1.084 E -6	2.21 E -11	1.084 E -6	1.614
102400	1.99 E -5	2.740 E -7	2.724 E -7	1.46 E -12	2.733 E -7	6.744
409600	9.96 E -6	6.855 E -8	6.673 E -8	6.59 E -14	6.603 E -8	27.962
convergence		sdr vs. N : -0.971 , sdr vs. t : -0.944				

Table 5.11: Example 5, Weighted Uniform Sampling.

Importance Sampling / Rej Meth, (variance= 2.86 E -5)				
Pseudorandom				
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.34 E -4	1.183 E -4	1.185 E -4	0.186
6400	6.65 E -5	7.520 E -5	7.526 E -5	0.754
25600	3.33 E -5	3.574 E -5	3.574 E -5	2.993
102400	1.66 E -5	1.608 E -5	1.619 E -5	12.052
409600	8.32 E -6	8.697 E -6	8.740 E -6	48.217
convergence	sdr vs. N : -0.502 , sdr vs. t : -0.497			
Halton Quasi-Random				
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.33 E -4	6.711 E -5	6.712 E -5	0.386
6400	6.66 E -5	1.859 E -5	1.861 E -5	1.642
25600	3.33 E -5	6.063 E -6	6.134 E -6	6.877
102400	1.66 E -5	2.766 E -6	2.766 E -6	28.805
409600	8.32 E -6	1.105 E -6	1.107 E -6	119.906
convergence	sdr vs. N : -0.705 , sdr vs. t : -0.679			

Table 5.12: Example 5, Importance Sampling with the Rejection Method.

Impo Sampl / Smooth Rej, $\delta = .2$ (variance= 2.84 E -5)				
	Pseudorandom			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.34 E -4	1.178 E -4	1.180 E -4	0.211
6400	6.65 E -5	6.524 E -5	6.543 E -5	0.853
25600	3.33 E -5	3.172 E -5	3.173 E -5	3.416
102400	1.66 E -5	1.454 E -5	1.457 E -5	13.677
409600	8.32 E -6	8.031 E -6	8.036 E -6	54.727
convergence	sdr vs. N : -0.505 , sdr vs. t : -0.505			
	Halton Quasi-Random			
N	$sd(N)$	$sdr(N)$	$rmse(N)$	$t(N)$
1600	1.33 E -4	3.416 E -5	3.417 E -5	0.417
6400	6.66 E -5	8.008 E -6	8.009 E -6	1.759
25600	3.33 E -5	1.754 E -6	1.754 E -6	7.279
102400	1.66 E -5	8.315 E -7	8.317 E -7	30.470
409600	8.32 E -6	1.593 E -7	1.600 E -7	126.505
convergence	sdr vs. N : -0.883 , sdr vs. t : -0.854			

Table 5.13: Example 5, Importance Sampling with Smoothed Rejection.

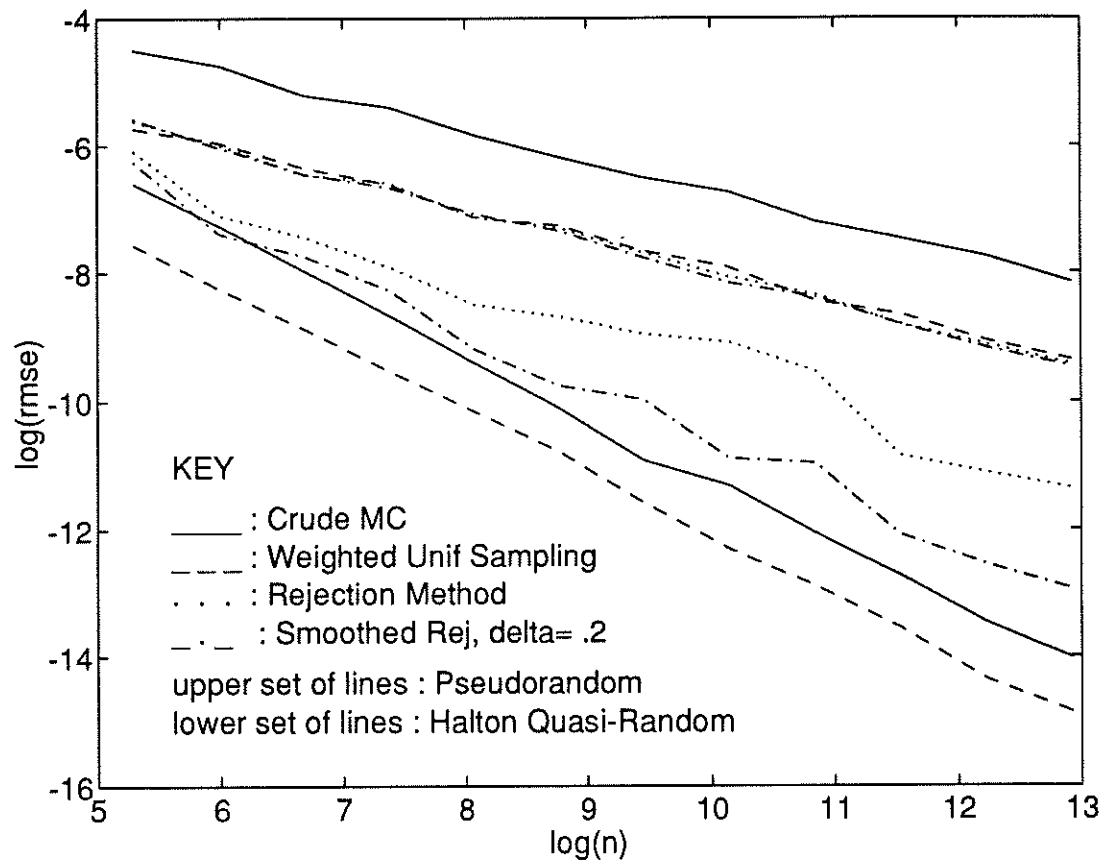


Figure 5.4: Log-Log Plot for Example 3.

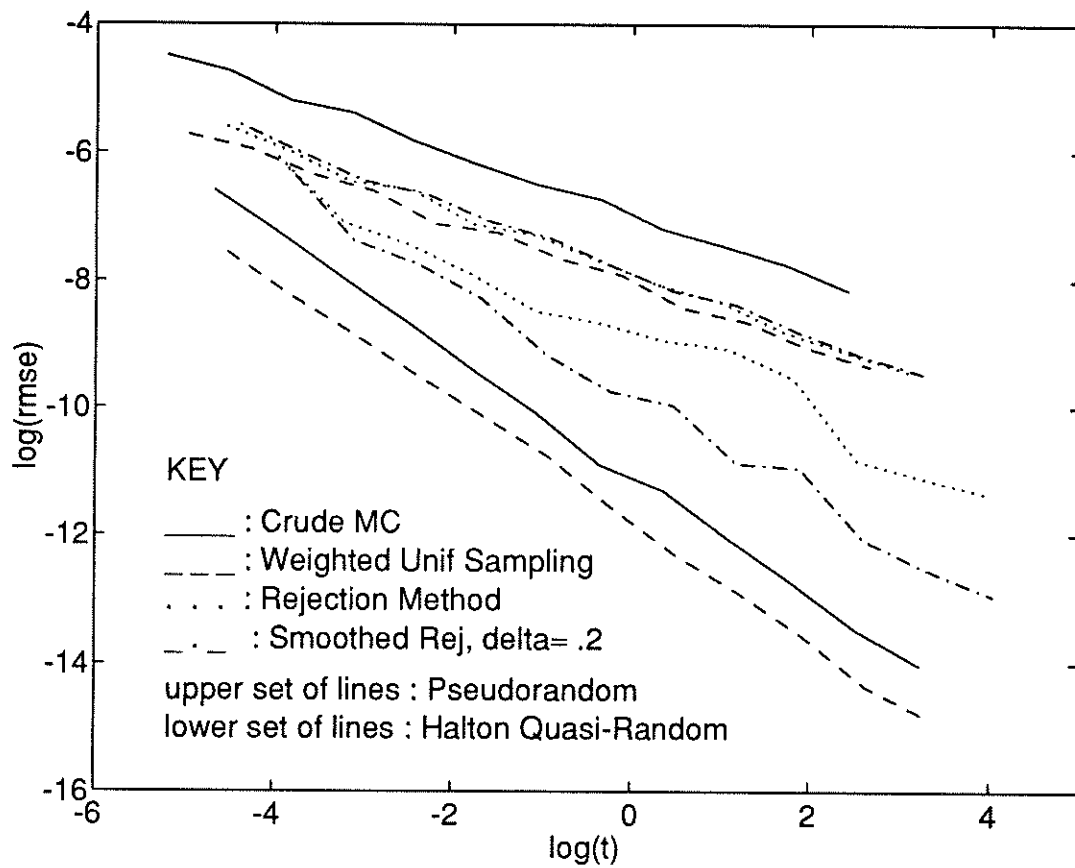


Figure 5.5: Log-Log Plot vs $t(\text{sec})$ for Example 3.

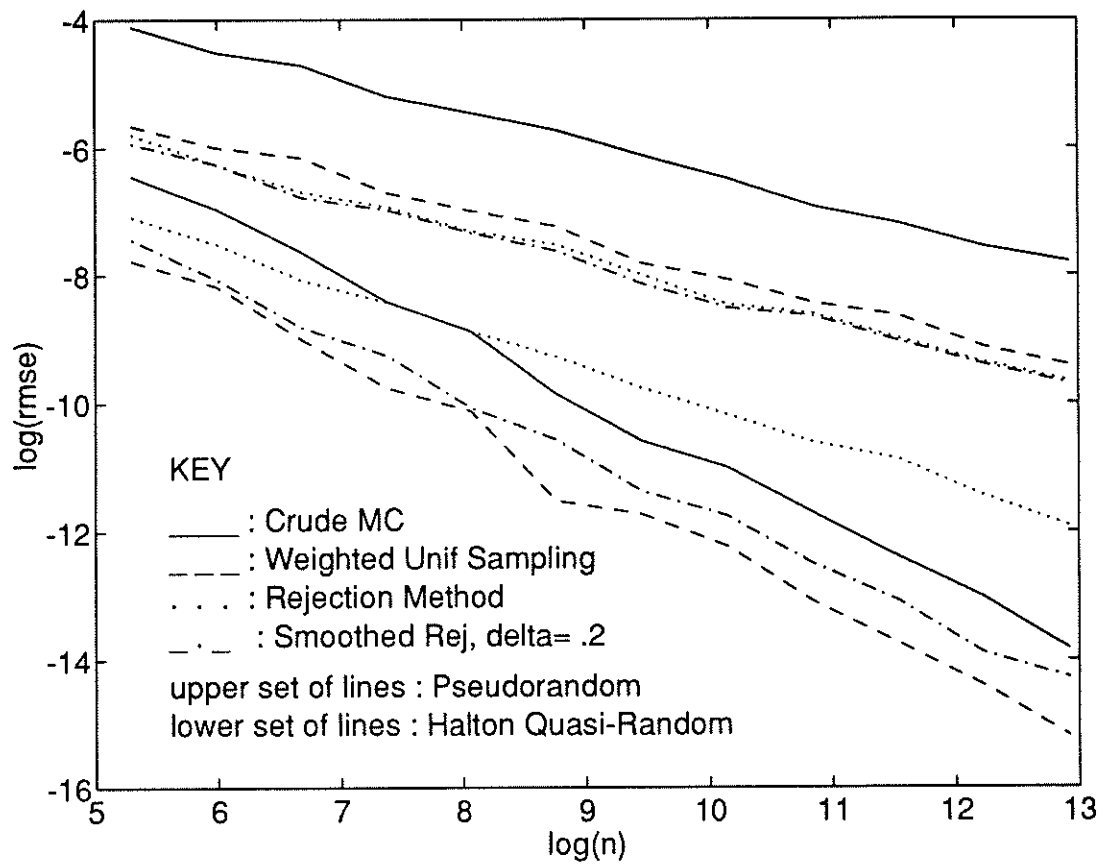


Figure 5.6: Log-Log Plot for Example 4.

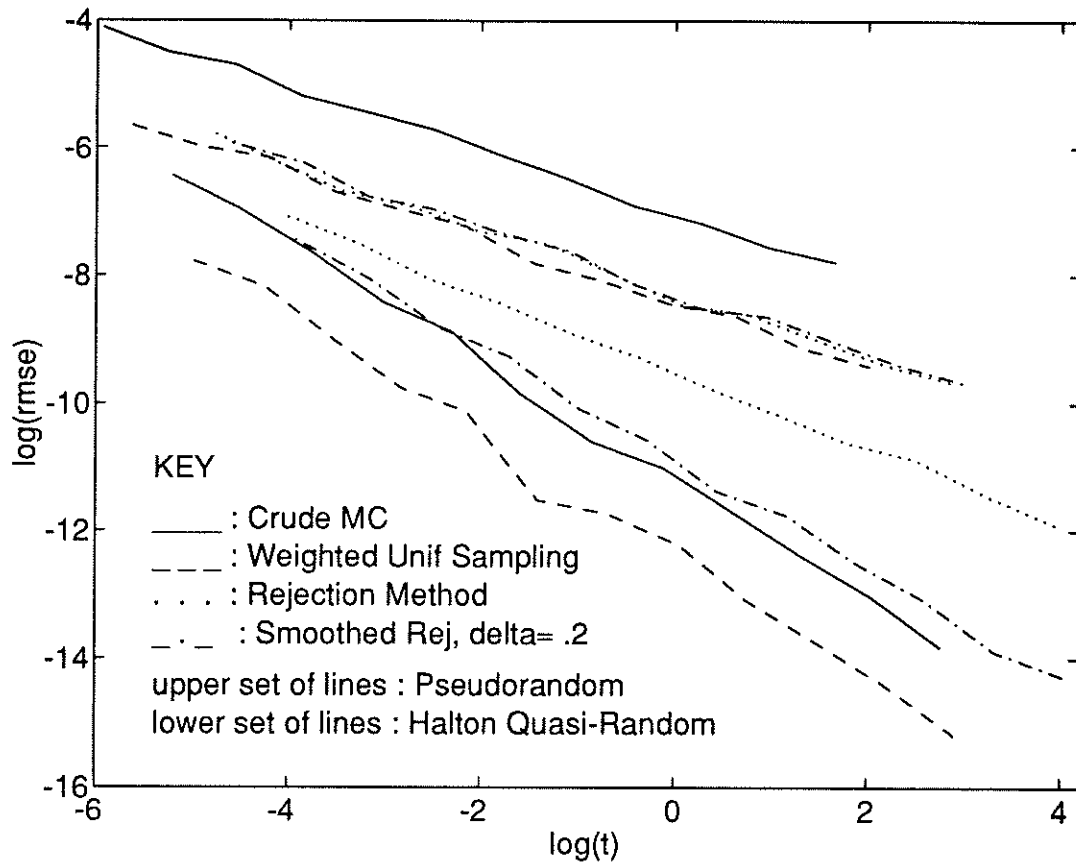


Figure 5.7: Log-Log Plot vs $t(\text{sec})$ for Example 4.

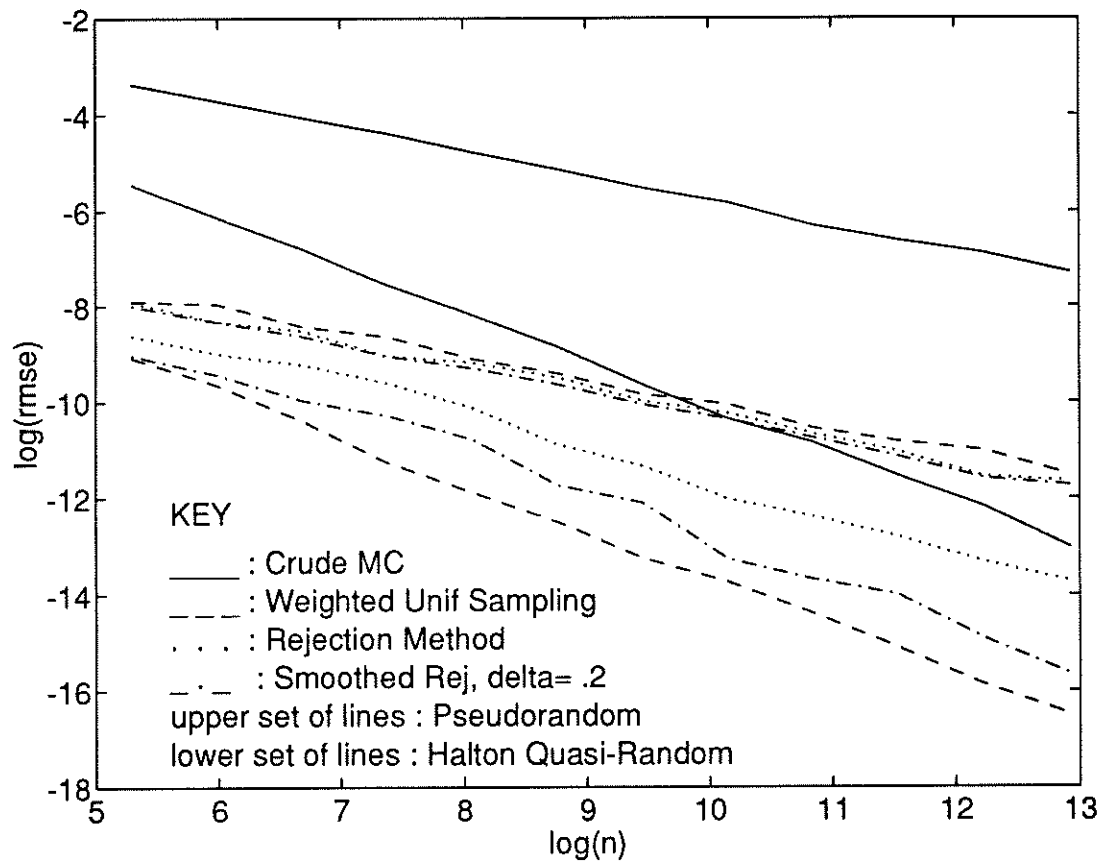


Figure 5.8: Log-Log Plot for Example 5.

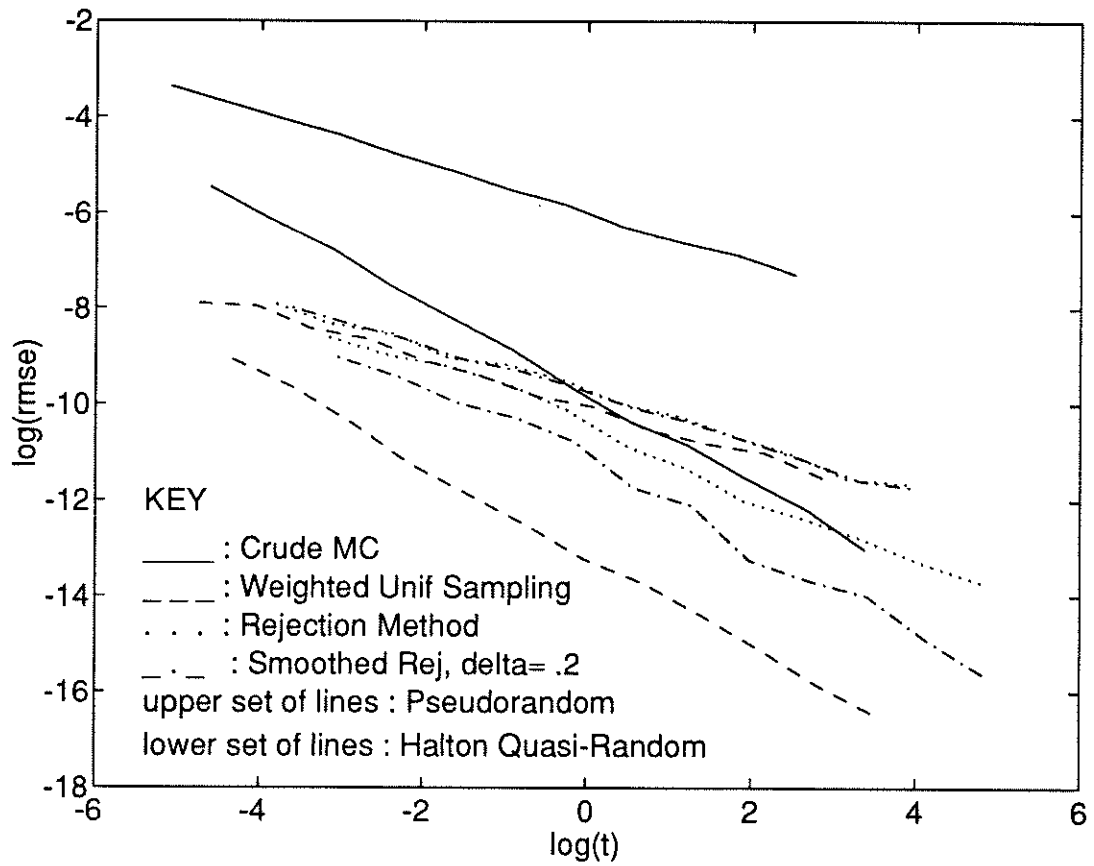


Figure 5.9: Log-Log Plot vs $t(\text{sec})$ for Example 5.

5.5 Discussion of Results, Conclusions

The computational results are not surprising. For the most part the estimators all perform as predicted in the discussions earlier in this chapter. The use of quasi-random (Halton) sequences clearly improves virtually every estimate, even when the extra time needed to generate the Halton sequence points is factored in.

The Rejection Method is clearly the worst option when quasi-random sequences are used.

On the other hand, with regard to Smoothed Rejection, this is always better than the ordinary Rejection Method with quasi-random points, but it is not too unusual to find that Crude Monte Carlo with quasi-random points actually works better when the amount of variance reduction is not very large. This happens because Crude Monte Carlo is often ‘smoother’ than Smoothed Rejection (less variation), also Crude Monte Carlo involves one less dimension than Smoothed Rejection. Both of these things can overcome the variance reduction, unless it is very large. Example 5 shows a case in which the variance reduction is large enough so that Smoothed Rejection outperforms Crude Monte Carlo at the levels of N considered.

Finally, Weighted Uniform Sampling is clearly the best choice in every example. It preserves most of the variance reduction of the Rejection Methods, but without adding any dimensions nor any significant increases in the variation.

My conclusion is the following: Use Weighted Uniform Sampling whenever

possible with quasi-random Monte Carlo estimates, unless for some reason Importance Sampling reduces the variance significantly more, in which case Smoothed Rejection may be preferred.

Although Weighted Uniform Sampling is biased, it is clear in the examples that, as expected, this bias goes to zero far more quickly than the mean square error, so it does not become a problem, even when quasi-random sequences are used. The theoretical justification for this comes from Equation 4.13.

The results here should make it clear that significant gains can be made by considering the discontinuities introduced by methods such as the Rejection Method. When these discontinuities are avoided, Monte Carlo estimates using quasi-random sequences often show a marked improvement in accuracy, thereby justifying the use of quasi-random sequences in many situations where preliminary results may have been disappointing precisely because of the discontinuities present.

CHAPTER 6

Reduction of Dimensions

6.1 Ineffectiveness of Quasi-Random Sequences for High Dimensions

Another factor besides continuity which strongly affects the success of quasi-random sequences when they are applied to Monte Carlo methods is the number of dimensions. As the number of dimensions increases, the effectiveness of quasi-random sequences is diminished. There is no specific number of dimensions above which quasi-random sequences are not recommended, since this also depends on the continuity of the integrand and the number of points, N , in the sequences that are used. However, it appears that at practical values of N , say up to the millions, the cutoff is somewhere around 30 dimensions. In other words, for Monte Carlo methods which involve approximation of higher than 30 dimensional integrands, quasi-random sequences will often not provide any improvement over pseudorandom sequences, and in some cases will actually give less accurate results.

The reasons for this have been studied by others in some depth. (See for example Morokoff [41].) They can be summarized by saying that for a high number of dimensions, it takes a very large value of N for the quasi-random points to fully

cover the entire integration region. This leads to the existence of large subregions containing no quasi-random points when N is not sufficiently high, and, as a result, large integration errors. When the number of dimensions is above 30, the values of N at which such problems disappear are beyond the millions. Note that this does not imply that the method fails to converge to the correct solution when quasi-random sequences are used in such cases. It simply means that at realistic values of N the errors obtained using quasi-random sequences can be higher than those obtained using pseudorandom sequences.

Another way to examine this is in terms of the Koksma-Hlawka Inequality from Chapter 3. A comparison of the asymptotic discrepancies of pseudorandom and quasi-random sequences (see Table 3.1) indicates that while the discrepancy of pseudorandom sequences is largely independent of the number of dimensions, for quasi-random sequences there is a factor $(\log N)^d$. At moderate dimensions, this factor does not seem to play a large role, but as the number of dimensions increases, the impact of this factor becomes more apparent and destructive. In this way, the Koksma-Hlawka Inequality provides some theoretical justification for the idea of reducing the dimensions when applying quasi-random sequences to Monte Carlo methods.

The emphasis of my work has been on finding ways to reduce the effective number of dimensions in applications, so that, as discussed above, quasi-random sequences will be better able to accelerate convergence rates. In doing so, I have found three approaches to be effective.

A first approach which has proven useful, and is usually trivial to apply, involves reordering the dimensions so the dimensions involving the greatest amount of variation or variance in the integrand are the lowest dimensions. When such a switch is made, there is virtually no change in the performance of pseudorandom sequences, but often a significant improvement in the performance of quasi-random sequences. In the examples in Chapter 5, I was careful to follow this principle. In the next section, I will look at the negative effects of ignoring it.

A second approach is to seek new or modified algorithms which have the effect of concentrating as much of the variance or variation as possible in the lowest dimensions. This approach is similar to the first approach, but can involve a great deal more work. To study such an approach, I have examined its usefulness with the Feynman-Kac Formula, and obtained significant improvements with quasi-random sequences. This is covered in Section 6.3.

Finally, a third alternative is to combine pseudorandom and quasi-random sequences, taking advantage of each of their benefits. This approach offers the possibility of applying quasi-random sequences to improve the accuracy of some very high dimensional problems. In Chapter 7, the effectiveness of this approach will be demonstrated for the case of Diffusion Monte Carlo.

An important note is that it is not clear whether variance or variation, or something else entirely, is the proper measure of the ‘variations of the integrand’ when using quasi-random sequences. This is an open question in the field of quasi-random sequences. However, in practice, it is often fairly clear that one dimension

is accounting for more of the variability than another. Since variance is often easiest to estimate I have tended to use that, with successful results.

6.2 Computational Examples: Reordered Dimensions

Here, I will present a few examples in which the effects of increasing the number of dimensions (or equivalently, making the higher dimensions account for a greater amount of variation) are studied. In all the examples, it will be observed that increasing the dimensions has relatively little impact when pseudorandom sequences are used, but has a negative impact when quasi-random sequences are used. The flip side of this is that by reducing the dimensions (or concentrating more of the variations in the lower dimensions), one can improve the performance of quasi-random sequences with Monte Carlo methods.

Example 6 *Compare the results of approximating these two equal integrals using pseudorandom and Halton points:*

$$A = \int \cdots \int_{I^d} 5 \cos \left(4(x_4 + x_5 + x_6) + \frac{(x_1 + x_2 + x_3)}{5} \right) dx_1 \dots dx_6$$

$$B = \int \cdots \int_{I^d} 5 \cos \left(4(x_1 + x_2 + x_3) + \frac{(x_4 + x_5 + x_6)}{5} \right) dx_1 \dots dx_6$$

These are of course equivalent, but the first has more of its variation contained in the higher dimensions. Therefore we should expect that using Halton points will

be more effective when estimating B than when estimating A . On the other hand, if we use pseudorandom points both cases should be roughly the same. The results are shown in Figure 6.1.

Example 7 In Example 3 of Chapter 5, we computed Monte Carlo estimates of the integral over I^5 of the following function:

$$f(\mathbf{x}) = \exp \left(\sum_{i=1}^5 a_i x_i^2 \frac{2 + \sin \left(\sum_{j=1, j \neq i}^5 x_j \right)}{2} \right)$$

where $\mathbf{a} = (1, \frac{1}{2}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$.

Now, I will compare this with the case where $\mathbf{a} = (\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{2}, 1)$, reversing the dimensions. By symmetry, the integral of this function is equal to the integral of the function above. However, more of the variation of this new function is contained in the higher dimensions. As a result, the Monte Carlo estimates involving quasi-random sequences should not be as accurate for the new function as they were for the original function in Example 3. Results are shown in Table 6.1 and Figure 6.2.

Example 8 Consider the Monte Carlo integration of the following two equivalent functions in 15 dimensions:

$$\begin{aligned} f_L(\mathbf{x}) &= \left(\sum_{i=1}^{15} \frac{16-i}{15} x_i \right)^3 \\ f_H(\mathbf{x}) &= \left(\sum_{i=1}^{15} \frac{i}{15} x_i \right)^3 \end{aligned}$$

Crude Monte Carlo, *sdr* (approximating *rmse*)

N	Original Function		Reversed Dimensions	
	Pseudorand	Halton	Pseudorand	Halton
1600	4.538 E -3	1.727 E -4	4.504 E -3	3.969 E -4
6400	2.064 E -3	4.142 E -5	2.150 E -3	8.603 E -5
25600	1.189 E -3	1.229 E -5	1.157 E -3	2.928 E -5
102400	5.723 E -4	2.977 E -6	4.692 E -4	5.413 E -6
409600	2.873 E -4	8.086 E -7	3.082 E -4	1.918 E -6
conv rates	-0.480	-0.980	-0.505	-0.972

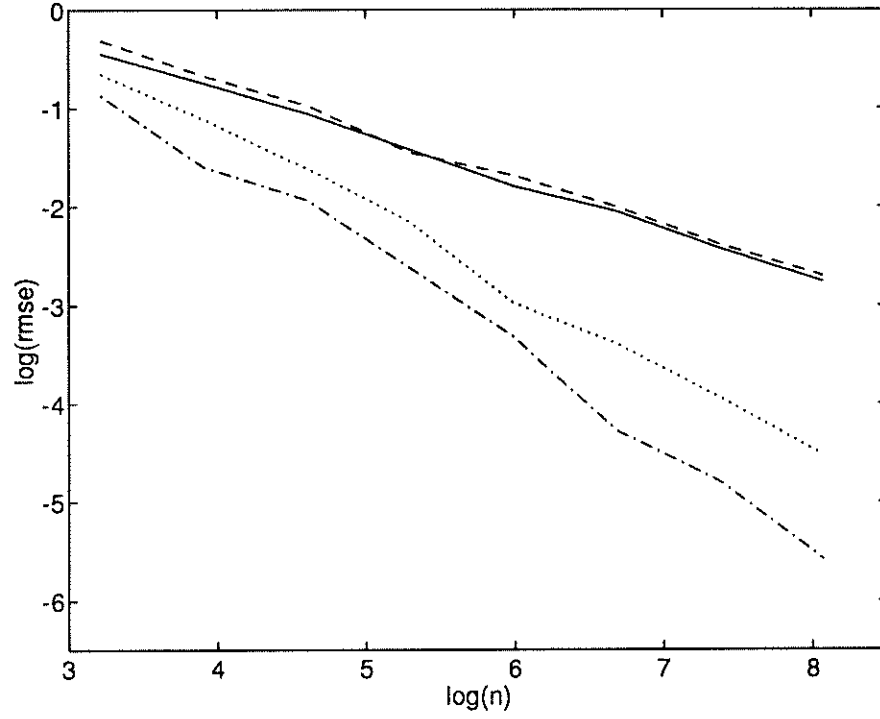
Table 6.1: Results for Example 7, 75 trials

over the domain $D = [-.5, .5]^{15}$.

The integral of each of these functions is exactly zero, since both functions are odd. Monte Carlo estimates are computed by simply subtracting $\frac{1}{2}$ from each component of the pseudorandom or quasi-random points to move them into the required domain, and then using a Crude Monte Carlo estimate.

By symmetry, the two functions above are clearly equivalent and pseudorandom estimates of the two integrals should be similar. However, the first one has more of its variations in the lower dimensions, while the second is the reverse. Therefore, as in the previous examples, we expect quasi-random sequences to produce more accurate Monte Carlo estimates for the integral of the first function, f_L , than for the second, f_H . The results are presented in Table 6.2 and Figure 6.3.

Examining the results for these examples, the quasi-random Monte Carlo estimates are consistently worse in the “higher dimensional” cases. It is somewhat surprising that the drop in convergence is so small in Example 7. I would at-



Integrand	Sequence	Slope of $rmse$ vs. N
A	pseudorandom	-0.479 (solid)
B	pseudorandom	-0.490 (dashed)
A	Halton	-0.814 (dotted)
B	Halton	-0.974 (dot-dash)

Figure 6.1: Convergence Results and Log-Log Plot for Example 6, 100 trials.

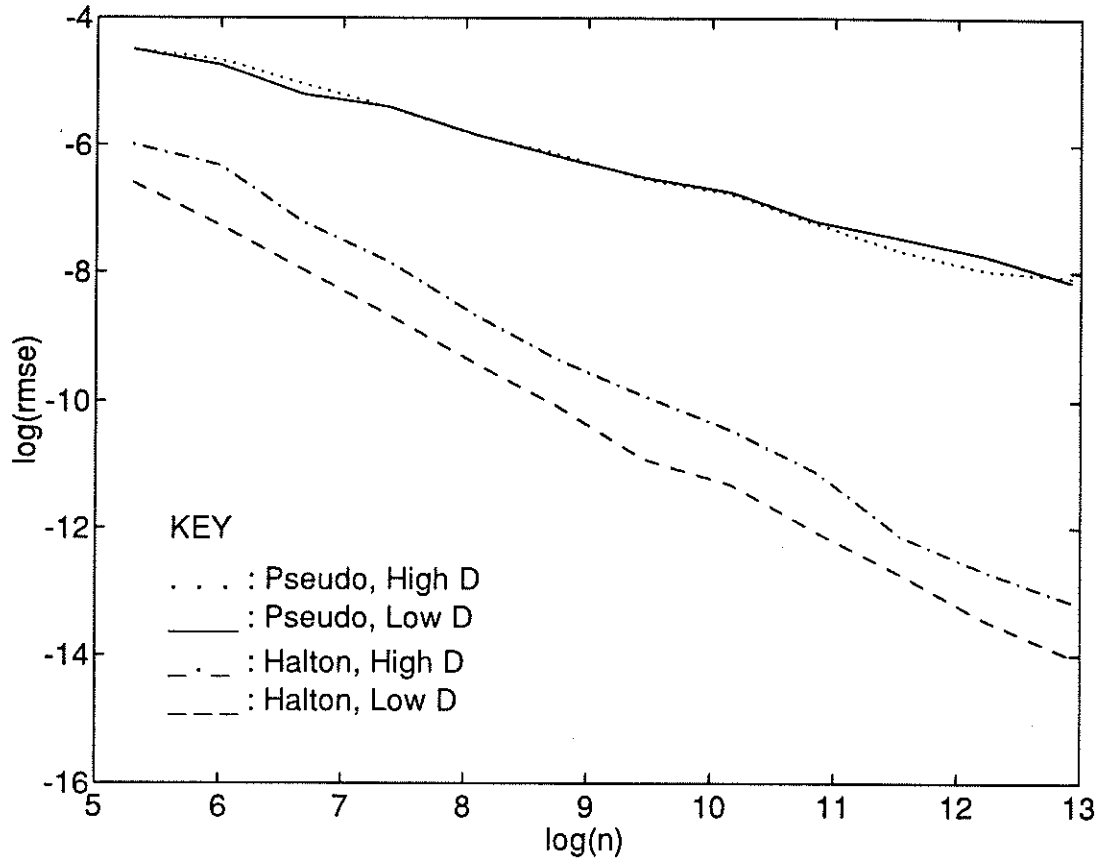


Figure 6.2: Log-Log Plot for Example 7

Crude Monte Carlo, sdr (approximating $rmse$)				
N	$f_L(x)$		$f_H(x)$	
	Pseudorand	Halton	Pseudorand	Halton
50	1.684 E -1	1.062 E -1	1.514 E -1	1.825 E -1
200	8.725 E -2	4.024 E -2	7.877 E -2	7.201 E -2
800	3.900 E -2	1.611 E -2	3.842 E -2	3.452 E -2
3200	1.895 E -2	5.573 E -3	2.117 E -2	1.588 E -2
12800	1.010 E -2	1.309 E -3	1.008 E -2	7.314 E -3
51200	4.776 E -3	3.139 E -4	4.796 E -3	1.429 E -3
conv rates	-0.511	-0.821	-0.502	-0.628

Table 6.2: Results for Example 8, 75 trials

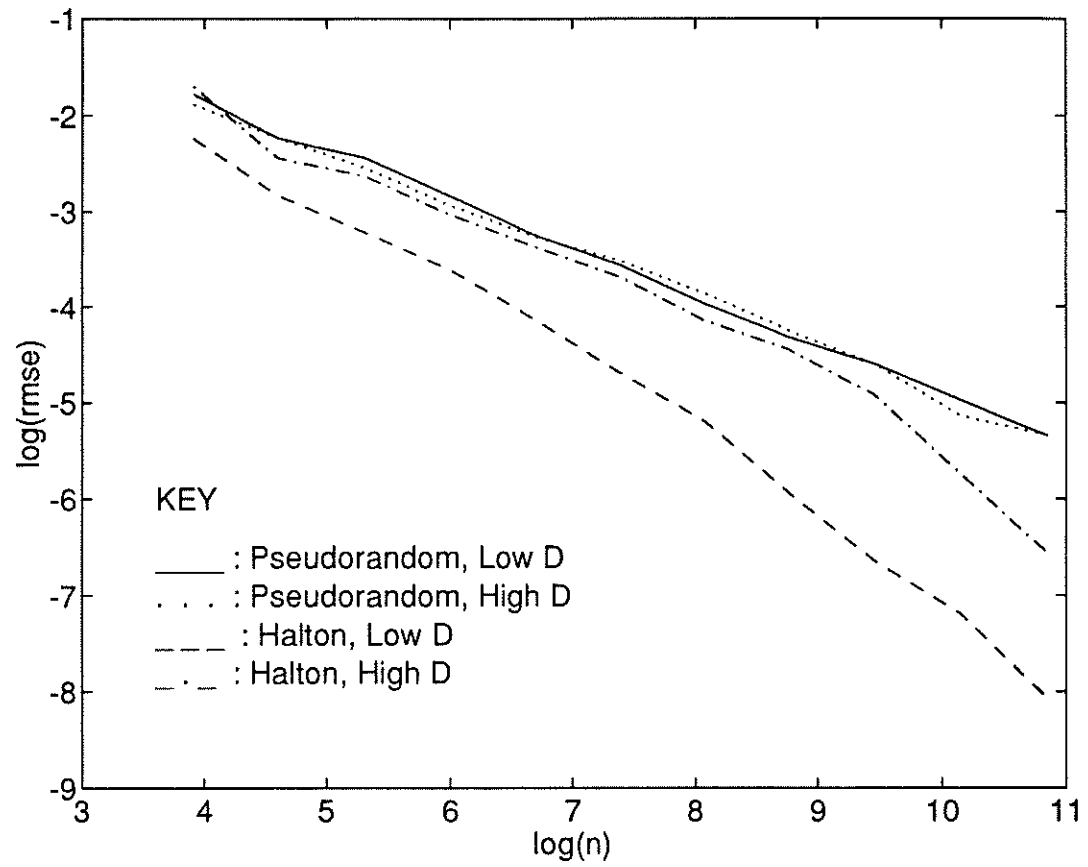


Figure 6.3: Log-Log Plot for Example 8, 75 trials

tribute this to a number of factors including the relatively low number, 5, of total dimensions, the smoothness and flatness of the functions being integrated, and the relatively high values of N used. The results for Example 8 are more extreme because a much higher number of dimensions are involved and the functions are not nearly as flat as in Example 7.

One can see in Example 8 a typical feature of quasi-random sequences in high dimensions (see Morokoff [41]), which is that they initially do not converge any better than pseudorandom until they reach some threshold of N , above which they start to exhibit accelerated convergence. The higher the number of dimensions, in general, the higher is the threshold for N . At greater than roughly 30 dimensions, this threshold becomes unreachable in most practical examples. This could also explain the results in Example 7 – where the threshold is apparently a very low value of N .

The important point for my purposes is that in all of the examples, the quasi-random Monte Carlo estimates are more accurate when the dimensions are shifted so that the lowest dimensions account for the largest share of the variations in the functions being integrated. This is the key property which will be used to improve the quasi-random Monte Carlo estimates for the Feynman-Kac Formula in the next section.

6.3 Stochastic Process Simulation: Feynman-Kac Formula

The Feynman-Kac Formula provides a connection between the solution of linear parabolic differential equations and stochastic paths which is similar to the method of characteristics for solving hyperbolic differential equations. In this section, I will deal with the case when the stochastic paths are simple Wiener processes, which are mathematical idealizations of Brownian motion. For such cases, a reformulation of standard methods for simulating the sample paths allows for a larger amount of the variance of the paths to be concentrated in the lower dimensional components of the pseudorandom or quasi-random points used to generate them. As discussed in Section 6.1, this should improve the associated Monte Carlo estimates when quasi-random sequences of points are used, while having little effect in the case of pseudorandom sequences. This has in fact been observed, and a computational example will be examined in Section 6.4.

6.3.1 Stochastic Basics, Wiener Process

A stochastic process is a family of random variables, $\{X(t)\}$, defined on probability space $(\Omega, \mathcal{F}, \mathcal{P})$, where Ω is the domain for $X(t)$, \mathcal{F} is the σ -algebra of probability measurable subsets, and \mathcal{P} is the probability measure. The parameter t is typically interpreted as time and varies continuously over the set of non-negative real numbers. In this chapter, Ω will be the real line. Later on in Chapter 7, it

will be \Re^d , $d > 1$.

Stochastic processes with the Markov property are defined by the following:

1. an exact initial state, $\mathbf{X}(t_0)$, or probability distribution, $\pi(x, t_0)$, of starting positions at time t_0 .
2. a transition density function, $p(x, s, y, t) = p(\mathbf{X}(t) = y | \mathbf{X}(s) = x)$, indicating the conditional probability density of $\mathbf{X}(t)$ given $\mathbf{X}(s)$, $t \geq s$.

The Markov property allows for this simple form of the transition density by requiring that the probability density of $\mathbf{X}(t)$ given $\mathbf{X}(s)$, $t \geq s$, be independent of any earlier values $\mathbf{X}(r)$, $r < s$. In other words, the process has no memory, so only its current state affects later states. All of the stochastic processes discussed here will satisfy the Markov property unless specifically stated otherwise.

The following quantities are used to characterize stochastic processes:

$$\text{mean: } \mu(t) = E(\mathbf{X}(t))$$

$$\text{variance: } \sigma^2(t) = \text{var}(\mathbf{X}(t)) = E((\mathbf{X}(t) - \mu(t))^2)$$

$$\text{covariance: } c(s, t) = \text{cov}(\mathbf{X}(s), \mathbf{X}(t)) = E((\mathbf{X}(s) - \mu(s))(\mathbf{X}(t) - \mu(t)))$$

A *wide sense stationary* process is defined as satisfying the following conditions:

$$\mu(t) = \mu, \text{ constant}$$

$$\sigma^2(t) = c(0), \text{ constant}$$

$$c(s, t) = c(t - s)$$

For wide sense stationary processes, the *frequency distribution* is defined by assigning the ν -frequency component, for any $\nu \geq 0$, the following value:

$$S(\nu) = \int_{-\infty}^{\infty} c(t) \cos(2\pi \nu t) dt$$

Note that c is real-valued and even. By standard Fourier methods we then have the following:

$$\text{var}(X(t)) = c(0) = \int_{-\infty}^{\infty} S(\nu) d\nu$$

An important case is **White Noise**. In this unusual case $c(t) = S_0 \delta(t)$ and $\mu(t) = 0$, where $\delta(t)$ is the Dirac-delta function. We then have the following:

$$S(\nu) = \int_{-\infty}^{\infty} S_0 \delta(t) \cos(2\pi \nu t) dt = S_0, \quad \forall \nu$$

Therefore, this is known as “white noise” because of the analogy to white light with a uniform frequency spectrum. This (generalized) process has infinite variance. It will be denoted by $\Upsilon(t)$. Its properties are summarized as follows:

$$E(\Upsilon(t)) = 0 \tag{6.1}$$

$$\text{var}(\Upsilon(t)) = \infty \tag{6.2}$$

$$\text{cov}(\Upsilon(s), \Upsilon(t)) = \delta(s - t) \tag{6.3}$$

This is not really a process in the usual sense, because of the delta function, but it can be interpreted in terms of generalized functions and distribution theory. (See for example Kloeden and Platen [32, pg.44].) It will prove useful as the limiting case of Wiener intervals which will be described below.

The **Wiener Process**, denoted by $W(t)$, is an important stochastic process which is not wide sense stationary. It has initial condition $W(0) = 0$, and is defined by the following transition density function:

$$p_w(x, s, y, t) = \frac{1}{\sqrt{2\pi(t-s)}} e^{-\frac{(y-x)^2}{2(t-s)}} \quad (6.4)$$

This is a Normal (Gaussian) distribution in terms of y and t with x and s fixed, with mean x and variance $(t-s)$, which is denoted by $\mathcal{N}(x, t-s)$. Successive increments of the Wiener process are therefore Gaussian and independent (by the Markov property). The mean, variance, and covariance for the Wiener process are computed as follows:

$$\mu(t) = E(W(t)) = E(y, y \in N(0, t)) = 0 \quad (6.5)$$

$$\sigma^2(t) = \text{var}(W(t)) = \text{var}(y, y \in N(0, t)) = t \quad (6.6)$$

$$\begin{aligned} c(s, t) &= E((W(s) - \mu(s))(W(t) - \mu(t))) \\ &= E(W(s)W(t)) \\ &= E((W(t) - W(s) + W(s)) \cdot W(s)) \quad , \quad \text{assume } t \geq s. \\ &= E((W(t) - W(s))W(s)) + E(W(s)^2) \\ &= E(W(s)^2) = s \quad , \quad \text{for } t \geq s. \\ c(s, t) &= \min\{s, t\} \quad , \quad \text{in general.} \end{aligned} \quad (6.7)$$

Therefore its sample paths do not tend to favor any particular direction, but do tend to spread out from their starting point, in the following root mean square

sense:

$$\sqrt{E((W(t) - W(0))^2)} = \sqrt{t}$$

The sample paths are continuous in many senses. (See for example Kloeden and Platen [32, pg.38-39].) In the root mean square sense, we have the following:

$$\lim_{h \rightarrow 0} \sqrt{E([W(t+h) - W(t)]^2)} = \lim_{h \rightarrow 0} \sqrt{h} = 0$$

However, they are not differentiable in any traditional sense (Kloeden and Platen [32, pg.42], using the Law of Iterated Logarithms). In the root mean square sense we have the following:

$$\lim_{h \rightarrow 0} \frac{1}{h} \sqrt{E([W(t+h) - W(t)]^2)} = \lim_{h \rightarrow 0} h^{-1/2} = \infty$$

On the other hand, recent advances in the theory of stochastic processes have led to the development of the Ito Calculus (and also the Stratanovich Calculus) which provides a way to define a stochastic derivative of the Wiener process and use it to define stochastic differential equations (SDE's), which will be introduced in the next section.

In order to motivate such concepts, we now consider a new set of processes given by the following:

$$W_h(t) = \frac{W(t+h) - W(t)}{h}$$

The mean, variance, and covariance are computed as follows:

$$\mu_h(t) = E(W_h(t)) = 0$$

$$c_h(s, t) = E((W_h(s) - \mu_h(s))(W_h(t) - \mu_h(t)))$$

$$\begin{aligned}
&= E(W_h(s) W_h(t)) \\
&= \frac{1}{h^2} E((W(t+h) - W(t))(W(s+h) - W(s))) \\
&= 0, \text{ if } t \geq s+h, \text{ by the Markov property.} \\
&= \frac{1}{h^2} [E((W(s+h) - W(t))^2) \\
&\quad + E((W(t+h) - W(s+h))(W(s+h) - W(s))) \\
&\quad - E((W(t) - W(s))(W(t+h) - W(t)))] , \quad s \leq t < s+h \\
&= \frac{1}{h^2} E((W(s+h) - W(t))^2) = \frac{s+h-t}{h^2} \\
&= \frac{1}{h} \left(1 - \frac{t-s}{h}\right) , \quad s \leq t < s+h. \\
c_h(s, t) &= \max \left\{ 0, \frac{1}{h} \left(1 - \frac{|t-s|}{h}\right) \right\} , \text{ in general.} \\
\sigma_h^2(t) &= c_h(t, t) = \frac{1}{h}
\end{aligned}$$

These new processes are therefore wide sense stationary.

Consider the frequency distribution for W_h (Kloeden and Platen [32, pg.44,558]) :

$$\begin{aligned}
S_h(\nu) &= \int_{-\infty}^{\infty} c_h(t) \cos(2\pi\nu t) dt \\
&= \int_{-h}^h \frac{1}{h} \left(1 - \frac{|t|}{h}\right) \cos(2\pi\nu t) dt \\
&= \frac{2}{h} \int_0^h \left(1 - \frac{t}{h}\right) \cos(2\pi\nu t) dt \\
&= \frac{2}{h} \cdot \frac{1 - \cos(2\pi\nu h)}{4\pi^2\nu^2 h} \\
&= \left(\frac{\sin(\pi\nu h)}{\pi\nu h}\right)^2
\end{aligned}$$

Therefore, as $h \rightarrow 0$, $S(\nu) \rightarrow 1$, for every frequency, ν . This recalls white noise which was described earlier.

One can then formally write the stochastic derivative of the Wiener process as follows:

$$\frac{dW}{dt} \stackrel{def}{=} \lim_{h \rightarrow 0} \frac{W(t+h) - W(t)}{h} = \lim_{h \rightarrow 0} W_h(t) = \Upsilon(t)$$

This of course makes no sense in ordinary calculus since the Wiener paths are not differentiable. However, within the scope of distribution theory and the Ito Calculus (which will be briefly discussed in the next section) the above equalities can be rigorously defined and justified. (See Gard [15].)

If one lets $\Delta_h W(t) = W(t+h) - W(t)$ denote the finite Wiener interval, then the limiting case as $h \rightarrow 0$ is written $dW(t)$, in similar fashion to the relationship between Δt and dt . The limiting case, $dW(t)$, is known as the (Ito) stochastic differential. It is a fundamental quantity in the Ito Calculus, along with the deterministic differential dt . Two important properties of $\Delta_h W(t)$ are as follows:

$$E(\Delta_h W(t)) = 0 \tag{6.8}$$

$$\sqrt{E((\Delta_h W(t))^2)} = \sqrt{h} \tag{6.9}$$

These suggest that dW also has an expected value of zero and a ‘magnitude’ of size $(dt)^{1/2}$ in a generalized root mean square sense. In Section 7.5 I will look at this in more detail.

6.3.2 Stochastic Differential Equations, and the Ito Formula

Based on the discussion in the last section, we now write the following to describe a process starting at x_0 with Gaussian intervals like the Wiener process:

$$\begin{aligned}X(0) &= x_0 \\dX(t) &= dW\end{aligned}$$

This is an example of a very simple stochastic differential equation (SDE). It is interpreted to mean the following:

$$X(t) = x_0 + \int_0^t dW = x_0 + W(t) - W(0) = x_0 + W(t)$$

Therefore this is simply a Wiener process with a shifted starting point.

Less trivial cases require a definition for the integral with respect to dW above, which is not well-defined in ordinary calculus.

The Ito definition is as follows:

$$I[f(t)] = \int_0^t f(t) dW = \lim_{n \rightarrow \infty} \left[\frac{1}{n} \sum_{i=1}^n f(t_i) (W(t_{i+1}) - W(t_i)) \right]$$

where $\{t_i\}_{i=1}^n$ is a partition of the interval $[0, t]$. This definition of stochastic integrals is the basis for the Ito Calculus. (See Gard [15] for a full description.) A different definition leads to the alternative Stratanovich Calculus, which in some ways is more like ordinary calculus (the chain rule is the same), but it does not have as many useful statistical properties as the Ito calculus does (Gard [15]). I will be working solely within the Ito Calculus. Note that since W is a stochastic

process above, the integral is itself a stochastic process with no definite value, but rather a probability distribution of possible values at any particular time, t .

With this definition, a more general stochastic differential equation could take the following form:

$$X(0) = x_0$$

$$dX(t) = a(X(t), t) dt + b(X(t), t) dW$$

This defines a stochastic process with initial condition $X(0) = x_0$ and evolution governed by the following expression:

$$X(t) = x_0 + \int_0^t a(X(t), t) dt + \int_0^t b(X(t), t) dW$$

where the first integral is an ordinary integral and the second is defined above as an Ito integral. This equation allows one to compute the probability densities for X in terms of the functions a and b , the initial condition, x_0 , and the transition densities of the Wiener process.

When the functions a and b are measurable and satisfy a Lipschitz condition and linear growth condition, then the above SDE has a solution known as a diffusion process with local drift a and local diffusion rate $\sigma = b^2$. This will be discussed further in Chapter 7.

Here, and later in Chapter 7, we will be dealing with stochastic processes which are the result of applying a function to a known stochastic process. **Ito's formula** provides a way to find the stochastic differential equation governing such a process as follows:

Let $X(t)$ satisfy the following SDE:

$$X(0) = x_0$$

$$dX(t) = a(X(t), t) dt + b(X(t), t) dW$$

Then, $Y(t) = f(X(t), t)$ satisfies the following SDE:

$$Y(0) = f(x_0, 0)$$

$$dY(t) = \left(\frac{\partial f}{\partial t} + a \cdot \frac{\partial f}{\partial x} + \frac{1}{2} b^2 \cdot \frac{\partial^2 f}{\partial x^2} \right) dt + \left(b \cdot \frac{\partial f}{\partial x} \right) dW$$

This fundamental result is often referred to as the Chain Rule for the Ito Calculus. (Proofs of the Ito Formula are given by Kloeden and Platen [32], Gard [15], and Friedman [14].)

6.3.3 Proof of Feynman-Kac Formula, Conversion to Backwards Time

The Feynman-Kac Formula, to be proven here, allows one to express the solutions to linear parabolic differential equations as the expectations of functions of stochastic processes. This enables one to numerically solve such differential equations by simulating the associated stochastic processes, and then taking Monte Carlo estimates of their expectations.

I will not deal here with the most general case, but a simpler case involving only the Wiener process which will be sufficient for the examples in this chapter.

Theorem 2 (Feynman-Kac Formula) *Let $\xi_{x,t}$ represent a stochastic process with initial condition $\xi(t) = x$ and stochastic differential $d\xi = dW$. This is simply the Wiener process shifted to a new initial position and time.*

Let $u(x, t)$ be the solution to the following linear parabolic differential equation:

$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} + v(x, t) u = 0 \quad (6.10)$$

with FINAL condition $u(x, T) = f(x)$.

Then, the solution to the differential equation is given by the following formula, for all x and all $t \leq T$:

$$u(x, t) = E \left[f(\xi_{x,t}(T)) e^{\int_t^T v(\xi_{x,t}(s), s) ds} \right] \quad (6.11)$$

where the expectation is taken over all possible realizations of the stochastic process $\xi_{x,t}$.

Proof:

Let $u(x, t)$ be given by Equation 6.11, and let $E_{x,t}$ denote expectation with respect to $\xi_{x,t}$. Then,

$$\begin{aligned} u(x, t-h) &= E_{x,t-h} \left[f(\xi(T)) e^{\int_{t-h}^T v(\xi(s), s) ds} \right] \\ &= E_{x,t-h} \left\{ E_{\xi(t),t} \left[f(\xi(T)) e^{\int_t^T v(\xi(s), s) ds} \right] e^{\int_{t-h}^t v(\xi(s), s) ds} \right\} , \end{aligned}$$

using the Markov property.

$$= E_{x,t-h} \left[u(\xi(t), t) e^{\int_{t-h}^t v(\xi(s), s) ds} \right] , \text{ by substitution.}$$

$$u(x, t) = E_{x,t} [u(\xi(t), t)] , \text{ trivially.}$$

Next,

$$\begin{aligned}
\frac{\partial u}{\partial t} &\stackrel{\text{def}}{=} \lim_{h \rightarrow 0} \frac{u(x, t) - u(x, t - h)}{h} \\
&= \lim_{h \rightarrow 0} \frac{1}{h} \left\{ E_{x,t} [u(\xi(t), t)] - E_{x,t-h} \left[u(\xi(t), t) e^{\int_{t-h}^t v(\xi(s), s) ds} \right] \right\} \\
&= \lim_{h \rightarrow 0} \frac{1}{h} \left\{ E_{x,t} [u(\xi(t), t)] - E_{x,t} \left[u(\xi(t+h), t) e^{\int_{t-h}^t v(\xi(s+h), s) ds} \right] \right\}
\end{aligned}$$

Let $g(x) = u(x, t)$, with t fixed. Then,

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \lim_{h \rightarrow 0} \frac{1}{h} \left\{ E_{x,t} \left[g(\xi(t)) - g(\xi(t+h)) e^{\int_{t-h}^t v(\xi(s+h), s) ds} \right] \right\} \\
\frac{\partial u}{\partial t} &= \lim_{h \rightarrow 0} \frac{1}{h} \left\{ E_{x,t} \left[g(\xi(t)) - g(\xi(t+h)) \left(1 + \int_{t-h}^t v(\xi(s+h), s) ds + O(h^2) \right) \right] \right\}, \\
&\quad \text{after expanding the exponential.} \\
&= E_{x,t} \left[\lim_{h \rightarrow 0} \frac{g(\xi(t)) - g(\xi(t+h)) - g(\xi(t+h)) \int_{t-h}^t v(\xi(s+h), s) ds + O(h^2)}{h} \right] \\
&= E_{x,t} \left[-\frac{dg}{dt}(x) - v(x, t) g(x) \right], \quad \text{taking the limit.} \\
&= E_{x,t} \left[-\frac{1}{2} \frac{\partial^2 g}{\partial x^2}(x) - \frac{\partial g}{\partial x}(x) \frac{dW}{dt} - v(x, t) g(x) \right], \quad \text{applying Ito's Formula.} \\
&= -\frac{1}{2} \frac{\partial^2 g}{\partial x^2}(x) - v(x, t) g(x), \quad (\text{Note: } E \left(\frac{dW}{dt} \right) = E(\Upsilon) = 0.) \\
\frac{\partial u}{\partial t} &= -\frac{1}{2} \frac{\partial^2 u}{\partial x^2}(x, t) - v(x, t) u(x, t)
\end{aligned}$$

Therefore, $u(x, t)$ satisfies the differential equation, Equation 6.10. In addition, the final condition $u(x, T) = f(x)$ is trivially true. •

The Feynman-Kac Formula as stated above is difficult to use directly because it refers to a differential equation with *final* conditions instead of initial conditions.

This is rectified as follows:

Let $\tau = T - t$ and then define $\tilde{u}(x, \tau) = u(x, t)$, $\tilde{v}(x, \tau) = v(x, t)$. This results

in the following differential equation in place of Equation 6.10:

$$\frac{\partial \tilde{u}}{\partial \tau} = \frac{1}{2} \frac{\partial^2 \tilde{u}}{\partial x^2}(x, \tau) + \tilde{v}(x, \tau) \tilde{u}(x, \tau) \quad , \quad \tau \geq 0$$

with initial condition $\tilde{u}(x, 0) = f(x)$.

The solution is given as follows, using Equation 6.11:

$$\begin{aligned} \tilde{u}(x, \tau) &= u(x, t) = E_{x,t} \left[f(\xi(T)) e^{\int_t^T v(\xi(s), s) ds} \right] \\ &= E_{x, T-\tau} \left[f(\xi(T)) e^{\int_{T-\tau}^T v(\xi(s), s) ds} \right] \\ &= E_{x, T-\tau} \left[f(\xi(T)) e^{\int_{T-\tau}^T \tilde{v}(\xi(s), T-s) ds} \right] \end{aligned}$$

Now let $\tilde{\xi}(r)$ be defined by the following:

$$\tilde{\xi}(r) = \xi(s)$$

where $r = s - (T - \tau)$.

This is a shift back of the local time of the stochastic process so that $\tilde{\xi}(0) = x$, in place of $\xi(T - \tau) = x$ for the original.

Then, we have the following:

$$\tilde{u}(x, \tau) = E_{x,0} \left[f(\tilde{\xi}(\tau)) e^{\int_0^\tau \tilde{v}(\tilde{\xi}(r), \tau-r) dr} \right]$$

Eliminating all of the tildes and renaming τ as t , we now have shown that the solution to the following initial value problem:

$$\frac{\partial u}{\partial t} = \frac{1}{2} \frac{\partial^2 u}{\partial x^2}(x, t) + v(x, t) u(x, t) \quad , \quad t \geq 0 \quad (6.12)$$

with initial condition $u(x, 0) = f(x)$, is given by the following formula:

$$u(x, t) = E_{x,0} \left[f(\xi(t)) e^{\int_0^t v(\xi(r), t-r) dr} \right] \quad (6.13)$$

6.3.4 Time Discretization, Monte Carlo Estimates

From the last section, we have the following Feynman-Kac Formula:

$$u(x, t) = E_{x,0} \left[f(\xi(t)) e^{\int_0^t v(\xi(r), t-r) dr} \right]$$

where the expectation is taken over Wiener paths starting at $\xi(0) = x$ and progressing until time t .

In practice, the solution is computed by discretizing time, generating Wiener process sample paths, and averaging approximate values of the quantity within the brackets above over several paths to obtain Monte Carlo estimates of the expectation.

For generating the Wiener process sample paths, the most commonly used method, which I will refer to as the **Standard Discretization**, can be defined as follows:

For the solution at a particular point (x, t) , choose m to be the number of equally spaced time steps. Let $\Delta t = \frac{t}{m}$, $t_i = i \Delta t$ ($i = 0, \dots, m$).

The Wiener paths can be exactly simulated at the discrete times t_i by summing up Gaussian random variables $\Delta W^{(i)} \in \mathcal{N}(0, \Delta t)$ as follows:

$$\begin{aligned} \xi^{(0)} &= x \\ \xi^{(i)} &= x + \sum_{j=1}^i \Delta W^{(j)} \quad , \quad (i = 1, \dots, m) \end{aligned} \tag{6.14}$$

where $\xi^{(i)} = \xi(t_i)$.

This produces independent Gaussian intervals as required. The Gaussian random variables are generated by transforming uniform random variables, or in practice pseudorandom or quasi-random sequences of points. The method of transformation is itself important, especially when one is using quasi-random sequences, as will be discussed shortly.

The sample paths generated in this way are exact at the discrete times t_i , but an approximation is involved when the path integral of v is computed. For example, one could use the left endpoint rule as follows:

$$\int_0^t v(\xi(r), t-r) dr \approx \Delta t \sum_{i=0}^{m-1} v(\xi^{(i)}, t-t_i)$$

One might expect a discretization error of order $O(\Delta t)$. However, the nondifferentiability of the paths actually leads to an order $O(\sqrt{\Delta t})$ error term in addition to an order $O(\Delta t)$ error term. Consider the following purely heuristic argument:

$$\begin{aligned} dv(\xi(r), t-r) &= \frac{\partial v}{\partial \xi} \frac{d\xi}{dr} dr - \frac{\partial v}{\partial r} dr \\ &= \frac{\partial v}{\partial \xi} d\xi - \frac{\partial v}{\partial r} dr = \frac{\partial v}{\partial \xi} dW - \frac{\partial v}{\partial r} dr \\ &= c_1(dt)^{1/2} + c_2 dt \end{aligned}$$

This will be considered in greater detail in Chapter 7. For this Chapter, I will deal with examples for which Δt is sufficiently small so that any such discretization errors, which act as biases, are insignificant relative to the statistical convergence errors of the Monte Carlo estimates which will be presented next. The main concern in this Chapter is to use quasi-random sequences to reduce the statistical errors.

Having approximated the integral as above, one can compute the following Monte Carlo estimate:

$$\hat{u}_N(x, t) = \frac{1}{N} \sum_{k=1}^N f(\xi_k^{(m)}) e^{I_k} \quad (6.15)$$

where,

$$\begin{aligned} \xi_k^{(i)} &= x + \sum_{j=1}^i \Delta W_k^{(j)} \quad , \quad (k = 1, \dots, N) \\ \Delta W_k^{(j)} &\in \mathcal{N}(0, \Delta t) \quad , \quad (j = 1, \dots, m; k = 1, \dots, N) \\ I_k &= \Delta t \sum_{i=0}^{m-1} v(\xi_k^{(i)}, t - t_i) \quad , \quad (k = 1, \dots, N) \end{aligned}$$

One then has the following:

$$E(\hat{u}_N(x, t)) = u(x, t) + \text{bias error}$$

where we will assume, for this chapter, that the bias error is trivial.

Next, define the following functions:

$$\begin{aligned} \tilde{f}(x_1, x_2, \dots, x_m) &= f(\xi^{(m)}) \\ \tilde{I}(x_1, x_2, \dots, x_m) &= \Delta t \sum_{i=0}^{m-1} v(\xi^{(i)}, t - t_i) \end{aligned}$$

where $\xi^{(i)} = x + \sum_{j=1}^i G(x_j)$, and G represents the transformation of points, x_j , which are uniformly distributed, e.g. pseudorandom or quasi-random points, into points which are normally distributed according to $\mathcal{N}(0, \Delta t)$.

This explicit representation allows us to write out the expectation of the Monte Carlo estimate in Equation 6.15 as the following integral:

$$E(\hat{u}_N(x, t)) = \int_0^1 \cdots \int_0^1 \tilde{f}(x_1, \dots, x_m) e^{\tilde{I}(x_1, \dots, x_m)} dx_1 \dots dx_m$$

It is this integral which the Monte Carlo estimate is in effect estimating using pseudorandom or quasi-random sequences of points. Note that the transformation method is important because, for example, if it is discontinuous it can hurt the effectiveness of quasi-random sequences. Therefore, I have used a method derived by George Marsaglia (Marsaglia [40]) which is continuous and monotonic as a function of the original sample points. Other methods without such properties, for example Box-Muller, were found to give poorer results with quasi-random sequences. (See Morokoff [41] for some further study of the Box-Muller case.)

6.3.5 Application of Quasi-Random Sequences, Reformulation

The Monte Carlo estimate described in the last section, Equation 6.15, can be improved by directly substituting quasi-random points for pseudorandom, as long as the function v is of bounded variation and the number of dimensions, m , is not too large. This will be the case in the example which I will consider in Section 6.4. In general, the direct use of quasi-random sequences in this way is limited to short times, t , since otherwise the reduction of time discretization errors to acceptable levels usually will require m to be rather large. In such cases when m is very large (e.g. > 100), then an entirely new approach is needed, such as some of the methods in Chapter 7.

When the number of dimensions is not large enough to require more advanced techniques, but large enough to start having a negative impact on the use of quasi-

Contributions to Variance						
Unif. Seq.	Steps of Wiener Path					Component
Components	$\xi(t_0)$	$\xi(t_1)$	$\xi(t_2)$	\cdots	$\xi(t_m)$	totals
x_1	0	t/m	t/m	\cdots	t/m	t
x_2	0	0	t/m	\cdots	t/m	$(m-1)t/m$
\vdots	\vdots	\vdots	\ddots	\ddots	\vdots	\vdots
x_m	0	0	0	\cdots	t/m	t/m
Path totals	0	t/m	$2t/m$	\cdots	t	

Table 6.3: Importance of Dimensions, Standard discretization

random sequences, then a reformulation of the Standard discretization method can lead to significant improvements in the quasi-random Monte Carlo estimates by concentrating more of the variance into the lowest dimensions. Such a reformulation, which I will refer to as the **Alternate Discretization**, is presented next.

First, consider Table 6.3, which shows the contribution of each dimension to the variance of the Wiener sample paths when the Standard discretization, given by Equation 6.14, is used. (Note that the variances add for independent random points by Equation 2.12.)

The values in the table indicate that the lower dimensions account for more of the variance in the Wiener paths than do the higher dimensions. This is a desirable feature if one intends to use quasi-random sequences. However, as we will see next, it is possible to make the lower dimensions account for an even larger share of the variance by reformulating the discretization method so that the

lower dimensions represent large steps, while the higher dimensions simply fill in intermediate positions. This will lead to more accurate quasi-random Monte Carlo estimates.

Now the **Alternate Discretization** will be defined. For convenience, we will assume that m is a power of two, This is optimal for the new discretization method, but a modification to handle any value of m is easily obtained.

Let,

$$\begin{aligned}
\xi^{(0)} &= x \\
\xi^{(m)} &= \xi^{(0)} + W_1 \quad , \quad W_1 \in \mathcal{N}(0, t) \\
\xi^{(\frac{m}{2})} &= \frac{\xi^{(0)} + \xi^{(m)}}{2} + W_2 \quad , \quad W_2 \in \mathcal{N}(0, t/4) \\
\xi^{(\frac{m}{4})} &= \frac{\xi^{(0)} + \xi^{(\frac{m}{2})}}{2} + W_3 \quad , \quad W_3 \in \mathcal{N}(0, t/8) \\
\xi^{(\frac{3m}{4})} &= \frac{\xi^{(\frac{m}{2})} + \xi^{(m)}}{2} + W_4 \quad , \quad W_4 \in \mathcal{N}(0, t/8) \\
\xi^{(\frac{m}{8})} &= \frac{\xi^{(0)} + \xi^{(\frac{m}{4})}}{2} + W_5 \quad , \quad W_5 \in \mathcal{N}(0, t/16) \\
\xi^{(\frac{3m}{8})} &= \frac{\xi^{(\frac{m}{4})} + \xi^{(\frac{m}{2})}}{2} + W_6 \quad , \quad W_6 \in \mathcal{N}(0, t/16) \\
\xi^{(\frac{5m}{8})} &= \frac{\xi^{(\frac{m}{2})} + \xi^{(\frac{3m}{4})}}{2} + W_7 \quad , \quad W_7 \in \mathcal{N}(0, t/16) \\
\xi^{(\frac{7m}{8})} &= \frac{\xi^{(\frac{3m}{4})} + \xi^{(m)}}{2} + W_8 \quad , \quad W_8 \in \mathcal{N}(0, t/16) \\
\xi^{(\frac{m}{16})} &= \frac{\xi^{(0)} + \xi^{(\frac{m}{8})}}{2} + W_9 \quad , \quad W_9 \in \mathcal{N}(0, t/32) \\
&\vdots
\end{aligned}$$

For this discretization method, the first step is directly from 0 to t . Then the intermediate steps are filled in by taking successive subdivisions of the time

intervals into halves. Each new intermediate path position is determined using the following general rule:

$$\xi^{(j)} = \frac{\xi^{(j_1)} + \xi^{(j_2)}}{2} + W \quad , \quad W \in \mathcal{N}(0, \frac{t_{j_2} - t_{j_1}}{4}) \quad (6.16)$$

where j_1, j_2 are the indices of the nearest prior and later time steps, respectively, for which the positions have already been determined.

This is based on the following conditional probability result, for a process referred to as a Brownian bridge or tied-down Wiener process. (Note: $p(x, s, y, t) = p\{W(t) = y | W(s) = x\}$ is the transition density function of the Wiener process.):

$$\begin{aligned} p\left\{\xi\left(\frac{t_a + t_b}{2}\right) = y \mid \xi(t_a) = x_a, \xi(t_b) = x_b\right\} \\ &= \frac{p(x_a, t_a, y, \frac{t_a + t_b}{2}) p(y, \frac{t_a + t_b}{2}, x_b, t_b)}{p(x_a, t_a, x_b, t_b)} \\ &= \frac{\frac{1}{2\pi T} e^{-\frac{(y-x_a)^2}{2T}} e^{-\frac{(x_b-y)^2}{2T}}}{\frac{1}{\sqrt{2\pi \cdot 2T}} e^{-\frac{(x_b-x_a)^2}{4T}}} \quad , \quad \text{where } T = \frac{t_b - t_a}{2} \\ &= \frac{1}{\sqrt{\pi T}} e^{-\frac{1}{4T} [2(y-x_a)^2 + 2(x_b-y)^2 - (x_b-x_a)^2]} \\ &= \frac{1}{\sqrt{\pi T}} e^{-\frac{1}{4T} [4y^2 - 4y(x_a+x_b) + 2x_a^2 + 2x_b^2 - x_b^2 + 2x_ax_b - x_a^2]} \\ &= \frac{1}{\sqrt{\pi T}} e^{-\frac{1}{4T} [4(y - \frac{x_a+x_b}{2})^2 - (x_a+x_b)^2 + x_a^2 + x_b^2 + 2x_ax_b]} \\ &= \frac{1}{\sqrt{2\pi \cdot \frac{T}{2}}} e^{-\frac{(y - \frac{x_a+x_b}{2})^2}{2 \cdot \frac{T}{2}}} \end{aligned}$$

Therefore,

$$y \in \mathcal{N}\left(\frac{x_a + x_b}{2}, \frac{t_b - t_a}{4}\right) \quad (6.17)$$

Note that it is the Markov property which allows each successive path location to be determined by a distribution which is dependent only on the nearest locations

already assigned.

The resultant scheme can be rewritten in a more convenient form which gives each of the path locations directly in terms of the normally distributed variables W_i ($i = 1, \dots, m$) as follows:

$$\xi^{(0)} = x \quad (6.18)$$

$$\xi^{(m)} = x + W_1, \quad W_1 \in \mathcal{N}(0, t) \quad (6.19)$$

$$\xi^{(\frac{m}{2})} = x + \frac{W_1}{2} + W_2, \quad W_2 \in \mathcal{N}(0, t/4) \quad (6.20)$$

$$\xi^{(\frac{m}{4})} = x + \frac{W_1}{4} + \frac{W_2}{2} + W_3, \quad W_3 \in \mathcal{N}(0, t/8) \quad (6.21)$$

$$\xi^{(\frac{3m}{4})} = x + \frac{3W_1}{4} + \frac{W_2}{2} + W_4, \quad W_4 \in \mathcal{N}(0, t/8) \quad (6.22)$$

$$\xi^{(\frac{m}{8})} = x + \frac{W_1}{8} + \frac{W_2}{4} + \frac{W_3}{2} + W_5, \quad W_5 \in \mathcal{N}(0, t/16) \quad (6.23)$$

$$\xi^{(\frac{3m}{8})} = x + \frac{3W_1}{8} + \frac{3W_2}{4} + \frac{W_4}{2} + W_6, \quad W_6 \in \mathcal{N}(0, t/16) \quad (6.24)$$

\vdots

Now, we can compare the variance contributed by each dimension to the Wiener sample paths using the Alternate discretization, with the contributions under the Standard discretization. Table 6.4 shows the results for the specific case, $m = 8$.

Clearly, the Alternate scheme concentrates much more of the variance in the lower dimensions.

Figure 6.4 is intended to be a graphical comparison of the Standard and Alternate discretization methods. It shows how more of the variance is concentrated in the lower dimensions for the Alternate method. The two plots show a simple case

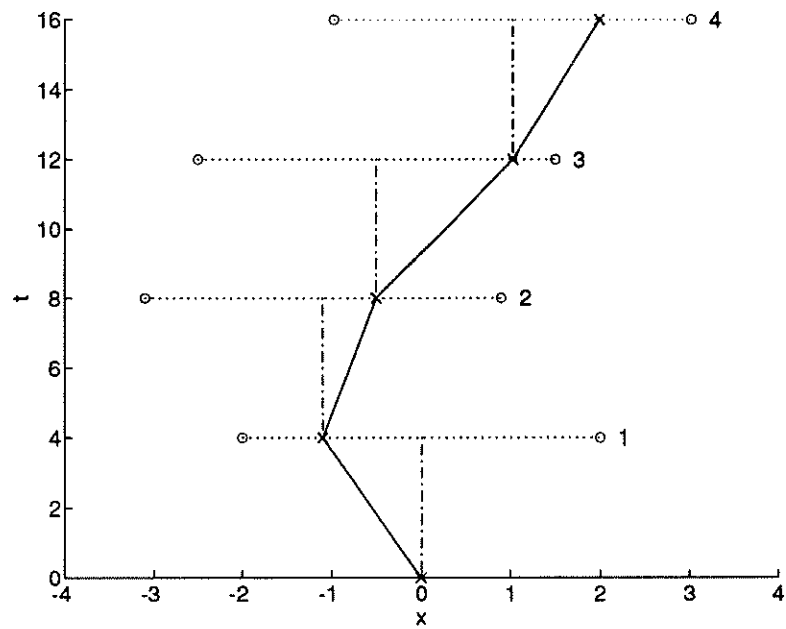
Contributions to Variance, $m = 8$

Dim	Steps of Wiener Path									Alt	Std
	ξ_0	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6	ξ_7	ξ_8	Tot	Tot
x_1	0	$\frac{t}{64}$	$\frac{t}{16}$	$\frac{9t}{64}$	$\frac{t}{4}$	$\frac{25t}{64}$	$\frac{9t}{16}$	$\frac{49t}{64}$	t	$3.1875t$	$1.0000t$
x_2	0	$\frac{t}{64}$	$\frac{t}{16}$	$\frac{9t}{64}$	$\frac{t}{4}$	$\frac{9t}{64}$	$\frac{t}{16}$	$\frac{t}{64}$	0	$0.6875t$	$0.8750t$
x_3	0	$\frac{t}{32}$	$\frac{t}{8}$	$\frac{t}{32}$	0	0	0	0	0	$0.1875t$	$0.7500t$
x_4	0	0	0	0	0	$\frac{t}{32}$	$\frac{t}{8}$	$\frac{t}{32}$	0	$0.1875t$	$0.6250t$
x_5	0	$\frac{t}{16}$	0	0	0	0	0	0	0	$0.0625t$	$0.5000t$
x_6	0	0	0	$\frac{t}{16}$	0	0	0	0	0	$0.0625t$	$0.3750t$
x_7	0	0	0	0	0	$\frac{t}{16}$	0	0	0	$0.0625t$	$0.2500t$
x_8	0	0	0	0	0	0	0	$\frac{t}{16}$	0	$0.0625t$	$0.1250t$
Tot	0	$\frac{t}{8}$	$\frac{t}{4}$	$\frac{3t}{8}$	$\frac{t}{2}$	$\frac{5t}{8}$	$\frac{3t}{4}$	$\frac{7t}{8}$	t		

Table 6.4: Importance of Dimensions, Comparison for $m = 8$.

where the number of steps, m , is just 4. The dotted lines show one standard deviation ranges. The solid lines show the actual discretized stochastic paths. In the upper figure, the dash/dot line shows the additive nature of the successive steps. In the lower figure, the dashed line shows the effects of the large first step on all of the intermediate positions, while the dash/dot lines show the effects of the second step on intermediate positions. Note that the two plots show the same resultant paths for convenience only. There is no connection between the particular path generated by the same pseudorandom or quasi-random point using each of the two different methods. There is only an equivalence in terms of the expectations over all such paths.

Standard Scheme, 4 Steps



Alternate Scheme, 4 Steps

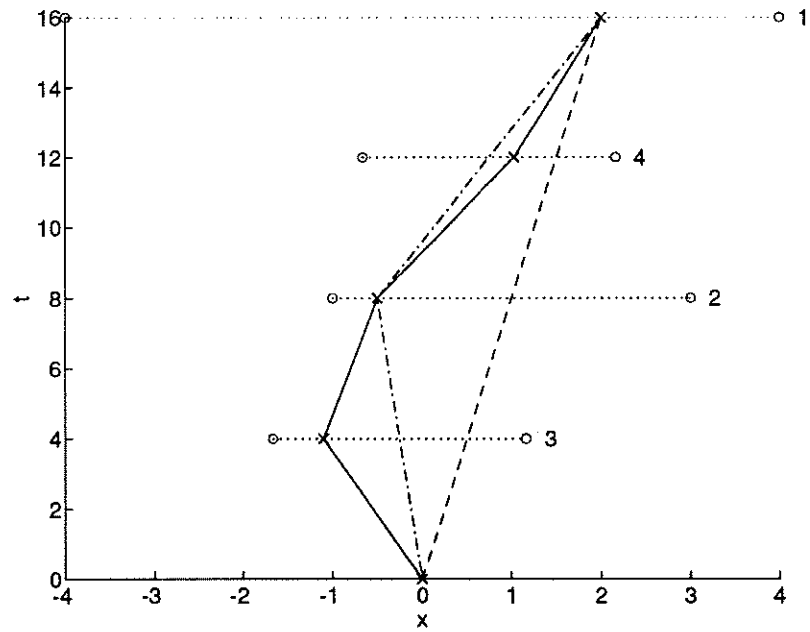


Figure 6.4: Graphical Comparison of Discretization Schemes.

6.4 Computational Example for the Feynman-Kac Formula

This example will test the accuracy of Monte Carlo estimates of the solution of a simple linear parabolic differential equation using Equation 6.15 at selected values of x and a fixed (small) time t . A comparison will be made of pseudorandom and quasi-random sequences with either the Standard discretization, Equation 6.14, or the Alternate discretization, Equation 6.16, being employed to generate the necessary Wiener paths.

Example 9 *Consider the following linear parabolic differential equation:*

$$\frac{\partial f}{\partial t}(x, t) = \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(x, t) + \left(\frac{1}{t+1} + \frac{1}{x^2+1} - \frac{4x^2}{(x^2+1)^2} \right) \frac{\partial f}{\partial x}(x, t)$$

with initial condition $f(x, 0) = \frac{1}{x^2+1}$.

It has the following analytical solution: $f(x, t) = \frac{t+1}{x^2+1}$.

This exact solution is compared with the following Monte Carlo estimates using the Feynman-Kac formula at a fixed (small) time, $T > 0$, and a set of fixed points in space, x_1, \dots, x_P :

$$\hat{f}_N(x_p, T) = \frac{1}{N} \sum_{k=1}^N f(\xi_{p,k}^{(m)}, 0) e^{I_{p,k}^{(m)}} \quad , \quad (p = 1, \dots, P)$$

where for the Standard discretization,

$$\begin{aligned} \xi_{p,k}^{(i)} &= x_p + \sum_{j=1}^i \Delta W_k^{(j)} \quad , \quad (i = 0, \dots, m; k = 1, \dots, N; p = 1, \dots, P) \\ I_{p,k} &= \Delta t \sum_{i=0}^{m-1} a(\xi_{p,k}^{(i)}, t - t_i) \quad , \quad (k = 1, \dots, N; p = 1, \dots, P) \end{aligned}$$

$$\Delta W_k^{(j)} \in \mathcal{N}(0, \Delta t) \text{ , } (j = 1, \dots, m; k = 1, \dots, N)$$

$$\Delta t = \frac{T}{m} \text{ , } t_i = i \Delta t \text{ , } m = \# \text{ of equal time steps, or dimensions.}$$

while for the Alternate discretization, the random paths, $\xi_{p,k}^{(i)}$, are obtained using the method described explicitly by Equations 6.18-6.24.

The Gaussian variables are obtained using Marsaglia's transformation on either pseudorandom or quasi-random points (see Section 6.3.4).

The estimates are then compared with the exact solution using the following L^2 -sense distance calculation:

$$L2e = \sqrt{\frac{1}{P} \sum_{p=1}^P [f(x_p, T) - \hat{f}_N(x_p, T)]^2}$$

Note: In the example P is 8 and x_p ranges from +3 to -3, equally spaced.

This can also be compared with a purely empirical estimate of the L^2 error using the sample variance for each estimate as follows (results not shown):

$$L2sd = \sqrt{\frac{1}{P} \sum_{p=1}^P \hat{s}_N^2(x_p)}$$

where

$$\hat{s}_N^2(x_p) = \frac{N}{N-1} \left(\frac{1}{N} \sum_{k=1}^N [f(\xi_{p,k}^{(m)}, 0) e^{I_{p,k}}]^2 - \hat{f}_N(x_p, T)^2 \right)$$

Notice that the same Wiener increments are used for each of the estimates above at a different point in space. This saves a great deal of computer time and is acceptable since each different x_p 's estimate is computed separately from the others. T is selected to be very small so that any discretization errors are insignificant relative to the Monte Carlo errors at the values of N and m considered.

In Table 6.5, the calculated values of $L2e$ are compared at selected values of N and m using pseudorandom and quasi-random sequences and either the Standard or Alternate discretization. T is taken to be .02 when $m = 8$, .04 when $m = 16$, and .08 when $m = 32$ (constant time step size). Figures 6.5-6.10 show plots for each of the T, M pairs above. (Note: cpu times are not listed in the table, but plots of the error measurement, $L2e$, versus cpu time are included for completeness in the figures.)

The results indicate that, as predicted, the Alternate discretization method allows for significantly more accurate results when using quasi-random sequences with the Feynman-Kac formula. In contrast with this is the fact that virtually no improvement is seen when using pseudorandom sequences. This shows that there is nothing inherently better or special about the new discretization method in general. It is only when it is combined with quasi-random sequences that the positive effect of concentrating more of the variance in the first few dimensions becomes apparent.

Feynman-Kac Formula, L_2 Comparisons

$T = 0.02, m = 8$ (dimensions)				
	Standard Discr		Alternate Discr	
N	Pseudorand	Halton	Pseudorand	Halton
100	4.810 E -3	2.175 E -3	4.823 E -3	6.671 E -4
400	3.005 E -3	4.877 E -4	3.001 E -3	1.903 E -4
1600	1.389 E -3	1.941 E -4	1.276 E -3	5.599 E -5
6400	6.116 E -4	4.729 E -5	7.736 E -4	1.396 E -5
25600	3.248 E -4	1.350 E -5	3.932 E -4	4.307 E -6
conv rates	-0.502	-0.876	-0.460	-0.901

$T = 0.04, m = 16$ (dimensions)				
	Standard Discr		Alternate Discr	
N	Pseudorand	Halton	Pseudorand	Halton
100	7.490 E -3	4.529 E -3	6.991 E -3	9.779 E -4
400	3.352 E -3	2.032 E -3	3.743 E -3	2.698 E -4
1600	1.617 E -3	4.946 E -4	2.210 E -3	7.684 E -5
6400	9.457 E -4	1.455 E -4	9.090 E -4	1.870 E -5
25600	4.307 E -4	4.530 E -5	4.416 E -4	5.695 E -6
conv rates	-0.493	-0.819	-0.493	-0.921

$T = 0.08, m = 32$ (dimensions)				
	Standard Discr		Alternate Discr	
N	Pseudorand	Halton	Pseudorand	Halton
100	1.214 E -2	1.032 E -2	1.092 E -2	1.484 E -3
400	5.642 E -3	4.051 E -3	5.621 E -3	3.877 E -4
1600	2.474 E -3	1.626 E -3	2.568 E -3	1.062 E -4
6400	1.504 E -3	5.859 E -4	1.371 E -3	2.468 E -5
25600	7.008 E -4	1.771 E -4	5.889 E -4	7.355 E -6
conv rates	-0.506	-0.774	-0.500	-0.951

Table 6.5: Results for Example 9, 75 trials

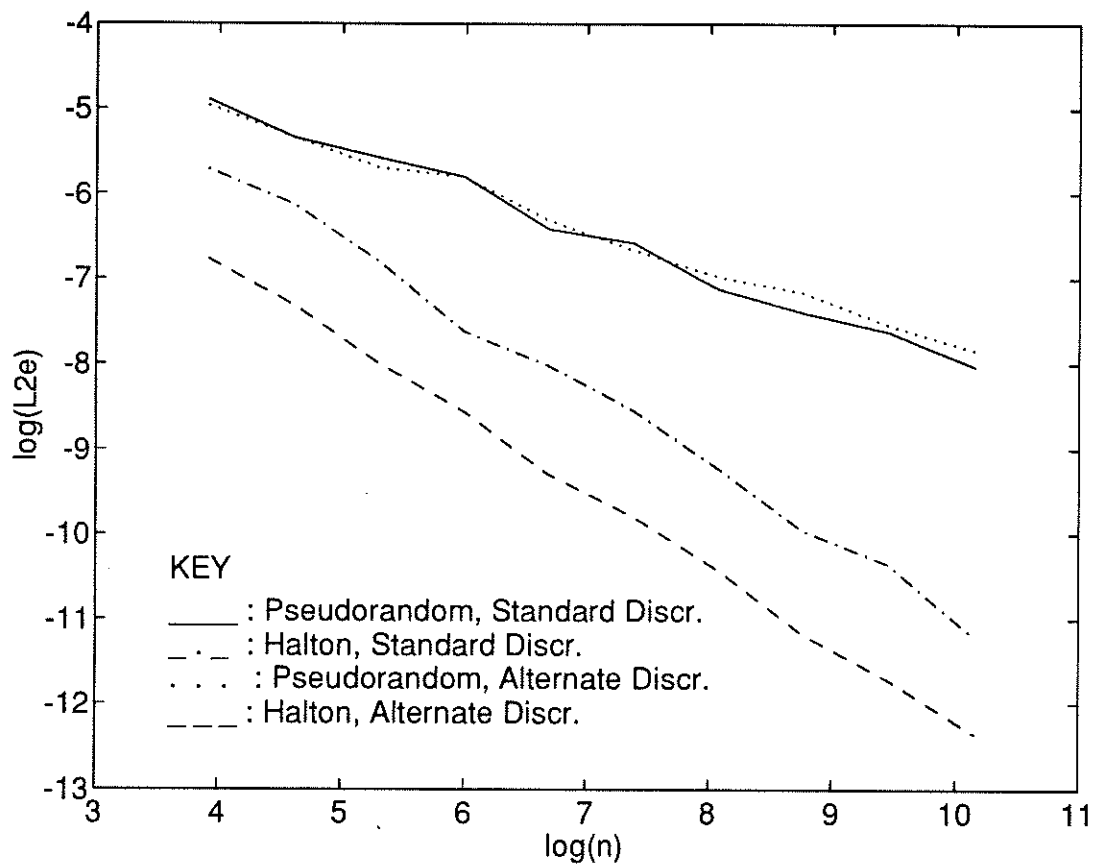


Figure 6.5: Log-Log Plot for Example 9, $T = 0.02$, $m = 8$.

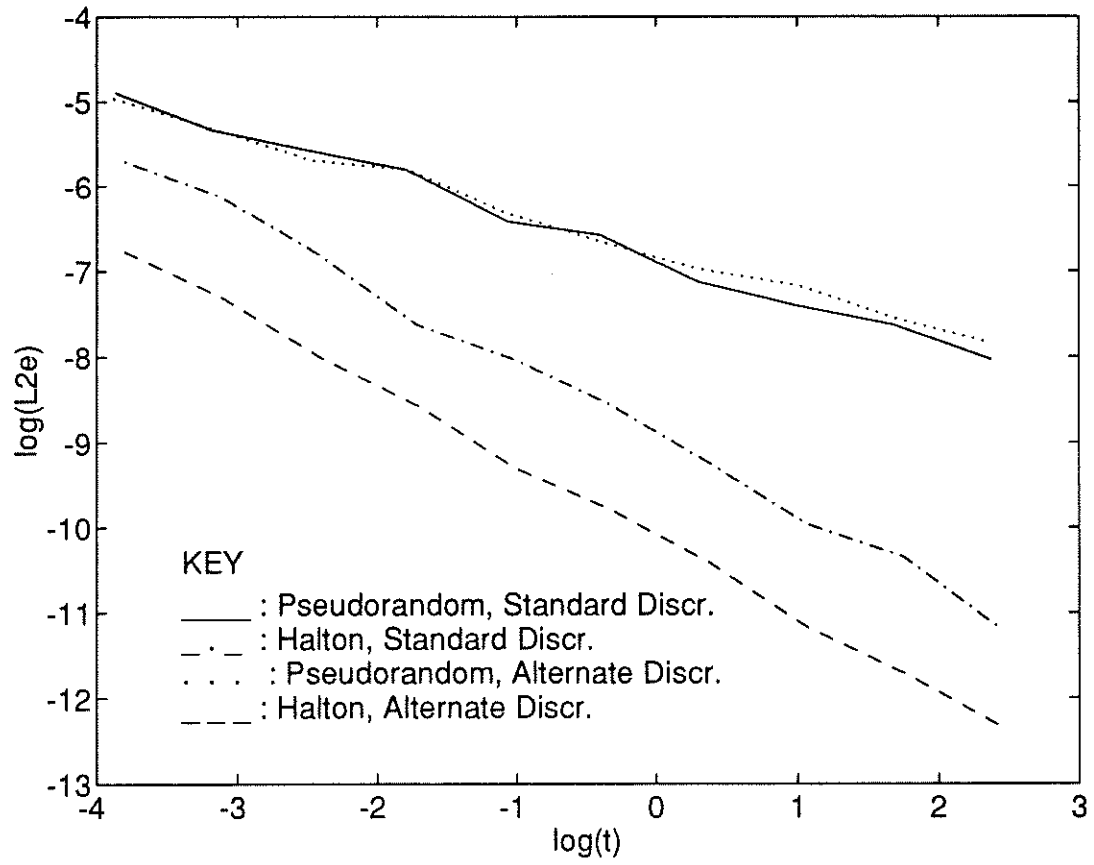


Figure 6.6: Log-Log Plot vs. $t(\text{sec})$ for Example 9, $T = 0.02$, $m = 8$.

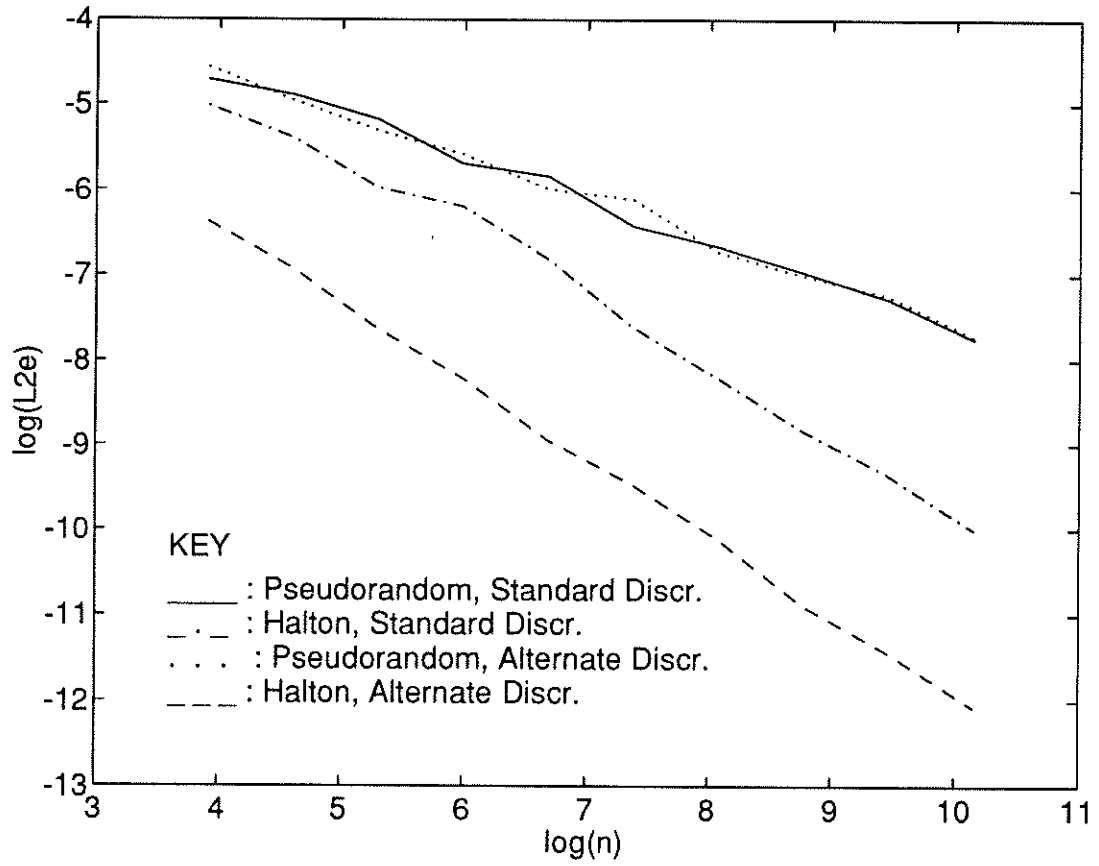


Figure 6.7: Log-Log Plot for Example 9, $T = 0.04$, $m = 16$.

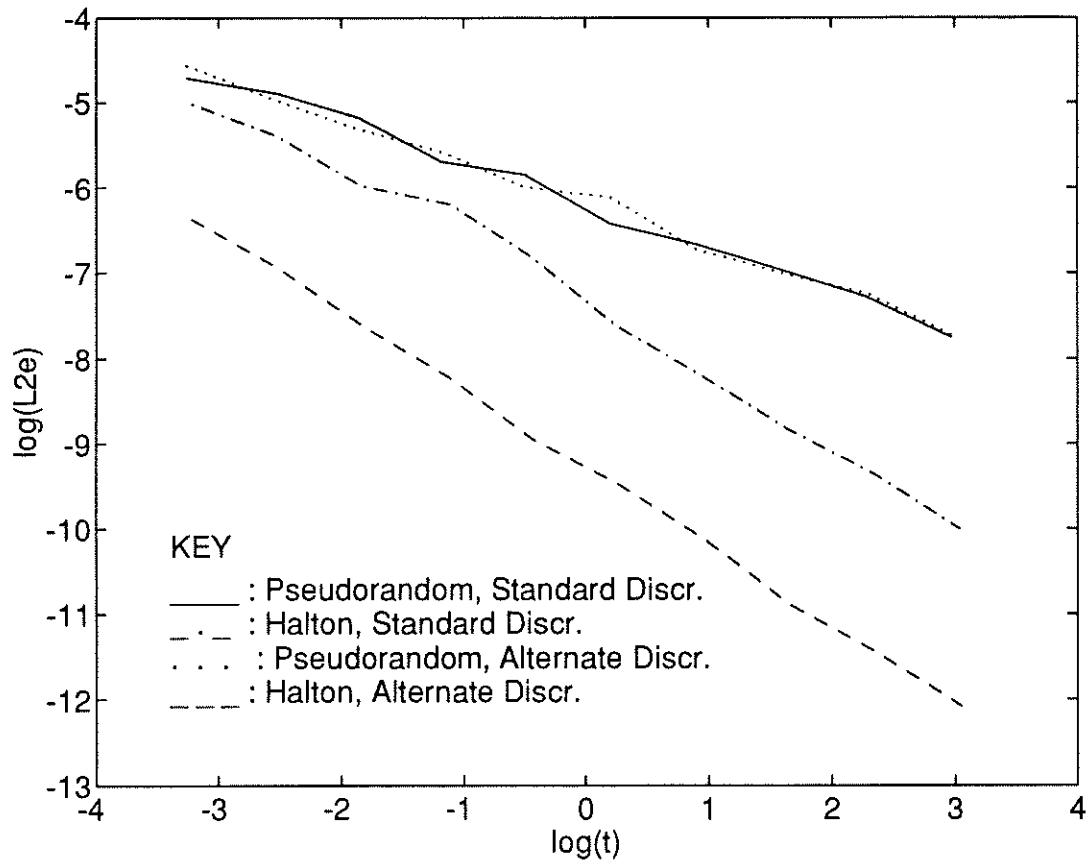


Figure 6.8: Log-Log Plot vs $t(\text{sec})$ for Example 9, $T = 0.04$, $m = 16$.

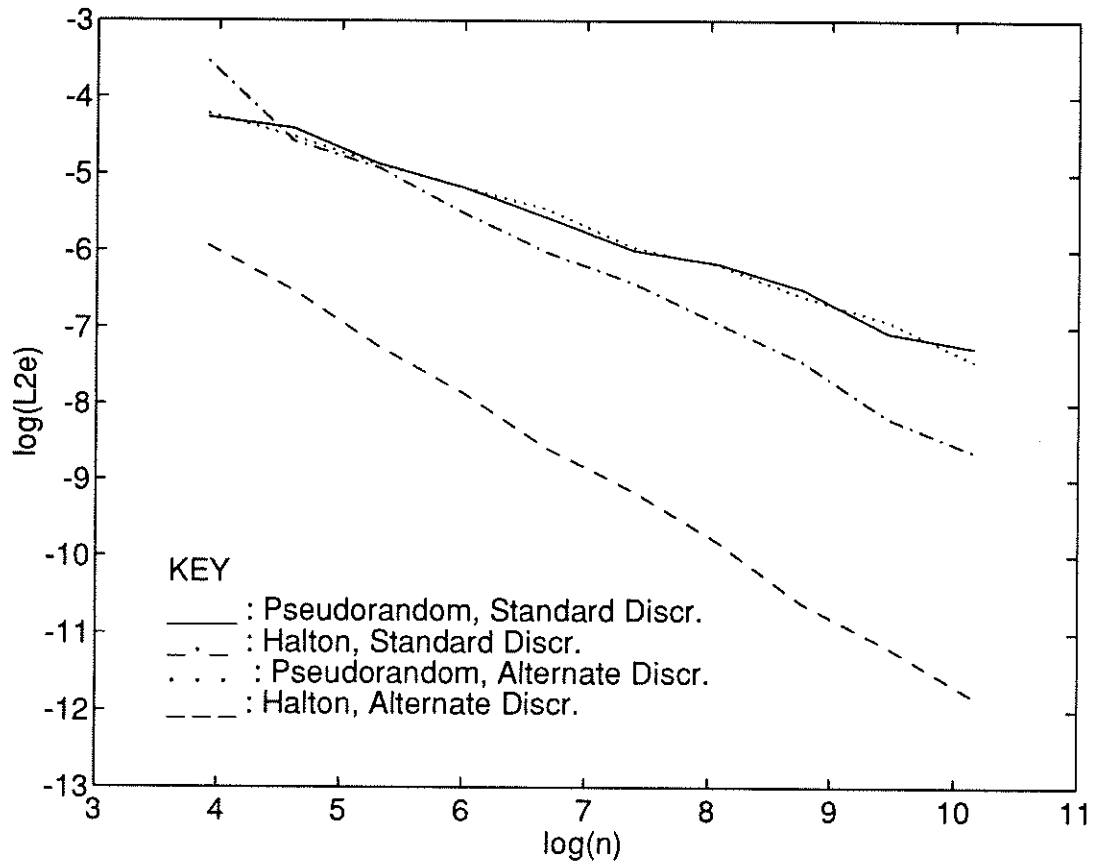


Figure 6.9: Log-Log Plot for Example 9, $T = 0.08$, $m = 32$.

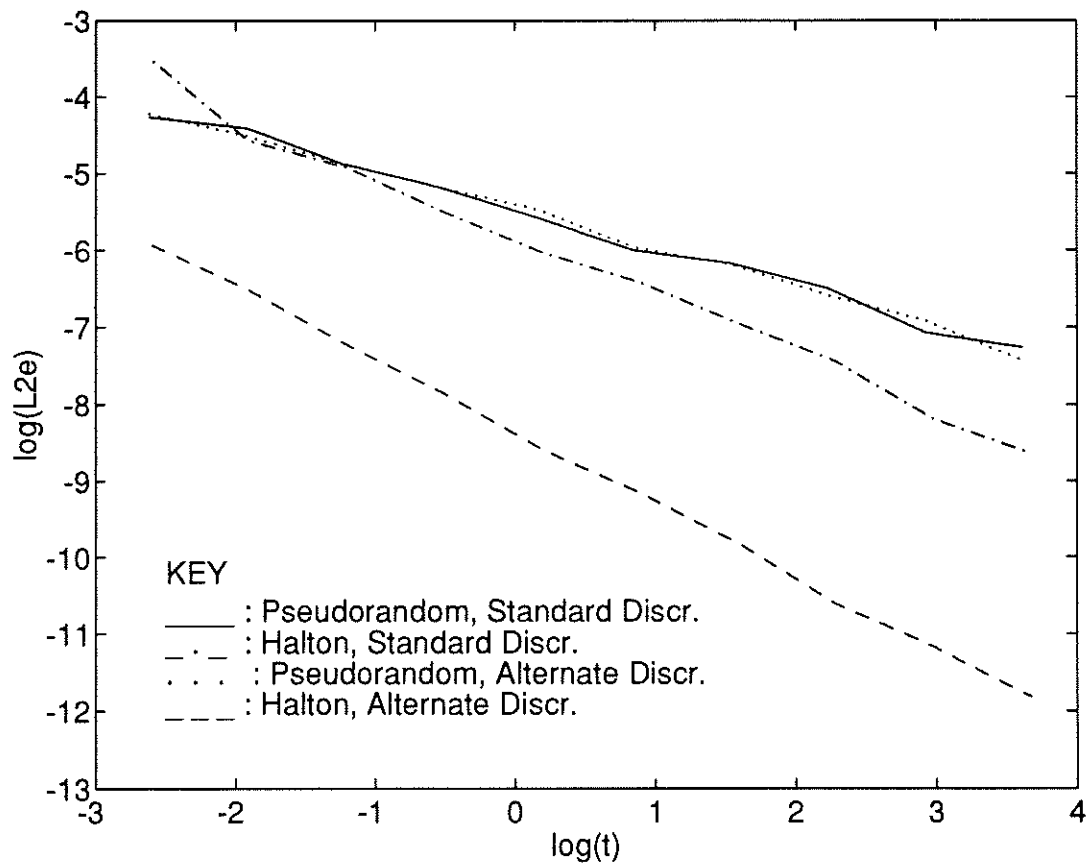


Figure 6.10: Log-Log Plot vs $t(\text{sec})$ for Example 9, $T = 0.08$, $m = 32$.

Part III

Advanced Applications : Stochastic Processes

CHAPTER 7

Quantum Monte Carlo Methods

One major application of Monte Carlo methods is in the area of quantum mechanics. Such techniques, known collectively as Quantum Monte Carlo, are regularly used by theoretical physicists and chemists. There are several varieties of Quantum Monte Carlo including Green's Function Monte Carlo, Path Integral Monte Carlo, and Diffusion Monte Carlo, but they are all similar in terms of their usage and basic theory. To my knowledge, the application of quasi-random sequences to the Quantum Monte Carlo methods discussed here has not been pursued previously, or if it has, the lack of success has led to few published results. Notwithstanding the difficulties involved, I have obtained encouraging results on the examples which I have studied.

Green's Function Monte Carlo (GFMC) refers to a class of Monte Carlo techniques for approximating the exact ground state characteristics of quantum mechanical systems with a fixed potential function in time. When these methods are based on the time-independent form of Schrödinger's Equation, they lead to what may be regarded as the iterative solution of an eigenvalue equation using Monte Carlo integration to perform the iterations.

An alternative approach is based on the time-**dependent** Schrödinger Equation

and involves an additional approximation for finite time steps. These methods are sometimes also referred to as Green's Function Monte Carlo, but they are more commonly known as Diffusion Monte Carlo (DMC), which is the term I will use for them. In contrast with the GFMC methods above, DMC methods may be regarded as the approximate solution of an evolution equation by transforming it into a stochastic diffusion process, and then using time-discretization and Monte Carlo simulation to approximate the process.

Although quasi-random sequences can be successfully applied to either of the above classes of methods, I have had more success dealing with the latter, so that is what I have chosen to concentrate on here. There is certainly a great deal more work which can be done with regards to the application of quasi-random sequences to either GFMC or DMC.

In the discussion of Diffusion Monte Carlo to follow, ground state energy will be treated as the unknown quantity whose value is to be estimated using Monte Carlo methods.

7.1 Diffusion Monte Carlo - Underlying Theory, Convergence Result

Consider the time-dependent Schrödinger Equation for a quantum system of M bodies, with a fixed (in time) potential function:

$$\begin{aligned}
 - \sum_{k=1}^M \frac{\hbar^2}{2m_k} \nabla_k^2 \psi(\mathbf{x}_1, \dots, \mathbf{x}_M, t) + v(\mathbf{x}_1, \dots, \mathbf{x}_M) \psi(\mathbf{x}_1, \dots, \mathbf{x}_M, t) \\
 = i\hbar \frac{\partial}{\partial t} \psi(\mathbf{x}_1, \dots, \mathbf{x}_M, t)
 \end{aligned} \tag{7.1}$$

In the above equation,

m_k is the mass of the k 'th body,

$\mathbf{x}_k = (x_k^{(1)}, x_k^{(2)}, x_k^{(3)})$ is the position of the k 'th body,

$\nabla_k^2 = \frac{\partial^2}{\partial x_k^{(1)2}} + \frac{\partial^2}{\partial x_k^{(2)2}} + \frac{\partial^2}{\partial x_k^{(3)2}}$ is the Laplacian operator on the k 'th body,

$v(\mathbf{x}_1, \dots, \mathbf{x}_M)$ is the fixed potential energy function,

$\psi(\mathbf{x}_1, \dots, \mathbf{x}_M, t)$ is the wavefunction or probability amplitude.

Note: boundary conditions may be specified, or implied by the physics.

The wavefunction's connection to physical reality is that the probability density at location $(\mathbf{x}_1, \dots, \mathbf{x}_M)$ at time t is equal to $|\psi(\mathbf{x}_1, \dots, \mathbf{x}_M, t)|^2$.

Let $\mathbf{y} = (\mathbf{x}_1, \dots, \mathbf{x}_M) \in \mathbb{R}^{3M}$ represent the locations of all the bodies. Then, define Hamiltonian operator, H , as follows :

$$H = - \sum_{k=1}^M \frac{\hbar^2}{2m_k} \nabla_k^2 + v(\mathbf{y}) \tag{7.2}$$

Then Equation 7.1 can be rewritten compactly as follows:

$$H\psi(\mathbf{y}, t) = i\hbar \frac{\partial}{\partial t} \psi(\mathbf{y}, t) \quad (7.3)$$

If one assumes that $v(\mathbf{x})$ is continuous, which I will do, then it can be proven using Sturm-Liouville theory that a complete set of orthonormal eigenfunctions with distinct eigenvalues satisfying the following equations exists:

$$H\psi_j(\mathbf{y}) = E_j \psi_j(\mathbf{y}) \ , \ (j = 0, 1, \dots) \ , \ E_0 < E_1 < \dots \quad (7.4)$$

It can be trivially shown that adding a fixed constant to the potential function will shift each eigenvalue by the same amount, without affecting the eigenfunctions. This allows one to set E_0 near zero by subtracting E_T , a rough estimate of E_0 , from the potential.

A formal solution to Equation 7.3 can now be derived using superposition since H is a linear operator:

Consider initial condition, $\psi(\mathbf{y}, 0) = a_i \psi_i(\mathbf{y})$, for any i .

As an Ansatz, assume that $\psi(\mathbf{y}, t) = T(t) \psi_i(\mathbf{y})$. Then we have:

$$\begin{aligned} H\psi(\mathbf{y}, t) &= i\hbar \frac{\partial}{\partial t} \psi(\mathbf{y}, t) \\ T(t) H\psi_i(\mathbf{y}) &= i\hbar \psi_i(\mathbf{y}) T'(t) \\ T(t) E_i \psi_i(\mathbf{y}) &= i\hbar \psi_i(\mathbf{y}) T'(t) \\ \frac{T'(t)}{T(t)} &= -\frac{i}{\hbar} E_i \\ \Rightarrow T(t) &= a_i e^{-\frac{i}{\hbar} E_i t} \end{aligned}$$

Therefore,

$$\psi(\mathbf{y}, t) = a_i e^{-\frac{i}{\hbar} E_i t} \psi_i(\mathbf{y})$$

Now, applying superposition, one obtains the formal solution to Equation 7.3:

$$\begin{aligned} \text{If } \psi(\mathbf{y}, 0) &= \sum_{i=0}^{\infty} a_i \psi_i(\mathbf{y}) \\ \text{Then } \psi(\mathbf{y}, t) &= \sum_{i=0}^{\infty} a_i e^{-\frac{i}{\hbar} E_i t} \psi_i(\mathbf{y}) \end{aligned} \quad (7.5)$$

where by the orthonormality of the eigenfunctions, we have the following:

$$a_i = \int_{\mathbb{R}^{3M}} \psi(\mathbf{y}, 0) \psi_i(\mathbf{y}) d\mathbf{y} \quad (7.6)$$

Next, we consider a transformation of Equation 7.3 to imaginary time, $\tau = it$.

Let $\phi(\mathbf{y}, \tau) = \psi(\mathbf{y}, it)$, then by using the chain rule an evolution equation for ϕ can be derived as follows:

$$\begin{aligned} H\phi(\mathbf{y}, \tau) &= i\hbar \frac{\partial}{\partial t} \phi(\mathbf{y}, \tau) \\ &= i\hbar \frac{\partial \tau}{\partial t} \frac{\partial}{\partial \tau} \phi(\mathbf{y}, \tau) \\ &= i\hbar i \frac{\partial}{\partial \tau} \phi(\mathbf{y}, \tau) \\ H\phi(\mathbf{y}, \tau) &= -\hbar \frac{\partial}{\partial \tau} \phi(\mathbf{y}, \tau) \end{aligned} \quad (7.7)$$

By using the same argument as above, we obtain a formal solution to Equation 7.7 (Note that the eigenfunctions and eigenvalues are unchanged because H is a purely spatial operator.) :

$$\phi(\mathbf{y}, \tau) = \sum_{i=0}^{\infty} a_i e^{-\frac{1}{\hbar} E_i \tau} \psi_i(\mathbf{y}) \quad (7.8)$$

with initial condition $\phi(\mathbf{y}, 0) = \psi(\mathbf{y}, 0) = \sum_{i=0}^{\infty} a_i \psi_i(\mathbf{y})$. Where again, using orthonormality, one obtains the following:

$$a_i = \int_{\mathbb{R}^{3M}} \phi(\mathbf{y}, 0) \psi_i(\mathbf{y}) d\mathbf{y} \quad (7.9)$$

The advantage of Equation 7.8 over Equation 7.5 is that as the system evolves forward in imaginary time, it will tend to approach its ground state, ψ_0 , which has the lowest energy eigenvalue. This will be examined next. First, however, a brief discussion of normalization is appropriate.

Wavefunctions should be normalized so that the L^2 norm over \mathbb{R}^{3M} is equal to one, i.e. $\int_{\mathbb{R}^{3M}} \phi^2(\mathbf{y}, \tau) d\mathbf{y} = 1$, $\forall \tau$, whenever the wavefunctions are to be understood physically. This is related to the probability amplitude interpretation of them. In real time, this is automatic, but in imaginary time one must normalize the wavefunctions explicitly to treat them physically. Let $\tilde{\phi}(\mathbf{y}, \tau)$ denote the normalized wavefunction. We then require the condition :

$$\int_{\mathbb{R}^{3M}} \tilde{\phi}(\mathbf{y}, t)^2 d\mathbf{y} = 1 \quad (7.10)$$

Consider the expansion of the wavefunction $\phi(\mathbf{y}, \tau)$ as follows:

$$\phi(\mathbf{y}, \tau) = \sum_{i=0}^{\infty} a_i(\tau) \psi_i(\mathbf{y})$$

Also, the expansion of its normalized version, $\tilde{\phi}(\mathbf{y}, \tau)$, as follows:

$$\tilde{\phi}(\mathbf{y}, \tau) = \sum_{i=0}^{\infty} \tilde{a}_i(\tau) \psi_i(\mathbf{y})$$

Then the normalization condition above can be restated using the orthonormality of the eigenfunctions as follows:

$$\sum_{j=0}^{\infty} \tilde{a}_j(\tau)^2 = 1$$

This can be satisfied by dividing $\phi(\mathbf{y}, \tau)$ by $\sqrt{\sum_{j=0}^{\infty} a_j(\tau)^2}$.

Thus, we have in imaginary time evolution, using Equation 7.8:

$$\tilde{\phi}(\mathbf{y}, \tau) = \sum_{i=0}^{\infty} \frac{a_i e^{-\frac{1}{\hbar} E_i \tau}}{\sqrt{\sum_{j=0}^{\infty} a_j^2 e^{-\frac{2}{\hbar} E_j \tau}}} \psi_i(\mathbf{y}) \quad (7.11)$$

This equation describes the evolution of the system in imaginary time in a way which is compatible with the usual normalization requirements of wavefunctions. Note however that Equation 7.8 remains the only true mathematical solution to Equation 7.7, while Equation 7.11 provides a way to relate that solution back to the physics.

Now it will be possible to show that as the system evolves forward in imaginary time, it approaches the ground state. This is done as follows:

$$\begin{aligned} \tilde{a}_0(\tau)^2 &= \frac{a_0^2 e^{-\frac{2}{\hbar} E_0 \tau}}{\left(\sum_{j=0}^{\infty} a_j^2 e^{-\frac{2}{\hbar} E_j \tau} \right)} = \frac{1}{\sum_{j=0}^{\infty} \frac{a_j^2}{a_0^2} e^{-\frac{2}{\hbar} (E_j - E_0) \tau}} \\ &= \frac{1}{1 + \sum_{j=1}^{\infty} \frac{a_j^2}{a_0^2} e^{-\frac{2}{\hbar} (E_j - E_0) \tau}} \\ &= 1 - a_0^{-2} \sum_{j=1}^{\infty} a_j^2 e^{-\frac{2}{\hbar} (E_j - E_0) \tau} + O(e^{-\tau^2}) \end{aligned}$$

Then since $E_j > E_0$, $\forall j > 0$, we have the following result (assuming $a_0 \neq 0$):

$$\tilde{a}_0(\tau)^2 = 1 + O(e^{-\tau}) \quad , \quad \text{as } \tau \rightarrow \infty.$$

Also, since $\sum_{i=0}^{\infty} \tilde{a}_i(\tau)^2 = 1$, this implies the following:

$$\tilde{a}_i(\tau)^2 = O(e^{-\tau}) \quad , \quad \forall i > 0 \quad , \quad \text{as } \tau \rightarrow \infty.$$

Therefore in the L^2 -norm there is exponentially fast convergence to the ground state in imaginary time (assuming $a_0 \neq 0$), which can be written as follows:

$$\|\tilde{\phi}(\mathbf{y}, \tau) - \psi_0(\mathbf{y})\|_2 = O(e^{-\tau}) \quad , \quad \text{as } \tau \rightarrow \infty. \quad (7.12)$$

A weaker expression of this convergence will sometimes be written as follows:

$$\lim_{\tau \rightarrow \infty} \tilde{\phi}(\mathbf{y}, \tau) \stackrel{L^2}{=} \psi_0(\mathbf{y}) \quad (7.13)$$

The assumption that the initial wavefunction is not exactly orthogonal to ψ_0 ($a_0 \neq 0$), is easily satisfied by taking the initial wavefunction to be a crude approximation to the ground state using other methods. This also helps shorten the amount of imaginary time before the wavefunction has ‘converged.’ However, even for an arbitrary initial wavefunction, it is virtually impossible to have no ground state component, given roundoff errors on the computer.

The basic convergence result, Equation 7.13, is what enables Diffusion Monte Carlo to estimate ground state properties of quantum systems.

7.2 Trial Function Reformulation

The evolution equation in imaginary time, Equation 7.7, can be rewritten in explicit form as follows:

$$\frac{\partial}{\partial \tau} \phi(\mathbf{y}, \tau) = \sum_{k=1}^M \frac{\hbar}{2m_k} \nabla_k^2 \psi(\mathbf{y}, \tau) - \frac{1}{\hbar} v(\mathbf{y}) \psi(\mathbf{y}, \tau) \quad (7.14)$$

This is a partial differential equation over $\mathbb{R}^{3M} \times \mathbb{R}$, and methods using time and space discretization to solve this kind of equation such as finite differences are well known. However, they are difficult to apply here in general because of the high number of dimensions involved. Diffusion Monte Carlo is an alternative based on the close relationship between the equation above, and an associated stochastic process involving diffusion and local sources and sinks. As will be shown in Section 7.4, the last term of Equation 7.14 corresponds to a source or sink at each location \mathbf{y} according to the sign and magnitude of $v(\mathbf{y})$, while the summation of terms corresponds to a diffusion of the probability amplitude of each body in the system according to its mass.

When $v(\mathbf{y})$ has a high amount of variation, or is singular at certain values of \mathbf{y} , this can lead to large variations in the results of simulations of the system and potentially explosive growth near singularity points of v . This can cause Monte Carlo estimates of the ground state energy to have very high variances therefore require very large sample sizes to obtain good accuracy. This is important because in particular, the Coulomb potential, $v(\mathbf{y}) = \frac{1}{r_{ab}}$, where r_{ab} is the distance between like-charged particles, is singular as this distance approaches zero.

A common method for avoiding such problems, and hence reducing the variance, involves a reformulation of Equation 7.14 using a *trial function*, $\phi_T(\mathbf{y})$, which is a crude approximation to the ground state, $\psi_0(\mathbf{y})$. For practical reasons, $\phi_T(\mathbf{y})$ should be a function which is relatively easy to compute at any specified value of \mathbf{y} , and to which it is also relatively easy to apply operator H .

Now, to obtain the *trial function reformulation* of Equation 7.14, define the following function:

$$f(\mathbf{y}, \tau) = \phi_T(\mathbf{y}) \phi(\mathbf{y}, \tau) \quad (7.15)$$

Then, Equation 7.14 can be rewritten as follows:

$$\begin{aligned} \frac{\partial}{\partial \tau} \left(\frac{f(\mathbf{y}, \tau)}{\phi_T(\mathbf{y})} \right) &= \sum_{k=1}^M \frac{\hbar}{2m_k} \nabla_k^2 \left(\frac{f(\mathbf{y}, \tau)}{\phi_T(\mathbf{y})} \right) - v(\mathbf{y}) \left(\frac{f(\mathbf{y}, \tau)}{\phi_T(\mathbf{y})} \right) \\ \frac{\partial}{\partial \tau} f(\mathbf{y}, \tau) &= \sum_{k=1}^M \frac{\hbar}{2m_k} \phi_T(\mathbf{y}) \nabla_k^2 \left(\frac{f(\mathbf{y}, \tau)}{\phi_T(\mathbf{y})} \right) - v(\mathbf{y}) f(\mathbf{y}, \tau) \end{aligned} \quad (7.16)$$

This can be further simplified by considering the following:

$$\begin{aligned} \nabla_k^2 \left(\frac{f}{\phi_T} \right) &= \nabla_k \cdot \left(\frac{\phi_T \nabla_k f - f \nabla_k \phi_T}{\phi_T^2} \right) \\ &= \frac{\phi_T^2 \nabla_k (\phi_T \nabla_k f - f \nabla_k \phi_T) - (\phi_T \nabla_k f - f \nabla_k \phi_T) \nabla_k (\phi_T^2)}{\phi_T^4} \\ &= \frac{\nabla_k^2 f}{\phi_T} - \frac{f \nabla_k^2 \phi_T}{\phi_T^2} - \frac{2 \nabla_k \phi_T \nabla_k f}{\phi_T^2} + \frac{2 f (\nabla_k \phi_T)^2}{\phi_T^3} \end{aligned}$$

Let the following quantities be defined:

$$b(\mathbf{y}) = \frac{H \phi_T(\mathbf{y})}{\phi_T(\mathbf{y})}, \text{ trial energy function.} \quad (7.17)$$

$$a_k(\mathbf{y}) = \frac{\hbar}{m_k} \cdot \frac{\nabla_k \phi_T(\mathbf{y})}{\phi_T(\mathbf{y})}, \quad (k = 1, \dots, M), \text{ local drift functions.} \quad (7.18)$$

Then Equation 7.16 can be rewritten as follows:

$$\frac{\partial}{\partial \tau} f(\mathbf{y}, \tau) = \sum_{k=1}^M \frac{\hbar}{2m_k} \nabla_k^2 f(\mathbf{y}, \tau) - \sum_{k=1}^M \nabla_k \cdot (\mathbf{a}_k(\mathbf{y}) f(\mathbf{y}, \tau)) - b(\mathbf{y}) f(\mathbf{y}, \tau) \quad (7.19)$$

This is the governing equation for the trial function reformulation of the system in imaginary time. It converges not to ψ_0 , but rather to $\phi_T \psi_0$, which will actually turn out to be beneficial. The equation is similar to Equation 7.14, but with an additional set of terms involving 1st derivatives of f . As will be shown, these extra terms correspond to a drift in the associated diffusion process with the drift of the k 'th body given by the vector valued function, \mathbf{a}_k . Hence the name “local drift functions.” The most important change is that in the final term, $v(\mathbf{y})$ has been replaced with $b(\mathbf{y})$, the trial energy function. If the trial function has been well chosen, then $b(\mathbf{y})$ will have a significantly lower variance than $v(\mathbf{y})$ and the singularities of $v(\mathbf{y})$ will also not be present in $b(\mathbf{y})$. In particular, if one were to hypothetically use exactly ψ_0 as the trial function, then $b(\mathbf{y})$ would be a constant valued function with zero variance. In reality one uses a trial function which is as close as possible to unknown ψ_0 in order to reduce the variance of $b(\mathbf{y})$ as much as possible. The next section will indicate how the ground state energy can be estimated from the values of $b(\mathbf{y})$.

7.3 Theoretical Ground State Energy Approximations

The solutions to the evolution equations, Equations 7.14 and 7.19, can be used in several ways to approximate the ground state energy, E_0 , of the physical system. Monte Carlo estimates, described in Section 7.5, will be based on such approximations. In this section references to convergence are meant to be in terms of imaginary time, τ , as it increases to infinity.

One such approximation is known as the *growth estimate*. It is used primarily with the solution of Equation 7.14, which does not involve a trial function. Consider that if the solution has converged to the exact ground state, i.e. $\phi(\mathbf{y}, \tau) = \psi_0(\mathbf{y})$, then by Equation 7.8 the following would be true:

$$\phi(\mathbf{y}, \tau + \Delta\tau) = e^{-\frac{1}{\hbar} E_0 \Delta\tau} \phi(\mathbf{y}, \tau)$$

Integrate over \mathbf{y} and solve for E_0 to obtain the following equation:

$$E_0 = -\frac{\hbar}{\Delta\tau} \log \left(\frac{\int \phi(\mathbf{y}, \tau + \Delta\tau) d\mathbf{y}}{\int \phi(\mathbf{y}, \tau) d\mathbf{y}} \right) , \quad \text{if } \phi(\mathbf{y}, \tau) = \psi_0(\mathbf{y}).$$

Then, by Equation 7.13, as τ increases $\phi(\mathbf{y}, \tau)$ approaches $\psi_0(\mathbf{y})$. Therefore, in general, assuming that $a_0 \neq 0$, we have the following useful result:

$$E_0 = \lim_{\tau \rightarrow \infty} \left\{ -\frac{\hbar}{\Delta\tau} \log \left(\frac{\int \phi(\mathbf{y}, \tau + \Delta\tau) d\mathbf{y}}{\int \phi(\mathbf{y}, \tau) d\mathbf{y}} \right) \right\}$$

In practice, the integrals in the above equation are estimated using Monte Carlo methods and stochastic simulations similar to those which are discussed in the later sections of this chapter. However, I will focus primarily on the second type of estimate below because it is generally much more accurate.

Note that since E_0 is, by definition, the lowest energy state of the system, the approximations should tend to converge to E_0 from above. This allows one to decide how high τ has to be for convergence by waiting until the energy estimates cease to steadily decline and instead simply fluctuate around the exact value of E_0 due to statistical errors in the Monte Carlo estimates.

A second way to approximate E_0 is called the *variational energy estimate*. It is particularly well suited to the solution of Equation 7.19, with a trial function, though it can also be used when no trial function is involved in the evolution of the system. It is based on the orthonormality conditions of the eigenfunctions of the Hamiltonian operator H . Consider the following:

Let $\phi_*(\mathbf{y})$ be an arbitrary wavefunction with a non-zero ground state component. By Equation 7.9, we then have the following:

$$\int_{\mathbb{R}^{3M}} \phi_*(\mathbf{y}) \psi_0(\mathbf{y}) d\mathbf{y} = a_0 \neq 0$$

Next use Equation 7.4 and orthonormality to obtain the equation:

$$\int_{\mathbb{R}^{3M}} H \phi_*(\mathbf{y}) \psi_0(\mathbf{y}) d\mathbf{y} = E_0 a_0$$

Therefore, solving for E_0 , we have the following exact formula:

$$E_0 = \frac{\int_{\mathbb{R}^{3M}} H \phi_*(\mathbf{y}) \psi_0(\mathbf{y}) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} \phi_*(\mathbf{y}) \psi_0(\mathbf{y}) d\mathbf{y}}$$

This is approximated for large τ by defining the quantity:

$$E(\tau) = \frac{\int_{\mathbb{R}^{3M}} H \phi_*(\mathbf{y}) \phi(\mathbf{y}, \tau) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} \phi_*(\mathbf{y}) \phi(\mathbf{y}, \tau) d\mathbf{y}} \quad (7.20)$$

Then, using Equation 7.13, one can show the following:

$$\begin{aligned}
\lim_{\tau \rightarrow \infty} E(\tau) &= \lim_{\tau \rightarrow \infty} \left\{ \frac{\int_{\mathbb{R}^{3M}} H \phi_*(\mathbf{y}) \phi(\mathbf{y}, \tau) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} \phi_*(\mathbf{y}) \phi(\mathbf{y}, \tau) d\mathbf{y}} \right\} \\
&= \frac{\int_{\mathbb{R}^{3M}} H \phi_*(\mathbf{y}) \psi_0(\mathbf{y}) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} \phi_*(\mathbf{y}) \psi_0(\mathbf{y}) d\mathbf{y}} \\
&= E_0
\end{aligned} \tag{7.21}$$

This becomes especially convenient for the system governed by Equation 7.19 with trial function, ϕ_T , if one uses ϕ_T as ϕ_* in the expressions above. Then, Equation 7.20 can be rewritten as follows:

$$\begin{aligned}
E(\tau) &= \frac{\int_{\mathbb{R}^{3M}} H \phi_T(\mathbf{y}) \phi(\mathbf{y}, \tau) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} \phi_T(\mathbf{y}) \phi(\mathbf{y}, \tau) d\mathbf{y}} \\
&= \frac{\int_{\mathbb{R}^{3M}} \left(\frac{H \phi_T(\mathbf{y})}{\phi_T(\mathbf{y})} \right) \phi_T(\mathbf{y}) \phi(\mathbf{y}, \tau) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} \phi_T(\mathbf{y}) \phi(\mathbf{y}, \tau) d\mathbf{y}} \\
&= \frac{\int_{\mathbb{R}^{3M}} \left(\frac{H \phi_T(\mathbf{y})}{\phi_T(\mathbf{y})} \right) f(\mathbf{y}, \tau) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} f(\mathbf{y}, \tau) d\mathbf{y}} \\
E(\tau) &= \frac{\int_{\mathbb{R}^{3M}} b(\mathbf{y}) f(\mathbf{y}, \tau) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} f(\mathbf{y}, \tau) d\mathbf{y}}
\end{aligned} \tag{7.22}$$

Thus, in this case, $E(\tau)$ is equal to the expected value of $b(\mathbf{y})$ with respect to $f(\mathbf{y}, \tau)$ as a normalized probability density, where $b(\mathbf{y})$ is the local trial energy which was defined by Equation 7.17. This expectation is estimated using the stochastic simulation and Monte Carlo methods of the next several sections. As discussed in the last section, $b(\mathbf{y})$ is nearly constant when $\phi_T(\mathbf{y})$ is close to $\psi_0(\mathbf{y})$. As a result Monte Carlo estimates based on the variational energy often have a significantly lower variance than growth energy estimates, and are therefore usually preferred.

There is an important assumption being made in the paragraph above, which is that $f(\mathbf{y}, \tau)$ is strictly positive. Otherwise, it would not make sense to use f as a probability density function. In fact, in certain physical systems, $f(\mathbf{y}, \tau)$ can become negative, and this leads to difficult problems with regards to all types of Monte Carlo estimates. This is an area of a great deal of active research (see for example Anderson and Traynor [4], Ceperly and Alder [9], and Zhang and Kalos [63]). However, in the examples which I have considered the value of $f(\mathbf{y}, \tau)$ is strictly positive. Also, in all of the discussion to follow, I will assume that $f(\mathbf{y}, \tau)$ is strictly positive. I anticipate that the methods I develop here will also be useful in cases where f does become negative, but that will have to remain an area of possible future research.

7.4 Connections to Stochastic Differential Equations

Corresponding to the partial differential equations, Equations 7.14 and 7.19, are diffusion processes which can be described by systems of stochastic differential equations (SDE's) such as those introduced in Section 6.3.2.

Physicists and Chemists often describe the diffusion processes in terms other than the stochastic differential framework, using such terms as transfer matrices, commutators, and averaging over ensembles of particles, but their resultant simulations are generally equivalent to those which I will obtain using the mathematical formalities of stochastic differential equation theory.

Recall Equation 7.19:

$$\frac{\partial}{\partial \tau} f(\mathbf{y}, \tau) = \sum_{k=1}^M \frac{\hbar}{2m_k} \nabla_k^2 f(\mathbf{y}, \tau) - \sum_{k=1}^M \nabla_k \cdot (\mathbf{a}_k(\mathbf{y}) f(\mathbf{y}, \tau)) - b(\mathbf{y}) f(\mathbf{y}, \tau)$$

This is rewritten explicitly in $3M$ dimensions, with t replacing τ for simplicity, as follows:

$$\frac{\partial}{\partial t} f(\mathbf{y}, t) = \sum_{i=1}^{3M} \frac{1}{2} \sigma_i^2 \frac{\partial^2}{\partial y_i^2} f(\mathbf{y}, t) - \sum_{i=1}^{3M} \frac{\partial}{\partial y_i} (a_i(\mathbf{y}) f(\mathbf{y}, t)) - b(\mathbf{y}) f(\mathbf{y}, t) \quad (7.23)$$

where $\sigma_i = \sqrt{\frac{\hbar}{m_k}}$, $a_i = \mathbf{a}_k^{(j)}$, $k = \text{Trunc}\left(\frac{i+2}{3}\right)$, and $j = 1 + (i-1) \bmod 3$ so that i refers to the j 'th component of the k 'th body (see Equation 7.1).

Now we define a related stochastic process. First, a note about normalization. Earlier, in Section 7.1, wavefunction were normalized so that $\int_{\mathbb{R}^{3M}} \tilde{\phi}(\mathbf{y}, t)^2 d\mathbf{y} = 1$ (see Equation 7.10). Here, however, a wavefunction, $f(\mathbf{y}, 0)$, will act as an initial probability density function (pdf). Therefore, its appropriate normalization is given as follows: $\int_{\mathbb{R}^{3M}} f(\mathbf{y}, 0) d\mathbf{y} = 1$. This is often a source of confusion. However, since H is a linear operator for which we are seeking an eigenvalue, namely E_0 , multiplication by constants doesn't really affect the results.

Consider the vector-valued stochastic process:

$$\mathbf{Z}(0) \in f(\mathbf{y}, 0) \text{ , } Y(0) = 0 \quad (7.24)$$

$$d\mathbf{Z}_i(t) = a_i(\mathbf{Z}(t)) dt + \sigma_i dW \text{ , } (i = 1, \dots, 3M) \quad (7.25)$$

$$dY(t) = b(\mathbf{Z}(t)) dt \quad (7.26)$$

where $\mathbf{Z}(t) = (Z_1(t), \dots, Z_{3M}(t))$.

This is a system of stochastic differential equations (SDE's). See Section 6.3.2 for more details. As will be explained in Section 7.5, the coefficients σ_i are the component-wise diffusion constants, while $a_i(Z(t))$ are the component-wise local drift functions. The function $b(Z(t))$ acts as a local growth/decay rate for the extra stochastic variable Y which is interpreted as a 'weight' factor in terms of total growth/decay along the stochastic path of Z .

The following theorem, which is closely related to the Feynman-Kac Formula (see Section 6.3.3), expresses the connection between the solution of Equation 7.23 and the stochastic process defined by the SDE's of Equations 7.24-7.26. We will need the vector version of Ito's Formula (see Gard [15] for derivation) which can be stated as follows (Note: each σ_i is a constant) :

If

$$dX_i(t) = a_i(X(t)) dt + \sigma_i dW \quad , \quad (i = 1, \dots, d)$$

Then

$$\begin{aligned} df(X(t)) = & \left(\sum_{i=1}^d \frac{1}{2} \sigma_i^2 \frac{\partial^2}{\partial x_i^2} f(X(t)) + \sum_{i=1}^d a_i \frac{\partial}{\partial x_i} f(X(t)) \right) dt \\ & + \left(\sum_{i=1}^d \sigma_i \frac{\partial}{\partial x_i} f(X(t)) \right) dW \end{aligned}$$

Theorem 3 *Let $(Z(t), Y(t))$ be defined by the SDE's of Equations 7.24-7.26, and let $p_z(\mathbf{y}, t) = \int_{\mathbb{R}^M} p_z(\mathbf{x}, 0, \mathbf{y}, t) f(\mathbf{x}, 0) d\mathbf{x}$ be the probability density for $Z(t) = \mathbf{y}$.*

(Note: $p_z(\mathbf{x}, 0, \mathbf{y}, t)$ is the transition density function for process $Z(t)$, and $f(\mathbf{x}, 0)$ is acting as its initial position distribution. See Section 6.3.1 for details.)

Define the following:

$$g(\mathbf{y}, t) = E \left\{ e^{-Y(t)} | Z(t) = \mathbf{y} \right\} p_z(\mathbf{y}, t) \quad (7.27)$$

Then, for any test function, $v(\mathbf{y})$, which is square integrable and twice differentiable, we have the following two results:

(i)

$$\begin{aligned} \int_{\mathbb{R}^{3M}} v(\mathbf{y}) \left(\frac{\partial}{\partial t} g(\mathbf{y}, t) \right) d\mathbf{y} = \\ \int_{\mathbb{R}^{3M}} v(\mathbf{y}) \left(\sum_{i=1}^{3M} \frac{1}{2} \sigma^2 \frac{\partial^2}{\partial y_i^2} g(\mathbf{y}, t) - \sum_{i=1}^{3M} \frac{\partial}{\partial y_i} [a_i(\mathbf{y}) g(\mathbf{y}, t)] - b(\mathbf{y}) g(\mathbf{y}, t) \right) d\mathbf{y} \end{aligned} \quad (7.28)$$

(i.e. $g(\mathbf{y}, t)$ satisfies Equation 7.23 in a weak or variational sense.), and

(ii)

$$\int_{\mathbb{R}^{3M}} v(\mathbf{y}) g(\mathbf{y}, t) d\mathbf{y} = E \left[v(Z(t)) e^{-Y(t)} \right] \quad (7.29)$$

where the expectation is taken over all possible realizations of the stochastic processes.

Proof:

Let $V(t) = E [v(Z(t)) e^{-Y(t)}]$. Then, using the definition of expectation and conditional probabilities, we have the following:

$$\begin{aligned} V(t) &= \int_{\mathbb{R}^{3M}} v(\mathbf{y}) E \left(e^{-Y(t)} | Z(t) = \mathbf{y} \right) p_z(\mathbf{y}, t) d\mathbf{y} \\ &= \int_{\mathbb{R}^{3M}} v(\mathbf{y}) g(\mathbf{y}, t) d\mathbf{y} \end{aligned}$$

This immediately proves part (ii).

Now consider,

$$\begin{aligned}\frac{dV(t)}{dt} &= \frac{d}{dt} \left(\int_{\mathfrak{R}^{3M}} v(\mathbf{y}) g(\mathbf{y}, t) d\mathbf{y} \right) \\ &= \int_{\mathfrak{R}^{3M}} v(\mathbf{y}) \frac{\partial}{\partial t} g(\mathbf{y}, t) d\mathbf{y}\end{aligned}\tag{7.30}$$

On the other hand, by definition,

$$\begin{aligned}\frac{dV(t)}{dt} &= \lim_{h \rightarrow 0} \left(\frac{v(t+h) - v(t)}{h} \right) \\ &= \lim_{h \rightarrow 0} \frac{E[v(\mathbf{Z}(t+h))e^{-Y(t+h)}] - E[v(\mathbf{Z}(t))e^{-Y(t)}]}{h} \\ &= E \left\{ \lim_{h \rightarrow 0} \left(\frac{v(\mathbf{Z}(t+h))e^{-Y(t+h)} - v(\mathbf{Z}(t))e^{-Y(t)}}{h} \right) \right\} \\ &= E \left\{ \lim_{h \rightarrow 0} \left(\frac{(v(\mathbf{Z}(t+h)) - v(\mathbf{Z}(t))e^{-Y(t)})}{h} \right) \right. \\ &\quad \left. + \lim_{h \rightarrow 0} \left(\frac{v(\mathbf{Z}(t+h)) (e^{-Y(t+h)} - e^{-Y(t)})}{h} \right) \right\} \\ &= E \left\{ \frac{dv(\mathbf{Z}(t))}{dt} e^{-Y(t)} - v(\mathbf{Z}(t)) \frac{dY(t)}{dt} e^{-Y(t)} \right\}\end{aligned}$$

Apply Ito's Formula:

$$\begin{aligned}&= E \left\{ \left(\sum_{i=1}^{3M} \frac{1}{2} \sigma_i^2 \frac{\partial^2}{\partial y_i^2} v(\mathbf{Z}(t)) + \sum_{i=1}^{3M} a_i(\mathbf{Z}(t)) \frac{\partial}{\partial y_i} v(\mathbf{Z}(t)) \right. \right. \\ &\quad \left. \left. + \sum_{i=1}^{3M} \sigma_i \frac{\partial}{\partial y_i} v(\mathbf{Z}(t)) \frac{dW}{dt} \right) e^{-Y(t)} - b(\mathbf{Z}(t)) v(\mathbf{Z}(t)) e^{-Y(t)} \right\}\end{aligned}$$

The 'stochastic' terms in this expectation containing $\frac{dW}{dt}$, which is the White Noise Process, all drop out since the expected value of each of them is zero. (See Sections 6.3.1-6.3.3 for details.)

Define the following linear operators:

$$\begin{aligned}L[v](\mathbf{Z}(t)) &= \left(\sum_{i=1}^{3M} \frac{1}{2} \sigma_i^2 \frac{\partial^2}{\partial y_i^2} + \sum_{i=1}^{3M} a_i(\mathbf{Z}(t)) \frac{\partial}{\partial y_i} - b(\mathbf{Z}(t)) \right) v(\mathbf{Z}(t)) \\ L^*[v](\mathbf{Z}(t)) &= \left(\sum_{i=1}^{3M} \frac{1}{2} \sigma_i^2 \frac{\partial^2}{\partial y_i^2} - \sum_{i=1}^{3M} \frac{\partial}{\partial y_i} a_i(\mathbf{Z}(t)) - b(\mathbf{Z}(t)) \right) v(\mathbf{Z}(t)), \text{ adjoint.}\end{aligned}$$

Then we can continue as follows:

$$\begin{aligned}
\frac{dV(t)}{dt} &= E \left[L[v](Z(t)) e^{-Y(t)} \right] \\
&= \int_{\mathbb{R}^{3M}} L[v](\mathbf{y}) E \left(e^{-Y(t)} \mid Z(t) = \mathbf{y} \right) p_z(\mathbf{y}, t) d\mathbf{y} \\
&= \int_{\mathbb{R}^{3M}} L[v](\mathbf{y}) g(\mathbf{y}, t) d\mathbf{y}
\end{aligned}$$

Finally, using integration by parts, we have:

$$\frac{dV(t)}{dt} = \int_{\mathbb{R}^{3M}} v(\mathbf{y}) L^*[g](\mathbf{y}, t) d\mathbf{y} \quad (7.31)$$

Therefore, combining Equations 7.30 and 7.31 gives us the following:

$$\int_{\mathbb{R}^{3M}} v(\mathbf{y}) \frac{\partial}{\partial t} g(\mathbf{y}, t) d\mathbf{y} = \int_{\mathbb{R}^{3M}} v(\mathbf{y}) L^*[g](\mathbf{y}, t) d\mathbf{y}$$

This is the desired equality for part (i).

Note also that $g(\mathbf{y}, 0) = f(\mathbf{y}, 0)$ trivially satisfies the initial conditions for Equation 7.23. Thus, $g(\mathbf{y}, t)$ is a weak solution of Equation 7.23. •

The same sort of argument can be made for the solution of the quantum system without a trial function, Equation 7.14. The appropriate stochastic processes are governed by SDE's which are like Equations 7.24-7.26, but without any drift terms and with b replaced by v , which can be much more poorly behaved (higher variations and singular). I will not deal with this case because of the strong similarities with the trial function case and because the trial function reformulation seems to be much more commonly used.

Using the above theorem, the variational energy estimate, $E(t)$, with t replacing τ , for convenience as in Equation 7.23, can now be computed using the stochastic

processes as follows:

$$\begin{aligned}
 E(t) &= \frac{\int_{\mathfrak{R}_{3M}} b(\mathbf{y}) f(\mathbf{y}, t) d\mathbf{y}}{\int_{\mathfrak{R}_{3M}} f(\mathbf{y}, t) d\mathbf{y}} \\
 &= \frac{E[b(\mathbf{Z}(t))e^{-Y(t)}]}{E[e^{-Y(t)}]}, \quad \text{over stochastic paths } (\mathbf{Z}, Y). \quad (7.32)
 \end{aligned}$$

This expectation will be estimated using Monte Carlo estimates (Section 7.6) over repeated simulations of the stochastic system using finite time steps (Section 7.5). Finally, the application of quasi-random sequences to these Monte Carlo estimates and simulation will be discussed (Section 7.7).

7.5 RMS Convergence, Time Discrete Stochastic Simulations

Let $\|X - Y\|_{rms} = \sqrt{E((X - Y)^2)}$ define the root mean square norm of the difference $X - Y$, where X and Y are random variables. For our purposes X and Y could be the values of two different stochastic processes at the same time or the values of a single stochastic process at two different times.

Then we define the following properties of a family of stochastic processes $X^{(h)}(t)$ intended to approximate $X(t)$:

$$\begin{aligned} \text{If} \quad & \lim_{h \rightarrow 0} \|X^{(h)}(t) - X(t)\|_{rms} = 0 \quad , \quad \text{at } t, \\ \text{then} \quad & X^{(h)}(t) \xrightarrow{rms} X(t) \quad , \quad \text{at } t. \end{aligned}$$

This is called rms-convergence.

$$\begin{aligned} \text{If} \quad & \lim_{h \rightarrow 0} \frac{\|X^{(h)}(t) - X(t)\|_{rms}}{h^\alpha} < \infty \quad , \quad \text{at } t, \\ \text{then} \quad & X^{(h)}(t) - X(t) = O_{rms}(h^\alpha) \quad , \quad \text{at } t. \end{aligned}$$

This is called rms-convergence of order α , or order α local accuracy.

One can also measure the magnitude of a stochastic quantity in terms of the corresponding rms-norm defined as follows:

$$\|X\|_{rms} = \sqrt{E(X^2)}$$

In particular, a few important results are as follows:

$$\|W(t)\|_{rms} = \sqrt{E(W(t)^2)} = t^{1/2}$$

$$\begin{aligned}
\left\| \int_0^t W(s) ds \right\|_{rms} &= \sqrt{E \left[\left(\int_0^t W(s) ds \right)^2 \right]} \\
&= \sqrt{E \left[\int_0^t \int_0^t W(s) W(r) dr ds \right]} \\
&= \sqrt{\int_0^t \int_0^t E(W(s) W(r)) dr ds} \\
&= \sqrt{\int_0^t \int_0^t \min\{s, r\} dr ds} \\
&= \sqrt{2 \int_0^t \int_0^s r dr ds} \\
&= \sqrt{\frac{1}{3} t^3} = \frac{1}{\sqrt{3}} t^{3/2}
\end{aligned}$$

This can be continued in the same fashion to show the following:

$$\int_0^h dW_s = W(h) = O_{rms}(h^{1/2}) \quad (7.33)$$

$$\int_0^h \int_0^s dW_r ds = \int_0^h W(s) ds = O_{rms}(h^{3/2}) \quad (7.34)$$

$$\int_0^h \int_0^s \int_0^r dW_q dr ds = \int_0^h \int_0^s W(r) dr ds = O_{rms}(h^{5/2}) \quad (7.35)$$

⋮

It can similarly be shown that the following are also true (Gard [15, pg.42]):

$$\int_0^h \int_0^s dW_r dW_s = \int_0^h W(s) dW_s = O_{rms}(h) \quad (7.36)$$

$$\int_0^h \int_0^s \int_0^r dW_q dW_r ds = \int_0^h \int_0^s W(r) dW_s ds = O_{rms}(h^2) \quad (7.37)$$

In general, we have order $O_{rms}(h^{\frac{a}{2}+b})$ when there are a stochastic differentials, dW , and b ordinary differentials, dt . (See Kloeden and Platen [32].)

The quantities on the left hand sides of the equations above are known as “Wiener iterates” and are important in deriving high order time discrete stochastic approximations, as we shall see.

Now, a stochastic analogue to the Taylor Expansion is derived using the Ito Formula. Its truncation errors can be interpreted in the rms sense as above. This will be for one dimension, but the same concept works in higher dimensions. Also for simplicity, but without loss of generality, I will assume that the starting time is $t = 0$.

Stochastic (Ito) Taylor Expansion:

Consider stochastic process $X(t)$ given by the following SDE:

$$dX(t) = a(x(t)) dt + \sigma dW$$

Then, given $X(t)$ at any t , one can express $X(t+h)$ as follows:

$$X(t+h) = X(t) + \int_t^{t+h} a(X(s)) ds + \sigma(W(t+h) - W(t))$$

(Note: refer to Section 6.3.2 for further explanation.) Assuming, without loss of generality, that $t = 0$ and $X(0) = 0$, this becomes:

$$X(h) = \int_0^h a(X(s)) ds + \sigma W(h) \quad (7.38)$$

Recall the Ito Formula:

$$df(X(t)) = [f'(X(t))a(X(t)) + \frac{1}{2}\sigma^2 f''(X(t))] dt + \sigma f'(X(t)) dW$$

Therefore, in general, $f(X(h))$ can be expanded as follows:

$$f(X(h)) = f(X(0)) + \int_0^h (f'a + \frac{1}{2}\sigma^2 f'')(X(s)) ds + \int_0^h \sigma f'(X(s)) dW_s$$

Now expand $a(X(s))$ according to the expansion above:

$$a(X(s)) = a(0) + \int_0^s (a'a + \frac{1}{2}\sigma^2 a'')(X(r)) dr + \int_0^s \sigma a'(X(r)) dW_r$$

Insert this formula for a into Equation 7.38:

$$\begin{aligned} X(h) &= \int_0^h \left(a(0) + \int_0^s \zeta(X(r)) dr + \int_0^s \sigma a'(X(r)) dW_r \right) ds + \sigma W(h) \\ &= \sigma W(h) + a(0) h + \int_0^h \int_0^s \sigma a'(X(r)) dW_r ds + \int_0^h \int_0^s \zeta(X(r)) dr ds \end{aligned}$$

where $\zeta(X(r)) = (a'a + \frac{1}{2}\sigma^2 a'')(X(r))$.

Next, expand $\zeta(X(r))$ and $\sigma a'(X(r))$ in similar fashion:

$$\begin{aligned} \sigma a'(X(r)) &= \sigma a'(0) + \int_0^r \sigma(a''a + \frac{1}{2}\sigma^2 a''')(X(q)) dq + \int_0^r \sigma^2 a''(X(q)) dW_q \\ \zeta(X(r)) &= \zeta(0) + \int_0^r (\zeta'a + \frac{1}{2}\sigma^2 \zeta'')(X(q)) dq + \int_0^r \sigma \zeta'(X(q)) dW_q \end{aligned}$$

Inserting these expansion into the expression for $X(h)$, integrating, and using Equations 7.33-7.37, we obtain the following:

$$\begin{aligned} X(h) &= \sigma W(h) + a(0) h \\ &\quad + \int_0^h \int_0^s \left[\sigma a'(0) + \int_0^r \alpha(X(q)) dq + \int_0^r \sigma^2 a''(X(q)) dW_q \right] dW_r ds \\ &\quad + \int_0^h \int_0^s \left[\zeta(0) + \int_0^r \beta(X(q)) dq + \int_0^r \sigma \zeta'(X(q)) dW_q \right] dr ds \\ &= \sigma W(h) + a(0) h + \sigma a'(0) \int_0^h W(s) ds + \frac{1}{2} \zeta(0) h^2 \\ &\quad + \sigma^2 a''(0) \int_0^h \int_0^s W(r) dW_r ds + O_{rms}(h^{5/2}) \end{aligned}$$

where $\alpha = \sigma a''a + \frac{1}{2}\sigma^3 a'''$ and $\beta = \zeta'a + \frac{1}{2}\sigma^2 \zeta''$.

This Stochastic Taylor Series is similar to the ordinary (deterministic) Taylor Series. However, the terms increase by half orders of h instead of whole orders. In particular, we have the following:

$$O_{rms}(h^{1/2}) : \sigma W(h)$$

$$\begin{aligned}
O_{rms}(h) &: a(0) h \\
O_{rms}(h^{3/2}) &: \sigma a'(0) \int_0^h W(s) ds \\
O_{rms}(h^2) &: \frac{1}{2} \zeta(0) h^2 + \sigma^2 a''(0) \int_0^h \int_0^s W(r) dW_r ds
\end{aligned}$$

This process of successive expansions can be continued as far as needed to obtain higher order terms of the Stochastic Taylor Series. This series is the basis for a wide variety of time discretization methods for approximating the solutions of SDE's. See Kloeden and Platen [32] for numerous examples.

Now we can define a time discrete approximation for the system $(Z(t), Y(t))$ at any arbitrary time t as follows:

Select the number of equal time steps m , and let $h = \frac{t}{m}$ be the size of the time steps, and $t_j = jh$, ($j = 0, \dots, m$) be the discrete times at which the process will be approximated. h should be small for low discretization errors.

Then, the *Euler-Maruyama* approximation scheme is defined as follows:

$$Z^{(0)} \in \phi_T^2(\mathbf{y}) \quad , \quad Y^{(0)} = 0 \quad (7.39)$$

$$Z_i^{(j)} = Z_i^{(j-1)} + \sigma_i \Delta W_i^{(j)} + h a_i(Z^{(j-1)}) \quad (7.40)$$

$$Y^{(j)} = Y^{(j-1)} + h b(Z^{(j-1)}) \quad (7.41)$$

with $i = 1, \dots, 3M$, $j = 1, \dots, m$, $\Delta W_i^{(j)} \in \mathcal{N}(0, h)$.

Note that $\phi_T^2(\mathbf{y})$, normalized as a pdf, is used for $f(\mathbf{y}, 0)$. This is intended to be the best available approximation to $\phi_T(\mathbf{y})\psi_0(\mathbf{y})$, which is the function to which the solution of Equation 7.23 converges. The better this approximation is, the less time (fewer steps) needed to reach this ground state, $\phi_T \psi_0$.

This approximation scheme has local truncation error of order $O_{rms}(h^{3/2})$ by the Stochastic Taylor Series. However, the errors at each step add up to give a global error at time t of order $O_{rms}(h^{1/2})$ as follows:

$$m O_{rms}(h^{3/2}) = mh O_{rms}(h^{1/2}) = t O_{rms}(h^{1/2})$$

It should now be clear from Equation 7.40 why the coefficients a_i are referred to as the local drift functions, and σ_i are the diffusion constants. Also since $Y(t)$ is interpreted physically as an exponential growth or decay factor in the term $e^{-Y(t)}$, we see that b in Equation 7.41 acts as the local growth/decay factor.

Most simulation methods for diffusion processes, including those used by physicists and chemists, are roughly equivalent to the above Euler-Maruyama scheme. Typically an “ensemble of random walkers” in $3M$ dimensional space are followed through time as they are drifted by $h a_i$, componentwise, and diffused by adding a sample $\sigma_i \Delta W \in \mathcal{N}(0, \sigma_i^2 h)$, a Wiener increment, componentwise. Growth or decay is modeled by either weighting or branching. For branching, walkers are killed with probability $b h$ when b is locally positive, or split in multiple branches if b is negative so that the expected number of branching walks is $1 + |b| h$. For weighting, the weight associated with each walk is multiplied by $1 - b h$, or by e^{-bh} , at each step. In either case, the effect is an estimation of $e^{-Y(t)}$ over each walker’s path. As they progress in time, the walkers’ positions produce a weighted sample from $f(\mathbf{y}, t)$ which approaches $\phi_T(\mathbf{y})\psi_0(\mathbf{y})$ exponentially fast (except for finite time step discretization error).

Although this simulation is only order $O_{rms}(h^{1/2})$ accurate in terms of the paths, it is actually of order $O(h)$ in a weaker sense in terms of estimating the correct expectation. This is related to the fact that $E\left(\int_0^h W_s ds\right) = 0$. So when expectations are taken, this error term drops out. (See Kloeden and Platen [32] for details of this weak convergence.) The result can be stated without explanation as follows:

Define the expected finite time step approximation as,

$$E^{(h)}(t) = \frac{E\left[b(Z^{(m)}) e^{-Y^{(m)}}\right]}{E\left[e^{-Y^{(m)}}\right]}, \text{ where } h = \frac{t}{m} \quad (7.42)$$

Recall the expression given by Equation 7.32 for the the variational energy estimate:

$$E(t) = \frac{E\left[b(Z(t)) e^{-Y(t)}\right]}{E\left[e^{-Y(t)}\right]}$$

Then, based on the results proven by Kloeden and Platen [32], one could write the following:

$$E^{(h)}(t) = E(t) + O(h) \quad (7.43)$$

where $O(h)$ is a simple deterministic order of convergence.

Computational results have confirmed this order of convergence for the expectations using the Euler-Maruyama Scheme on the examples I have examined.

Higher order accurate methods are possible by using further terms in the Stochastic Taylor Series. The quantity $\int_0^h W(s) ds$ can be shown to be a Gaussian distributed according to $\mathcal{N}(0, \frac{h^3}{3})$ which is correlated with $W(h)$ with the following covariance: $cov(W(h), \int_0^h W(s) ds) = \frac{h^2}{2}$. These two quantities can in fact be

sampled exactly in terms of two normally distributed variables, $g_1, g_2 \in \mathcal{N}(0, 1)$, as follows:

$$\begin{aligned} W(h) &= h^{1/2} g_1 \\ \int_0^h W(s) ds &= \frac{1}{2} h^{3/2} (g_1 + \frac{1}{\sqrt{3}} g_2) \end{aligned}$$

(Note: The details of the above results are derived and discussed in depth by Kloeden and Platen in [32].)

Thus, higher order methods typically require more than one Gaussian variable per time step. However, this is made up for by allowing for larger time steps. Also Runge-Kutta type approximations can be used for derivatives such as a' appearing in higher order terms.

For the examples at the end of this Chapter, I have used a method derived by Helfand and Greenside (see [26] and [18]) which is third order accurate in terms of the expected finite time step error, as in Equation 7.43. The method requires the use of two Gaussian variables per time step, but allows for significantly larger time steps without large discretization errors.

In order to avoid the details of this discretization method, which are not directly relevant to the rest of this thesis, I will use the following symbolic notation to represent the method, which will be referred to as **Helfand/Greenside 3**:

$$Z^{(0)} \in \phi_T^2(\mathbf{y}) \tag{7.44}$$

$$Y^{(0)} = 0 \tag{7.45}$$

$$Z^{(j)} = Z^{(j-1)} + \mathcal{D}_1 \left(\sigma, \mathbf{a}, \Delta W_1^{(j)}, \dots, \Delta W_{6M}^{(j)}, Z^{(j-1)} \right) \tag{7.46}$$

$$Y^{(j)} = Y^{(j-1)} + \mathcal{D}_2 \left(b, \sigma, \mathbf{a}, \Delta W_1^{(j)}, \dots, \Delta W_{6M}^{(j)}, \mathbf{Z}^{(j-1)} \right) \quad (7.47)$$

with $j = 1, \dots, m$, $\Delta W_k^{(j)} \in \mathcal{N}(0, h)$ for $k = 1, \dots, 6M$, $\mathbf{a} = (a_1, \dots, a_{3M})$, and $\sigma = (\sigma_1, \dots, \sigma_{3M})$.

As it was for the Euler-Maruyama Scheme, $\phi_T^2(\mathbf{y})$ is used for $f(\mathbf{y}, 0)$ here.

This scheme involves more work than the Euler-Maruyama scheme for each time step, and also twice as many Gaussians per time step. However, much larger time steps can be taken using Helfand/Greenside 3 for a given discretization error level. As we will see, the size of the time steps is important (they need to be as large as possible), when attempting to apply quasi-random sequences to Diffusion Monte Carlo estimates, which are discussed next.

7.6 Diffusion Monte Carlo Estimates

Monte Carlo estimates of the ground state energy, E_0 , are obtained by averaging over repeated samples of the stochastic process $(\mathbf{Z}(t), Y(t))$ generated using the time discretization methods of the last section. First, a review of some notation:

$$\begin{aligned} E_0 &= \text{unknown ground state energy, to be estimated.} \\ E(t) &= \frac{\int_{\mathbb{R}^{3M}} b(\mathbf{y}) f(\mathbf{y}, t) d\mathbf{y}}{\int_{\mathbb{R}^{3M}} f(\mathbf{y}, t) d\mathbf{y}}, \text{ trial energy estimate.} \\ &= \frac{E[b(\mathbf{Z}(t)) e^{-Y(t)}]}{E[e^{-Y(t)}]}, \text{ by Equation 7.32.} \\ E^{(h)}(t) &= \frac{E[b(\mathbf{Z}^{(m)}) e^{-Y^{(m)}}]}{E[e^{-Y^{(m)}}]}, \quad m = \frac{t}{h}, \text{ the \# of time steps.} \end{aligned}$$

Now, given a sample of stochastic results, $(Z_k^{(m)}, Y_k^{(m)})$, $k = 1, \dots, N$, define the following Monte Carlo estimate:

$$\hat{E}_N^{(h)} = \frac{\sum_{k=1}^N b(Z_k^{(m)}) e^{-Y_k^{(m)}}}{\sum_{k=1}^N e^{-Y_k^{(m)}}} \quad (7.48)$$

This is simply a ratio of two Crude Monte Carlo estimates. It is the **Diffusion Monte Carlo Estimate**, and it estimates E_0 as follows:

$$\begin{aligned} \hat{E}_N^{(h)} - E_0 &= (\hat{E}_N^{(h)} - E^{(h)}(t)) + (E^{(h)}(t) - E(t)) + (E(t) - E_0) \\ &= \epsilon_{mc} + \epsilon_{discr} + \epsilon_{distr} \\ &= O(N^{-1/2}) + O(h^\alpha) + O(e^{-(E_1 - E_0)t}) \quad , \quad \text{as } t, N \rightarrow \infty, h \rightarrow 0 \end{aligned} \quad (7.49)$$

where α is the order of accuracy of the time discretization method, ϵ_{mc} is the “Monte Carlo error,” ϵ_{discr} is the “discretization error,” and ϵ_{distr} is the “distribution error.”

The value of h is determined according to accuracy requirements, while N is typically as high as time constraints will allow. The value of t required to reduce the distribution error to acceptable levels depends on how close the system is initially to its ground state. For the trial function reformulated system with ϕ_T^2 used as the initial distribution (see Equation 7.39), this convergence is from ϕ_T^2 to $\phi_T \psi_0$, and the size of t required depends on how well ϕ_T approximates ψ_0 .

Even for a well chosen trial function, ϕ_T , though, the size of t needed is typically much larger than the time step size h . Therefore, the number of time steps $m = \frac{t}{h}$ can be quite high, even up to the hundreds.

However, as we will see next, when m is large, there is little hope of using quasi-random sequences directly to improve the estimates. Therefore, we will need to use a different approach involving the combination of pseudorandom and quasi-random sequences, which will be discussed in Section 7.7.

Consider the following representation of Diffusion Monte Carlo:

$$\begin{aligned} E(\hat{E}_N^{(h)}) &= \frac{E \left[b(\mathbf{Z}^{(m)}) e^{-Y^{(m)}} \right]}{E \left[e^{-Y^{(m)}} \right]} \\ &= \frac{E \left\{ b \left(\mathbf{Z}^{(0)} + \Delta \mathbf{Z}^{(1)} + \dots + \Delta \mathbf{Z}^{(m)} \right) e^{-Y^{(1)} - \Delta Y^{(2)} - \dots - \Delta Y^{(m)}} \right\}}{E \left\{ e^{-Y^{(1)} - \Delta Y^{(2)} - \dots - \Delta Y^{(m)}} \right\}} \end{aligned}$$

where $\Delta \mathbf{Z}^{(j)}$ represents the j 'th (vector) step along the stochastic path of \mathbf{Z} , and similarly for $\Delta Y^{(j)}$.

The quantities $\Delta \mathbf{Z}^{(j)}$ and $\Delta Y^{(j)}$ are computed according to the discretization scheme being used. In the case of Euler-Maruyama, $3M$ Gaussian variables are used in determining each $\Delta \mathbf{Z}^{(j)}$, $\Delta Y^{(j)}$ pair, while for Helfand/Greenside 3, twice as many, $6M$, are needed for each step. The Gaussians are obtained by applying a transformation method to the uniformly distributed points of a pseudorandom or quasi-random sequence. As discussed in Section 6.3.4, the choice of this transformation method can be an important factor, since it can add unwanted discontinuities. I will be using Marsaglia's method (Marsaglia [40]), which is continuous and monotonic. Let \mathcal{M} represent the Marsaglia transformation method, so that if $x_i \in U[0, 1]$, then $\mathcal{M}(x_i) \in \mathcal{N}(0, 1)$.

The initial location $\mathbf{Z}^{(0)}$ for each stochastic path also must be determined. Recall that these starting locations should be distributed like $\phi_T^2(\mathbf{y})$. This is generally

done using a method known as the Metropolis algorithm, which will not be dealt with here. Generally however, one can think of this as requiring an additional $3M$ uniformly distributed components, which will be transformed to the correct distribution, for each new path. Let this be represented symbolically by \mathcal{M}_0 , so that if $(x_1, \dots, x_{3M}) \in U([0, 1]^{3M})$, then $\mathcal{M}_0(x_1, \dots, x_{3M}) \in \phi_T^2$.

Now, referring back to the symbolic notation for the Helfand/Greenside 3 method, one can write the following, for $j = 1$ to m :

$$\begin{aligned}\Delta Z^{(j)} &= \mathcal{D}_1(\sigma, \mathbf{a}, \Delta W_{j,1}, \dots, \Delta W_{j,6M}, Z^{(j)}) \\ &= \mathcal{D}_1(\sigma, \mathbf{a}, \mathcal{M}(x_1^{(j)}), \dots, \mathcal{M}(x_{6M}^{(j)}), Z^{(j)}) \\ \Delta Y^{(j)} &= \mathcal{D}_1(b, \sigma, \mathbf{a}, \Delta W_{j,1}, \dots, \Delta W_{j,6M}, Z^{(j)}) \\ &= \mathcal{D}_1(b, \sigma, \mathbf{a}, \mathcal{M}(x_1^{(j)}), \dots, \mathcal{M}(x_{6M}^{(j)}), Z^{(j)})\end{aligned}$$

Also, represent the initial locations as follows:

$$\begin{aligned}Z^{(0)} &= \mathcal{M}_0(x_1^{(0)}, \dots, x_{3M}^{(0)}) \\ Y^{(0)} &= 0\end{aligned}$$

Then suppressing all of the arguments of each function except for the x 's, we can express:

$$\begin{aligned}E(\hat{E}_N^{(h)}) &= \frac{A}{B} \\ A &= \int_{I^d} \{b(\mathcal{M}_0(x_1, \dots, x_{3M}) + \mathcal{D}_1(\mathcal{M}(x_{3M+1}), \dots, \mathcal{M}(x_{9M})) + \dots) \\ &\quad \cdot e^{-\mathcal{D}_2(\mathcal{M}(x_{3M+1}), \dots, \mathcal{M}(x_{9M})) - \mathcal{D}_2(\mathcal{M}(x_{9M+1}), \dots, \mathcal{M}(x_{15M})) - \dots}\} dx_1 \dots dx_d \\ B &= \int_{I^d} \{e^{-\mathcal{D}_2(\mathcal{M}(x_{3M+1}), \dots, \mathcal{M}(x_{9M})) - \mathcal{D}_2(\mathcal{M}(x_{9M+1}), \dots, \mathcal{M}(x_{15M}^{(2)})) - \dots}\} dx_1 \dots dx_d\end{aligned}$$

where $d = 3M + 6Mm$ and the x 's have all been renumbered in order as follows:

$$\begin{aligned} x_i^{(0)} &= x_i \\ x_i^{(j)} &= x_{3M+6M(j-1)+i} \end{aligned}$$

with $j = 1, \dots, m$ the time steps, and $i = 1, \dots, 6M$ the components needed for each time step.

Thus, the Diffusion Monte Carlo estimate is essentially a pair of very large dimensional integration estimates. It could even be thought of as a kind of weighted uniform sampling estimate (but I won't attempt any in depth analysis for this). The number of dimensions is $3M + 6Mm$, where M is the number of bodies in the quantum system and m is the number of time steps. This is assuming that two Gaussians are needed per time step, as for Helfand/Greenside 3. For Euler-Maruyama, the dimensions are only $3M + 3Mm$, but m will need to be significantly larger for the same discretization error level. Therefore, Helfand/Greenside 3 is preferred (at least for my purposes here). However, even with a higher order accurate approximation scheme, there are, in general, still too many dimensions for the direct application of quasi-random sequences to be effective.

For example, if $M = 2$ (two body system), then at most only about $m = 3$ or 4 time steps are possible before the number of dimensions is out of the optimal range for quasi-random sequences. This is well below the number of time steps typically required to reduce the distribution error, ϵ_{distr} , to acceptable levels, which could be in the hundreds, as mentioned before. Therefore, an entirely new approach is

needed, if quasi-random sequences are to be applied to Diffusion Monte Carlo.

The next section will describe two new methods for dealing with the problem of high dimensions by combining pseudorandom and quasi-random sequences. These will prove effective for Diffusion Monte Carlo when the number of bodies, M , is relatively small (possibly up to 3 or 4 at most), which includes a great many interesting systems nonetheless. If M is large (many body systems), purely pseudorandom sequences appear to remain the best option, at least so far, but when M is small, the methods discussed next can provide more accurate results.

7.7 Continuation and Empirical Transformation Methods

It is possible to combine pseudorandom and quasi-random sequences for Diffusion Monte Carlo in such a way as to obtain higher convergence rates than $O(N^{-1/2})$, at reasonable values of N . This is in contrast with the direct use of quasi-random sequences which would require astronomically high values of N to have any hope of improving the results because the number of dimensions necessary for good accuracy is typically so high. As discussed above, this is a result of the need for t to be relatively high in order to allow the distribution to converge to ground state (reducing ϵ_{distr}), combined with the need for the size of the time steps, h , to be relatively small in order to control the discretization errors (ϵ_{discr}).

The underlying concept of continuation is to first use a high dimensional pseudorandom sequence to generate stochastic path simulations (or “realizations of the

stochastic process”) for the process $(Z(t), Y(t))$ up to some large time, t_p . Since we are using a pseudorandom sequence, the number of iterations can be as large as necessary. The value of t_p should be chosen large enough so that ϵ_{distr} is insignificant relative to the statistical, or Monte Carlo, error ϵ_{mc} . One says then that the distribution has converged in time. Also, h should be small enough so that any discretization errors are also insignificant relative to the Monte Carlo error. (Note: The size of the Monte Carlo error depends, of course, on the sample size, N , so t_p and h should be selected accordingly.)

Next, these paths generated using pseudorandom sequences are “continued” for just a few additional time steps using quasi-random sequences. If done properly, this has the effect of reducing the Monte Carlo error, thereby producing more accurate results for a relatively small amount of extra work. The quasi-random sequences are able to work because with a low number of time steps only a moderate number of dimensions are needed.

The key point which allows this to work is that the set of endpoints of the pseudorandomly generated paths, $\{Z_k^{(m)}\}_{k=1}^N$, $m = \frac{t_p}{h}$, weighted by their integrated growth or decay factors, $e^{-Y_k^{(m)}}$, provides a highly accurate sample from the ground state of the trial function transformed system, Equation 7.19, given exactly by $\phi_T \psi_0$. In fact this sample can be made arbitrarily close by raising t_p , lowering h , and increasing N . The only practical limitation is cpu time.

Whereas, using quasi-random sequences directly with an initial transformation, such as \mathcal{M}_0 , to ϕ_T^2 requires many subsequent time steps depending on the quality

of ϕ_T , on the contrary the number of time steps required in the “continuation” part of the method above becomes very small as N increases (assuming that t_p and h have been set appropriately). This is because the paths have already converged to the ground state distribution, so only Monte Carlo errors remain, which are immediately reduced by taking a few steps using quasi-random sequences.

The main difficulty lies in setting up the “continuation,” since if it is not done properly, most of the benefits of the quasi-random sequences are lost, or if new approximation errors are introduced at this stage, more iterations will be required subsequently to eliminate them. I will explain what I mean by this shortly.

(Note: In the case that ϕ_T is itself an extremely accurate approximation to ψ_0 , then Diffusion Monte Carlo may not be needed at all, since one could simply use $E(0)$, the variational energy over ϕ_T^2 (see Equation 7.22), as an accurate approximation to E_0 . This can be evaluated as a simple $3M$ dimensional integral by ordinary Monte Carlo methods, to obtain what is known as a variational Monte Carlo estimate of E_0 . Such estimates are often used to obtain upper bounds on E_0 .)

I have found two effective ways of using the endpoints of the pseudorandom paths to determine the starting points of the “continued” quasi-random paths. One involves creating a transformation function and will be called the **Empirical Transformation Method**, while the other uses the endpoints directly and will be called **Correlated Continuation**.

First, we examine an obvious method which is not recommended. This will be

referred to as **Straight Continuation**. In Straight Continuation the endpoint of each pseudorandom path is taken in any arbitrary order to be the starting point for a new quasi-random path. For example, one might use $Z_k^{(m)}$ as the starting point for the k 'th quasi-random walk, which is weighted by $e^{-Y_k^{(m)}}$. Each path is then followed for a few additional steps using a single Halton point, with each set of $3M$ or $6M$ components corresponding to a time step. With a small number of steps, the dimensions can be moderate. This seems to be easiest and most sensible way to proceed, and indeed its simplicity and "naturalness" have probably led others to try it in some form or another. (I have not, however, found any evidence of such attempts, probably because of their lack of success.)

As hinted above, such a method does not work well. The problem is that this type of continuation essentially randomizes the starting points of each of the quasi-random paths. This has the effect of destroying the correlations between successive quasi-random points that enable quasi-random Monte Carlo estimates to converge quickly. As a result, without these correlations (since the starting points are arbitrarily randomized), the Monte Carlo estimates are just marginally better than pseudorandom.

The main problem is, again, that the starting location of each of the continued paths is uncorrelated with the subsequent steps, which use quasi-random components. The situation is analogous to taking a neatly spread out pattern of dots and then adding a random offset to each one. This has the effect, in most cases, of destroying the original pattern as well as most of its distinctive features. In the

case of quasi-random points, the “distinctive feature” is that they are more evenly spread out than purely random points, in general. This feature is lost for quasi-randomly generated paths, when a random (or pseudorandom) offset is arbitrarily added to each one, which is essentially what is happening in Straight Continuation.

Consider the following simple example:

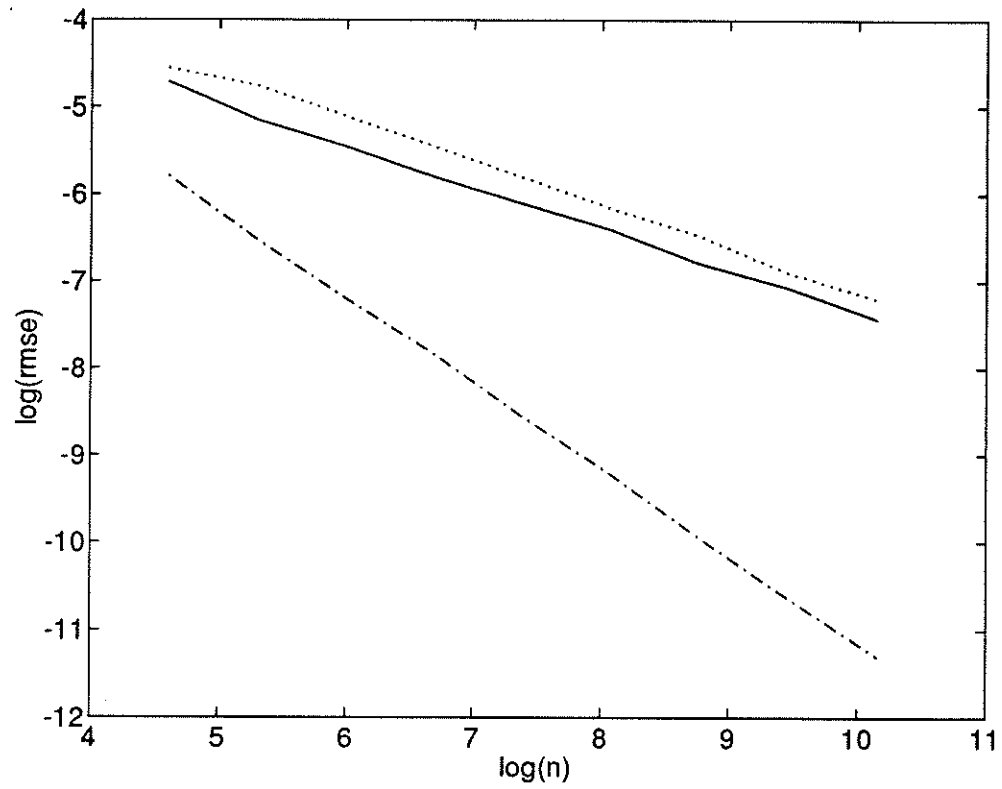
Example 10 *The following three-dimensional integral:*

$$I = \int_0^1 \int_0^1 \int_0^1 \sin(x + y + z) \, dx dy dz$$

is approximated in three different ways. The first is by a Crude Monte Carlo estimate using a three dimensional pseudorandom sequence of points. The second is a Crude Monte Carlo estimate using a three dimensional Halton sequence. Finally, the third way is a Crude Monte Carlo estimate using a combination of pseudorandom and Halton where each x_i is pseudorandom, while each y_i, z_i pair is from a two dimensional Halton sequence.

For each i , $i = 1, \dots, N$, this can be thought of as an extremely simple process which starts at x_i and then takes two jumps given by y_i and z_i , before finally having the sine function applied to its position to obtain a function value. Results are shown in Figure 7.1.

The results for Example 10 show that the estimates using combined pseudorandom and Halton sequences of points are just slightly better than the purely pseudorandom estimates and significantly worse than the purely Halton estimates.



Sequence Type	Slope	Plot Style
pseudorandom	-0.492	dotted line
Halton	-0.992	dash-dot line
pseudo/Halton	-0.480	solid line

Figure 7.1: Results for Example 10, 100 trials.

This is to be expected given the discussion above. The reason that the combined sequence estimate has an even lower convergence rate than pseudorandom in this example is not clear, but the important result is that this convergence rate is nowhere near $O(N^{-1})$. Thus, virtually all of the advantages to using quasi-random sequences are lost. For the Straight Continuation method, the observed results have been similar, though perhaps not as extreme, as will be seen in the examples in Section 7.8.

However, there are better ways to proceed, two of which will be discussed next. For both of the methods to follow, an extra set of components is required for each of the Halton points in order to determine a starting location for the path generated by the remaining components of that point. In one case (Empirical Transformation), the extra components are transformed into a starting location, using a transformation which is built up using the full set of pseudorandom paths. In the other case (Correlated Continuation), the extra components of each Halton point are used to pick out which particular pseudorandom path is to be continued using the remaining components of that same Halton point. Both of the above methods have the effect of reintroducing correlations which allow for faster convergence using quasi-random sequences.

7.7.1 Empirical Transformation Method

The Empirical Transformation Method involves using the weighted sample of endpoints of pseudorandom paths to construct a transformation function which will take points distributed according to $U([0, 1]^{3M})$ into points which are distributed like the pseudorandom endpoints in order to get very near to the ground state $\phi_T \psi_0$.

This transformation is applied to the first $3M$ components of each quasi-random point to obtain a starting point for a path, which is then continued for just a few time steps using the remaining components of that quasi-random point. The dimensions are moderate because only a few time steps are taken. Since the starting points are based on quasi-random components, none of the correlations are disturbed and faster quasi-random convergence is achieved than with Straight Continuation.

For simplicity, I will consider an unweighted sample of pseudorandom endpoints $\{Z_k\}_{k=1}^N$ generated using the branching method of simulation, in which paths are killed or split into multiple branches instead of assigning them weights. (See Section 7.5 for further details.) In the examples, this is the type of method which I will use for the pseudorandom stochastic paths.

The empirical transformation can be described as follows:

Partition the $3M$ dimensional domain of the Z_k 's by first selecting a set of $3M$

finite sequences of points in the domain as follows:

$$B_{1,j} = \nu_1^{(j)} < \nu_2^{(j)} < \dots < \nu_{m_j+1}^{(j)} = B_{2,j} \quad , \quad (j = 1, \dots, 3M)$$

where $\prod_{j=1}^{3M} [B_{1,j}, B_{2,j}]$ bounds the domain of \mathbf{Z} . If the domain is unbounded, choose $B_{1,j}, B_{2,j}$ so that the probability of any stochastic path wandering out to beyond $\prod_{j=1}^{3M} [B_{1,j}, B_{2,j}]$ is insignificant relative to the Monte Carlo errors and then restrict all the paths to stay within these bounds. Often the physics will provide a guide to selecting appropriate bounds.

Then, the partition consists of the following subregions:

$$P_{i_1, i_2, \dots, i_{3M}} = [\nu_{i_1}^{(1)}, \nu_{i_1+1}^{(1)}] \times [\nu_{i_2}^{(2)}, \nu_{i_2+1}^{(2)}] \times \dots \times [\nu_{i_{3M}}^{(3M)}, \nu_{i_{3M}+1}^{(3M)}]$$

over all sets of i_j with $1 \leq i_j \leq m_j$, ($j = 1, 2, \dots, 3M$).

The number of subintervals is equal to $\prod_{j=1}^{3M} m_j$, which could be large. This can cause memory size problems if it is too large, as will be discussed below.

Now, generate pseudorandom paths $\{\mathbf{Z}_k\}_{k=1}^N$, and then count the number of endpoints falling within each region of the partition.

Let $N_{i_1, \dots, i_{3M}}$ be the number of points in subregion $P_{i_1, \dots, i_{3M}}$.

Normalize the point counts, in order to take account of the fact that some subregions may be larger volume-wise than others, as follows:

$$\tilde{N}_{i_1, \dots, i_{3M}} = N_{i_1, \dots, i_{3M}} \cdot \frac{\text{vol}(P_{i_1, \dots, i_{3M}})}{\prod_{j=1}^{3M} (B_{2,j} - B_{1,j})}$$

Now, compute the following sets of marginal and conditional sums:

$$\tilde{N}_{1; i_1} = \sum_{i_2=1}^{m_2} \dots \sum_{i_{3M}=1}^{m_{3M}} \tilde{N}_{i_1, \dots, i_{3M}} \quad , \quad i_1 = 1 \text{ to } m_1$$

$$\begin{aligned}
\tilde{N}_{2;i_2|i_1} &= \sum_{i_3=1}^{m_3} \cdots \sum_{i_{3M}=1}^{m_{3M}} \tilde{N}_{i_1, \dots, i_{3M}} \quad , \quad i_1 = 1 \text{ to } m_1 \quad , \quad i_2 = 1 \text{ to } m_2 \\
\tilde{N}_{3;i_3|i_1, i_2} &= \sum_{i_4=1}^{m_4} \cdots \sum_{i_{3M}=1}^{m_{3M}} \tilde{N}_{i_1, \dots, i_{3M}} \quad , \quad i_1 = 1 \text{ to } m_1 \quad , \dots, \quad i_3 = 1 \text{ to } m_3 \\
&\vdots \\
\tilde{N}_{3M;i_{3M}|i_1, \dots, i_{3M-1}} &= \tilde{N}_{i_1, \dots, i_{3M}} \quad , \quad i_1 = 1 \text{ to } m_1 \quad , \dots, \quad i_{3M} = 1 \text{ to } m_{3M}
\end{aligned}$$

Next, add up the accumulations of each of the sums above, normalized to fall between zero and one as follows:

$$\begin{aligned}
C_{1;i_1} &= \frac{\sum_{i=1}^{i_1} \tilde{N}_{1;i}}{\sum_{i=1}^{m_1} \tilde{N}_{1,i}} \quad , \quad i_1 = 1 \text{ to } m_1 \\
C_{2;i_2|i_1} &= \frac{\sum_{i=1}^{i_2} \tilde{N}_{2;i|i_1}}{\sum_{i=1}^{m_2} \tilde{N}_{2;i|i_1}} \quad , \quad i_1 = 1 \text{ to } m_1 \quad , \quad i_2 = 1 \text{ to } m_2 \\
C_{3;i_3|i_1, i_2} &= \frac{\sum_{i=1}^{i_3} \tilde{N}_{3;i|i_1, i_2}}{\sum_{i=1}^{m_3} \tilde{N}_{3;i|i_1, i_2}} \quad , \quad i_1 = 1 \text{ to } m_1 \quad , \dots, \quad i_3 = 1 \text{ to } m_3 \\
&\vdots \\
C_{3M;i_{3M}|i_1, \dots, i_{3M-1}} &= \frac{\sum_{i=1}^{i_{3M}} \tilde{N}_{3M;i|i_1, \dots, i_{3M-1}}}{\sum_{i=1}^{m_{3M}} \tilde{N}_{3M;i|i_1, \dots, i_{3M-1}}} \quad , \quad i_1 = 1 \text{ to } m_1 \quad , \quad \dots
\end{aligned}$$

The transformation can then be explicitly defined.

Given $\mathbf{x} = (x_1, \dots, x_{3M}) \in U([0, 1]^{3M})$, we can produce a transformed $3M$ dimensional point, $\mathcal{T}(\mathbf{x}) = (\mathcal{T}_1(x_1), \dots, \mathcal{T}_{3M}(x_{3M}))$, with a probability distribution with is nearly like the distribution of the set of points $\{\mathbf{Z}_k\}_{k=1}^N$ as follows:

$$\mathcal{T}_1(x_1) = \nu_{i_1} + \frac{x_1 - C_{1;i_1}}{C_{1;i_1+1} - C_{1;i_1}} (\nu_{i_1+1} - \nu_{i_1})$$

where i_1 is selected such that $C_{1;i_1} \leq x_1 < C_{1;i_1+1}$.

$$\mathcal{T}_2(x_2) = \nu_{i_2} + \frac{x_2 - C_{2;i_2|i_1}}{C_{2;i_2+1|i_1} - C_{2;i_2|i_1}} (\nu_{i_2+1} - \nu_{i_2})$$

where i_2 is selected such that $C_{2;i_2|i_1} \leq x_1 < C_{2;i_2+1|i_1}$, given i_1 above.

⋮

This transformation has the nice property that it is (nearly) continuous, which is important because as observed in Chapter 5, any large discontinuities would have a negative impact on the quasi-random convergence rate. (There are slight discontinuities where jumps between adjacent partition regions cause different conditional densities to be used in the higher dimensions.)

The number of partitions should be small enough relative to N so that the majority of the partition regions are non-empty. However, on the other hand, if the number of partitions is too small, large approximation errors will result because information is lost when the set of pseudorandom endpoints is reduced to a set of counts in the partition regions. A careful and informed choice of the number and location of partitions, based on knowledge of the physics, can make a big difference in the success of this method. If a very large number of partitions are needed to avoid approximation errors, memory size problems can arise, which is a system-dependent limitation of this method. However, with memory sizes rapidly increasing, this should become less of a problem in the future. For one of the examples in Section 7.8, the harmonic oscillator in three dimensions, this method worked quite well without memory size problems. For the other example, the hydrogen atom, the memory size did become a concern when I ignored the obvious symmetries of the system. In general, if one takes advantage of any symmetries

(by combining symmetrical partitions) and also uses unequally sized partitions, this method could be made to work for a large number of interesting systems. However, if memory problems persist, the empirical transformation can instead be incorporated into Correlated Continuation which is discussed next.

7.7.2 Correlated Correlation

An alternative to the Empirical Transformation Method is **Correlated Continuation**. As mentioned earlier, straight continuation is ineffective because it destroys the correlations between successive quasi-random paths by randomizing the starting locations. However, a nice feature of such direct continuation is that there is no new approximation error introduced as there was for the Empirical Transformation Method above. This is trivially true because the *exact* pseudorandom path endpoints are reused.

The idea of correlated continuation is to use a direct continuation such as straight continuation so that no approximation error is involved, but with the important distinction that an “informed” choice is made in selecting which particular pseudorandom path endpoint to use as the start of each new quasi-random path. This is done by using the first $3M$ components of each quasi-random point to select the pseudorandom path to be continued using the remaining components of that same quasi-random point, as follows:

These first $3M$ components are transformed, using a rough empirical transfor-

mation (no great accuracy is needed so the number of partitions can be small), into the same general domain and distribution as the pseudorandom path endpoints. Then, an exact pseudorandom endpoint which is “nearby” to this transformed point is used as the starting position for the path generated using the remaining components of that particular quasi-random point. This allows a lot of the necessary correlations to be reintroduced, since the particular starting point is “near” to where the first $3M$ quasi-random components suggest that it should be.

Ideally one might wish to use a nearest neighbor approach, where the pseudorandom endpoint which is closest to the transformed point above is selected. However, unfortunately, such an approach is computationally too costly since it involves $O(N^2)$ work when done naively, or at best $O(N \log N)$ work using a method known as tessellation (see Bowyer [6]), while everything else involves only order $O(N)$ work.

Instead, I have found it to be sufficient to simply choose a starting point which is at least in the same general area as the transformed point above. This appears to produce enough correlation to be effective in the examples which I have studied. In practice one stores the pseudorandom endpoint locations according to the region of space in which they lie (using memory stacks). Then, for the transformed quasi-random components, any pseudorandom endpoint within the same region of space is used as the starting point for the path continued using the remaining components of that particular quasi-random point. If no pseudorandom endpoints are left in the region, then the nearby regions are searched until a pseudorandom endpoint is

found.

Importantly, each pseudorandom path endpoint is used once and only once as a new starting point. This ensures that the new set of quasi-random starting positions is distributed exactly like the old set of pseudorandom endpoints. Hence no new approximation error is introduced.

This method seems to hold the most promise because unlike the Empirical Transformation Method, there are no memory storage problems.

The examples below will provide some more details about these methods for combining pseudorandom and quasi-random sequences. As will be shown in the examples, for systems in which the number of bodies, M , is small, both the Empirical Transformation Method and Correlated Continuation provide the opportunity to combine pseudorandom and quasi-random sequences of points in order to obtain more accurate results than those computed using pseudorandom sequences alone.

7.8 Diffusion Monte Carlo Examples

In this section I will examine the computational results when the two methods described in the last section are applied to two particular quantum systems, and compare these results with those obtained using ordinary pseudorandom simulation and straight continuation.

The four methods to be considered, all of which are based on the trial function transformed system, Equation 7.19, and variational energy estimates, Equa-

tion 7.32, are the following:

1. Standard Pseudorandom

An ‘ensemble’ of $N_1 < N$ random walkers is initially distributed according to ϕ_T^2 . They are advanced by finite time steps, h , using branched walks as described in Section 7.5 and the Helfand/Greenside 3 third order accurate discretization method. This is continued up to a large time t_p . The proper sizes of h and t_p are determined empirically by observing the resultant root mean square errors in estimating E_0 .

Then, starting at time step $m = \frac{t_p}{h}$, a sample of “endpoints” is accumulated by taking the set of walk locations at every s ’th step after the m ’th, where s is determined empirically to be large enough so that the new set of endpoints is (nearly) independent of the previous set. This continues until a total of N “endpoints” has been accumulated. (Note: This is a very common time saving technique, which is equivalent to the more straightforward method of generating N independent paths, as long as s is sufficiently high. I have used it in order to make my comparisons in terms of cpu time more realistic.)

Given accumulated set of endpoints, $\{Z_k\}_{k=1}^N$, we then have the following Monte Carlo estimate of E_0 :

$$\hat{E}_N^{(sp)} = \frac{1}{N} \sum_{k=1}^N b(Z_k)$$

Note: There are no weight factors because branched walks are used.

2. Straight Continuation

The sample of pseudorandom endpoints $\{Z_k\}_{k=1}^N$ is accumulated as in case (1) above. Then, Z_k is taken as the starting point, $Z_k^{(0)}$, for the k 'th quasi-random (Halton) path. (Note: $Y_k^{(0)} = 0$). This path is followed for m additional steps, where m is small, using the components of the k 'th Halton point with Helfand/Greenside 3 to obtain the weighted sample, $\{(Z_k^{(m)}, Y_k^{(m)})\}_{k=1}^N$, where each $Z_k^{(m)}$ represents a spatial location and the associated weights are given by $e^{-Y_k^{(m)}}$.

Then, the following Monte Carlo estimate is computed:

$$\hat{E}_N^{(sc)} = \frac{\sum_{k=1}^N b(Z_k^{(m)}) e^{-Y_k^{(m)}}}{\sum_{k=1}^N e^{-Y_k^{(m)}}}$$

3. Empirical Transformation Method

The sample of pseudorandom endpoints $\{Z_k\}_{k=1}^N$ is again accumulated as in case (1) above. Then an empirical transformation function, \mathcal{T} , is constructed as described in Section 7.7.1 using P equally sized cubes as partition regions with bounds $B_{1,j}$, $B_{2,j}$ $\forall j$. Note: I have ignored any symmetries in the examples in order to get an idea of the performance of this method in a more general setting.

Next, use the empirical transformation function to obtain starting points, setting $Z_k^{(0)} = \mathcal{T}(h_k^{(1)}, h_k^{(2)}, h_k^{(3)})$ and $Y_k^{(0)} = 0$ where $h_k^{(1)}, h_k^{(2)}, h_k^{(3)}$ are the first three components of each point in a $(3 + 6m)$ -dimensional Halton sequence.

As in case (1) above, the path is then continued for m time steps to obtain the weighted sample, $\{(Z_k^{(m)}, Y_k^{(m)})\}_{k=1}^N$.

Finally, the following Monte Carlo estimate is computed:

$$\hat{E}_N^{(etf)} = \frac{\sum_{k=1}^N b(Z_k^{(m)}) e^{-Y_k^{(m)}}}{\sum_{k=1}^N e^{-Y_k^{(m)}}}$$

4. Correlated Continuation

The sample of pseudorandom endpoints $\{Z_k\}_{k=1}^N$ is again accumulated as in the previous cases. Then an empirical transformation function, \mathcal{T} , is constructed as in case (3) above, but with a much smaller number of partitions, P . In addition, S memory stacks are created, each corresponding to a particular region in $3M$ dimensional space. Each Z_k 's coordinates are then "pushed" on to the stack associated with the spatial region in which it lies. Next, set $Z_k^* = \mathcal{T}(h_k^{(1)}, h_k^{(2)}, h_k^{(3)})$ where $h_k^{(1)}, h_k^{(2)}, h_k^{(3)}$ are again the first three components of each $(3 + 6m)$ -dimensional Halton point. Determine which spatial region Z_k^* lies in, and then "pop" the coordinates of a point Z_j off of the stack associated with that particular region. If this stack is empty, search the other stacks from nearest to furthest (according to a fixed table of search orders), until a non-empty stack is found and a point is "popped" of that stack.

Now, set $Z_k^{(0)} = Z_j$, $Y_k^{(0)} = 0$ and proceed as in cases (2) and (3) above to obtain the sample $\{(Z_k^{(m)}, Y_k^{(m)})\}_{k=1}^N$.

Finally, compute the following Monte Carlo estimate of E_0 :

$$\hat{E}_N^{(cc)} = \frac{\sum_{k=1}^N b(Z_k^{(m)}) e^{-Y_k^{(m)}}}{\sum_{k=1}^N e^{-Y_k^{(m)}}}$$

Example 11 *The ground state energy, E_0 , of a three dimensional harmonic oscillator is estimated using Diffusion Monte Carlo. This quantity is known exactly, which allows for exact computation of root mean square errors for comparison purposes. The governing potential energy function is the following:*

$$v(x, y, z) = \frac{1}{2} k r^2$$

where $k > 0$ is the oscillator strength, and $r^2 = x^2 + y^2 + z^2$ is the squared distance from the oscillator's center, which is taken to be the origin.

Setting the strength to be $k = 1$ and the mass to $m = 1$ for the sake of simplicity, and working with atomic units (in which $\hbar = 1$), the exact ground state wave function given by the following:

$$\psi_0(x, y, z) = \eta_0 e^{-r^2/2}$$

where η_0 is simply a normalization constant.

The exact ground state energy is then,

$$E_0 = 1.5 \quad \text{atomic units}$$

The trial function used is the following:

$$\phi_T(x, y, z) = \eta_T e^{-.51r^2}$$

which is intended to simulate a typical situation in which the trial function is well chosen, but not exactly correct.

The resultant trial energy function is then the following:

$$b(\mathbf{y}) = \frac{H\phi_T(\mathbf{y})}{\phi_T(\mathbf{y})} = 1.53 - 0.202r^2$$

While the vector drift function is the following:

$$\mathbf{a}(\mathbf{y}) = (-1.02x, -1.02y, -1.02z)$$

The results of all four methods described above are shown in Table 7.1 and Figures 7.2-7.3. For the pseudorandom paths, the parameters are ensemble size, $N_1 = 50$; startup time, $t_p = 12$ sec; time step size, $h = 0.12$; and steps between saves, $s = 15$. For all the quasi-random methods the number of quasi-random steps is $m = 4$. For the Empirical Transformation there are $P = 100^3$ partitions with bounds -5 to +5 in each component. For Correlated Continuation there are only $P = 30^3$ partitions, $S = 8^3$ storage stacks, and bounds -5 to +5.

Example 12 The ground state energy, E_0 , of a hydrogen atom is estimated using Diffusion Monte Carlo. Using atomic units, as in the last example, the exact solution is known to be $E_0 = -0.5$. Therefore this can be used to compute the root mean square errors of the estimates.

The governing potential energy function is the following:

$$v(x, y, z) = -\frac{1}{r}$$

where $r = \sqrt{x^2 + y^2 + z^2}$ is the distance from the nucleus, which is taken to be the origin.

The exact ground state wave function given by the following:

$$\psi_0(x, y, z) = \eta_1 e^{-r}$$

where η_1 is a normalization constant.

The trial function used is the following:

$$\phi_T(x, y, z) = \eta_T e^{-r+\lambda r^2}$$

which is specifically selected so that it eliminates the singularity at $r = 0$, but is also not quite exactly correct.

The resultant trial energy function is then the following:

$$b(\mathbf{y}) = -\frac{1}{2}(1 - 2\lambda r)^2 + 3\lambda$$

While the vector drift function is the following:

$$\mathbf{a}(\mathbf{y}) = (2\lambda x - \frac{x}{r}, 2\lambda y - \frac{y}{r}, 2\lambda z - \frac{z}{r})$$

The results of all four methods described above are shown in Table 7.2 and Figures 7.4-7.5 with $\lambda = 0.01$. For the pseudorandom paths, the parameters are ensemble size, $N_1 = 50$; startup time, $t_p = 16.8$ sec; time step size, $h = 0.21$; and steps between saves, $s = 20$. For all the quasi-random methods the number of quasi-random steps is $m = 4$. For the Empirical Transformation there are $P = 100^3$ partitions and bounds -5 to +5. For Correlated Continuation there are only $P = 30^3$ partitions, $S = 8^3$ storage stacks, and bounds -10 to +10.

The results for these two examples indicate that the Empirical Transformation Method and Correlated Continuation can each produce more accurate results than the usual pseudorandom method, even when the extra time needed to implement these new methods is added in.

The Empirical Transformation Method has the advantage of using fully quasi-random starting points, but a great deal of memory storage is needed and there is a long initial start-up time to set up the transformation, which is evident in the time plots. From the plots, it is seen, though, that this start-up time is not a major factor as N gets large.

Correlated Continuation has the advantage of avoiding any new approximation errors from the transformation, and therefore allowing for a smaller number of partitions. I believe that for more complex systems with M equal to two or three bodies, this method holds the best promise, because it avoids new approximation errors.

Straight Continuation actually does slightly better than I might have expected, but it is clearly not as good as the two methods above. Also, it is more erratic in its results, sometimes coming close to the two methods above and other times giving results close to pseudorandom.

The convergence rates for the Empirical Transformation Method and Correlated Continuation Method are faster than the pseudorandom convergence rate, thus giving improved accuracy for a given amount of cpu time. The rates are not quite as high as in some of the earlier examples in Chapters 5 and 6, because in

the few quasi-random steps, the pseudorandom Monte Carlo errors cannot be totally eliminated. (Note: The rates against time for the Empirical Transformation Method are misleading because the large start-up time makes them look better than they actually are.)

In general then, although huge improvements such as seen in some of the earlier examples are not present, these combination methods do in fact work, and even when the extra time involved in these methods is factored in, the results are still better than the original pseudorandom estimates.

3D Harmonic Oscillator, Comparison of Results

	Method 1 (Pseudorandom Paths)		Method 2 (Straight Continuation)	
N	$rmse(N)$	t_N	$rmse(N)$	t_N
200	1.888 E -3	1.06	1.735 E -3	1.16
800	8.083 E -4	2.29	8.542 E -4	2.73
3200	4.228 E -4	7.12	4.408 E -4	9.01
12800	2.392 E -4	26.52	1.558 E -4	34.12
51200	1.058 E -4	105.08	8.506 E -5	134.36
	Convergence of $rmse$		Convergence of $rmse$	
	vs N :	-0.504	vs N :	-0.566
	vs t_N :	-0.619	vs t_N :	-0.675

	Method 3 (Empirical Transf.)		Method 4 (Correlated Contin.)	
N	$rmse(N)$	t_N	$rmse(N)$	t_N
200	1.442 E -3	2.20	1.554 E -3	1.21
800	6.834 E -4	3.91	7.061 E -4	2.85
3200	2.791 E -4	10.59	2.664 E -4	9.40
12800	1.234 E -4	37.07	1.254 E -4	35.74
51200	4.971 E -5	142.76	5.120 E -5	141.72
	Convergence of $rmse$		Convergence of $rmse$	
	vs N :	-0.649	vs N :	-0.631
	vs t_N :	-0.881	vs t_N :	-0.748

Table 7.1: Results for Example 11, 70 trials.

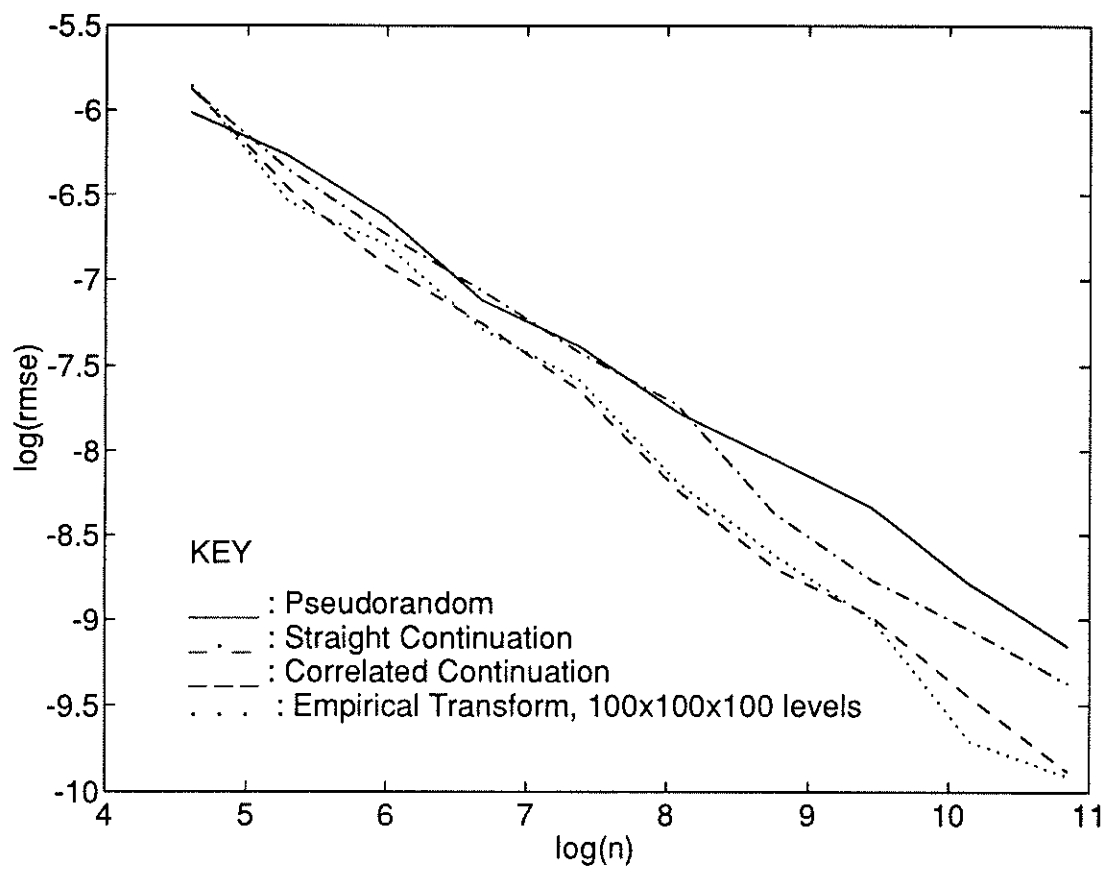


Figure 7.2: Log-Log Plot for Example 11.

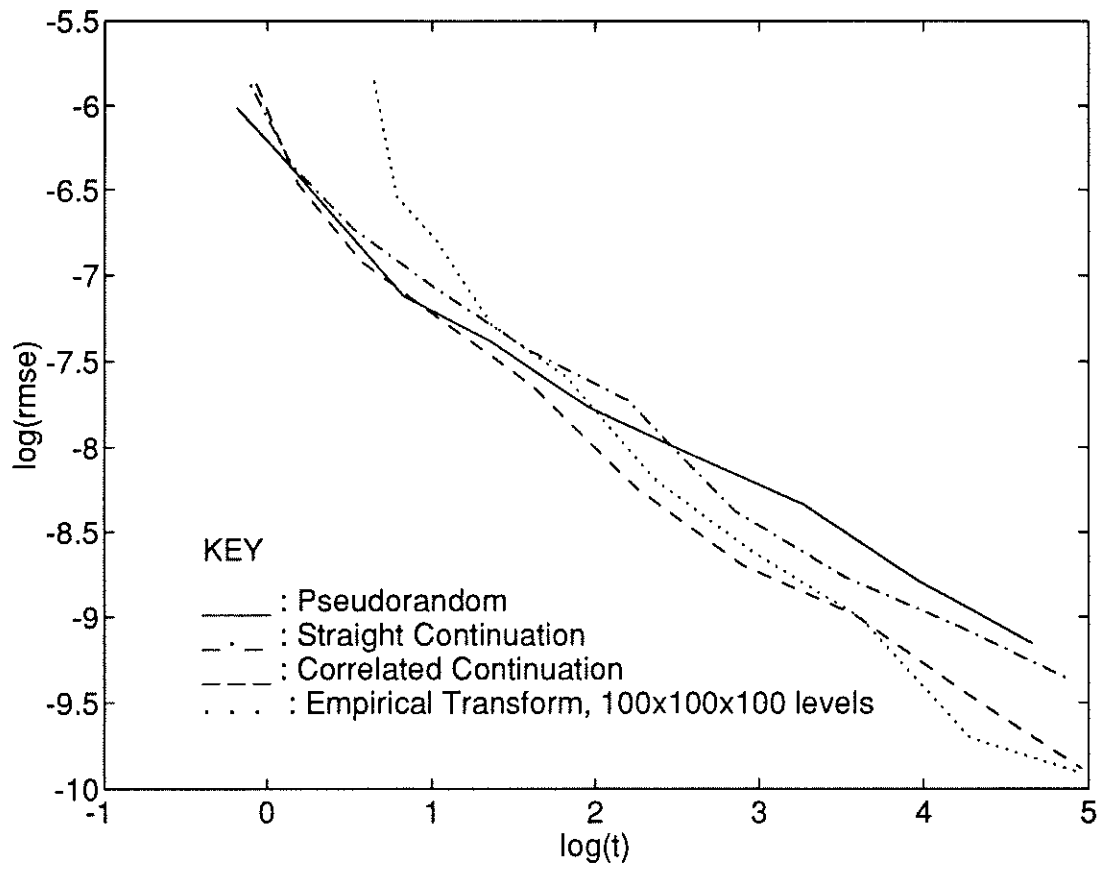


Figure 7.3: Log-Log Plot vs $t(\text{sec})$ for Example 11.

Hydrogen Atom, Comparison of Results

	Method 1 (Pseudorandom Paths)		Method 2 (Straight Continuation)	
N	$rmse(N)$	t_N	$rmse(N)$	t_N
200	1.396 E -3	1.33	1.238 E -3	1.47
800	6.972 E -4	3.21	5.604 E -4	3.76
3200	3.116 E -4	10.77	2.960 E -4	12.91
12800	1.534 E -4	40.94	1.329 E -4	49.65
51200	9.193 E -5	163.22	8.789 E -5	196.49
	Convergence of $rmse$		Convergence of $rmse$	
	vs N :	-0.512	vs N :	-0.519
	vs t_N :	-0.602	vs t_N :	-0.598

	Method 3 (Empirical Transf.)		Method 4 (Correlated Contin.)	
N	$rmse(N)$	t_N	$rmse(N)$	t_N
200	1.163 E -3	2.47	1.172 E -3	1.54
800	5.382 E -4	4.82	5.651 E -4	3.87
3200	2.509 E -4	14.23	2.386 E -4	13.35
12800	1.250 E -4	51.64	1.214 E -4	51.40
51200	6.178 E -5	201.14	6.143 E -5	204.59
	Convergence of $rmse$		Convergence of $rmse$	
	vs N :	-0.550	vs N :	-0.554
	vs t_N :	-0.704	vs t_N :	-0.636

Table 7.2: Results for Example 12, 60 trials.

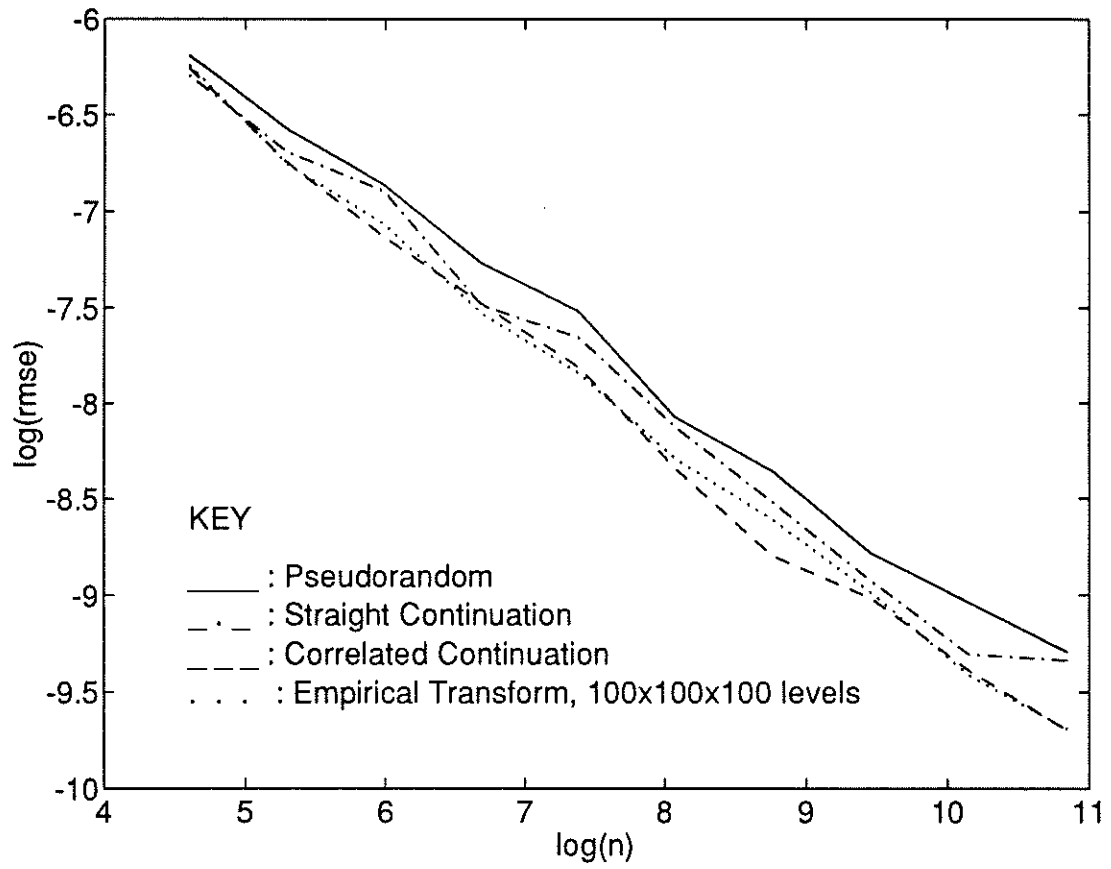


Figure 7.4: Log-Log Plot for Example 12.

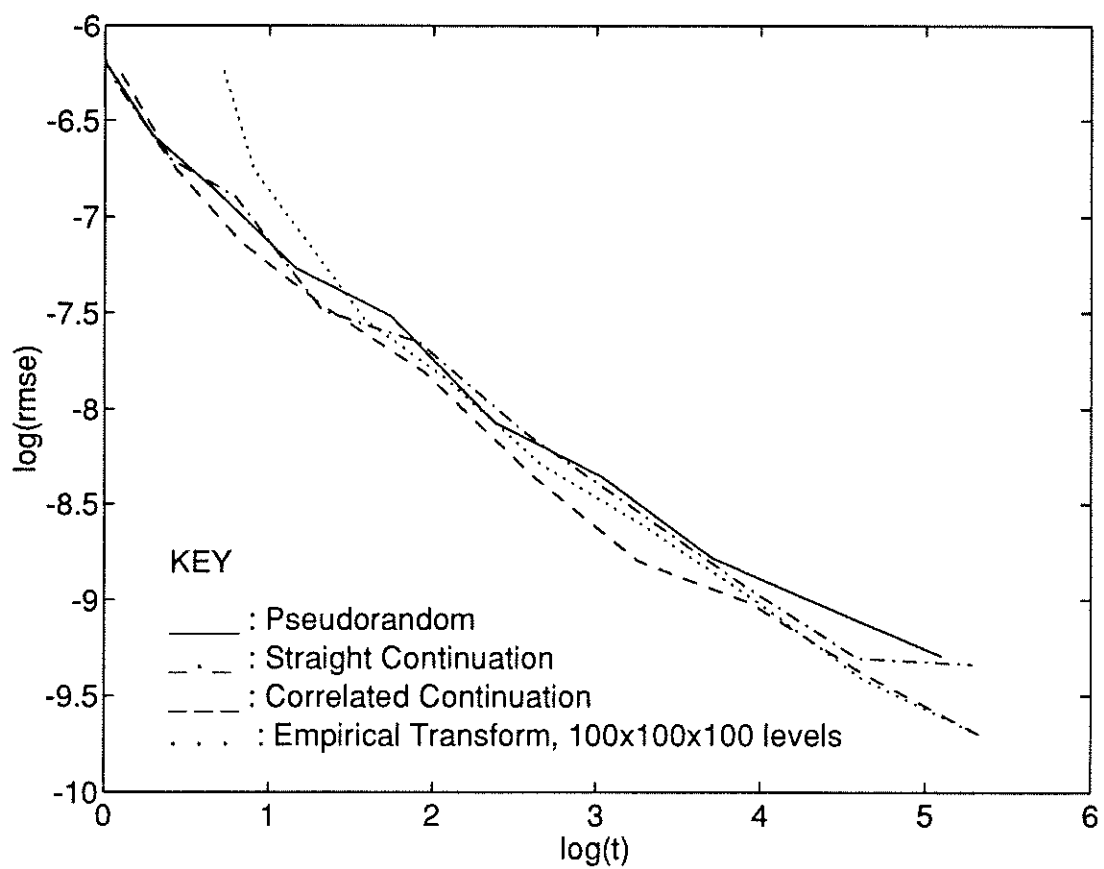


Figure 7.5: Log-Log Plot vs $t(\text{sec})$ for Example 12.

CHAPTER 8

Conclusion

Quasi-random sequences are certainly much more versatile than many people may believe. In fact, they can be successfully applied to a wide variety of Monte Carlo methods in order to obtain faster convergence rates and higher accuracy in a given amount of computation time on the computer.

However, one must be careful in selecting the specific algorithms used and the ways in which quasi-random sequences are applied. In particular, it is important to avoid any unnecessary discontinuities which are produced by the algorithms. It is also important to try to concentrate as much of the variance or variation into the lowest dimensions, or to reduce the number of dimensions whenever possible.

In the case of variance reduction using an importance function, we found that the Rejection Method introduces large discontinuities which have a strong negative impact on the success of quasi-random sequences. When one avoids these discontinuities by using a smoothed version of the Rejection Method, or Weighted Uniform Sampling, the results are significantly improved, as was observed computationally. In particular, I believe that Weighted Uniform Sampling holds great promise as a variance reduction method with quasi-random sequences to compute Monte Carlo estimates of integrals of moderate dimensions (roughly 4 to 30).

With another example, the Feynman-Kac formula, it is clearly demonstrated that reformulating an algorithm to concentrate the variance in the lower dimensions can have a huge impact on quasi-random Monte Carlo estimates, while having virtually no effect on pseudorandom Monte Carlo estimates. This is a striking example of the new issues which arise when quasi-random sequences are applied, and of the improvements possible when one pays attention to these issues.

Quasi-random sequences can also be used for Monte Carlo methods involving iterations or successive time steps, as I have done in the case of Diffusion Monte Carlo to obtain improved estimates of the ground state energy of few-body quantum mechanical systems. In these cases, because of the large number of dimensions involved, quasi-random sequences are used in combination with pseudorandom sequences. Continuation methods such as the *Empirical Transformation Method* and *Correlated Continuation*, allow one to use pseudorandom sequences to converge to some equilibrium state or ground state and then use quasi-random sequences to reduce the remaining statistical or Monte Carlo errors.

In conclusion, quasi-random sequences can be successfully applied to many problems in physics, chemistry, reliability analysis, and other fields in which Monte Carlo methods are commonly utilized. This dissertation should serve as a guide to some ways in which this can be done.

APPENDIX A

Moments of Sums of Random Variables

Assume $|f| \leq M_1$ and $0 < h \leq M_2$ over domain $I^d = [0, 1]^d \subset \mathbb{R}^d$. Also assume that $|Ef| \geq \epsilon > 0$. (Note: These conditions are sufficient, but not necessary for the results that follow.)

Let,

$$\mu'_i(f) = E[(f - Ef)^i] \text{ , } i\text{'th central moment of } f.$$

$$\mu'_i(h) = E[(h - Eh)^i] \text{ , } i\text{'th central moment of } h.$$

$$\mu'_{i,j}(f, h) = E[(f - Ef)^i (h - Eh)^j] \text{ , } i, j\text{'th mixed central moment.}$$

Then,

$$|Ef| \leq M_1 \quad \Rightarrow \quad |f - Ef| \leq 2 M_1$$

$$0 < |Eh| \leq M_2 \quad \Rightarrow \quad |h - Eh| \leq M_2$$

Therefore,

$$|\mu'_i(f)| \leq |2 M_1|^i < \infty$$

$$|\mu'_i(h)| \leq |M_2|^i < \infty$$

$$|\mu'_{i,j}(f, h)| \leq |2 M_1|^i |M_2|^j < \infty$$

Also,

$$\mu'_1(f) = E(f - Ef) = Ef - Ef = 0$$

$$\mu'_1(h) = E(h - Eh) = Eh - Eh = 0$$

Now, define the following quantities:

$$\begin{aligned} F &= \sum_{i=1}^N f(\mathbf{x}_i) & H &= \sum_{i=1}^N h(\mathbf{x}_i) \\ \Delta F &= \frac{F - EF}{EF} & \Delta H &= \frac{H - EH}{EH} \end{aligned}$$

where the \mathbf{x}_i 's are an i.i.d. random sample from $U([0, 1]^d)$.

Using Equation 2.11, we find the following immediate results:

$$EF = N Ef \qquad EH = N Eh$$

$$E(\Delta F) = 0 \qquad E(\Delta H) = 0$$

Now, consider the following:

$$\begin{aligned} E[(\Delta H)^j] &= E\left(\left[\frac{\sum_{i=1}^N h(\mathbf{x}_i) - EH}{EH}\right]^j\right) \\ &= E\left(\left[\frac{\sum_{i=1}^N (h(\mathbf{x}_i) - Eh)}{N Eh}\right]^j\right) \\ &= \frac{1}{N^j (Eh)^j} E\left(\left[\sum_{i=1}^N (h(\mathbf{x}_i) - Eh)\right]^j\right) \\ &= \frac{1}{N^j (Eh)^j} E\left(\left[\sum_{i=1}^N a_i\right]^j\right), \quad a_i = h(\mathbf{x}_i) - Eh \\ &= \frac{1}{N^j (Eh)^j} E\left(\sum_{i_1=1}^N \sum_{i_2=1}^N \cdots \sum_{i_j=1}^N a_{i_1} a_{i_2} \cdots a_{i_j}\right) \\ &= \frac{1}{N^j (Eh)^j} \sum_{i_1=1}^N \sum_{i_2=1}^N \cdots \sum_{i_j=1}^N E(a_{i_1} a_{i_2} \cdots a_{i_j}) \end{aligned}$$

For each term in this sum, let m be the number of distinct a_i 's contained in that term, so that $1 \leq m \leq j$. Given m for each term, we then assume that the first m factors of the term are in fact unique, while the remaining factors are by necessity repetitions of these. This involves a lumping together of 2^{m-1} equivalent terms, by commutation, since there may be up to $m - 1$ switches of position to create the situation described above.

This allows the sum to be rewritten as follows:

$$\begin{aligned}
E[(\Delta H)^j] &= \frac{1}{N^j (Eh)^j} \sum_{m=1}^j 2^{m-1} \left(\sum_{i_1=1}^N \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^N \cdots \sum_{\substack{i_m=1 \\ i^m \notin \{i_1, \dots, i_{m-1}\}}}^N \right. \\
&\quad \left. \sum_{\substack{i_{m+1}=1 \\ i_{m+1} \in \{i_1, \dots, i_m\}}}^N \cdots \sum_{\substack{i_j=1 \\ i_j \in \{i_1, \dots, i_m\}}}^N E(a_{i_1} \cdots a_{i_j}) \right) \\
&= \frac{1}{N^j (Eh)^j} \sum_{m=1}^j 2^{m-1} \left(\sum_{i_1=1}^N \sum_{\substack{j_1=1 \\ j_1 \neq i_1}}^{j-m} \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^N \sum_{j_2=1}^{j-m+1-j_1} \cdots \right. \\
&\quad \left. \sum_{\substack{i_m=1 \\ i^m \notin \{i_1, \dots, i_{m-1}\}}}^N \sum_{j_m=1}^{j-\sum_{r=1}^{m-1} j_r} E(a_{i_1}^{j_1} \cdots a_{i_m}^{j_m}) \right) \\
&= \frac{1}{N^j (Eh)^j} \sum_{m=1}^j 2^{m-1} \left(\sum_{i_1=1}^N \sum_{\substack{j_1=1 \\ j_1 \neq i_1}}^{j-m} \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^N \sum_{j_2=1}^{j-m+1-j_1} \cdots \right. \\
&\quad \left. \sum_{\substack{i_m=1 \\ i^m \notin \{i_1, \dots, i_{m-1}\}}}^N \sum_{j_m=1}^{j-\sum_{r=1}^{m-1} j_r} E(a_{i_1}^{j_1}) \cdots E(a_{i_m}^{j_m}) \right)
\end{aligned}$$

where the last step uses the independence of the different a_i 's.

Now, all of the terms in the sum above in which any j_i equals 1 will disappear since $E(a) = E(h - Eh) = Eh - Eh = 0$. Therefore, we have for all of the remaining terms: $j_i \geq 2, \forall i$. This implies that $m \leq \frac{j}{2}$ for all the remaining terms

as follows:

$$j = \sum_{i=1}^m j_i \geq 2m \Rightarrow m \leq \frac{j}{2}$$

Therefore, the sum can be reduced to the following:

$$\begin{aligned} E[(\Delta H)^j] &= \frac{1}{N^j (Eh)^j} \sum_{m=1}^{\frac{j}{2}} 2^{m-1} \left(\sum_{i_1=1}^N \sum_{j_1=1}^{j-m} \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^N \sum_{j_2=1}^{j-m+1-j_1} \dots \right. \\ &\quad \left. \sum_{\substack{i_m=1 \\ i_m \notin \{i_1, \dots, i_{m-1}\}}}^N \sum_{j_m=1}^{j-\sum_{r=1}^{m-1} j_r} E(a_{i_1}^{j_1}) \dots E(a_{i_m}^{j_m}) \right) \\ &\leq \frac{1}{N^j (Eh)^j} \frac{j}{2} 2^{\frac{j}{2}-1} N^{\frac{j}{2}} j^{\frac{j}{2}} M_2^j \\ &= c(j, M_2) N^{-j/2} \end{aligned}$$

Thus, we have proven the following result:

$$E[(\Delta H)^j] = O(N^{-j/2})$$

This result can in fact be improved for odd values of j , but the present result is sufficient for our purposes.

An argument which is virtually identical to the one above leads to similar results for $E[\Delta F (\Delta H)^j]$ as follows:

$$E[\Delta F (\Delta H)^j] = d(j, M_1, M_2) N^{-(j+1)/2}$$

where $d(j, M_1, M_2) = \frac{1}{(Eh)^j} \frac{j}{2} 2^{\frac{j}{2}-1} j^{\frac{j}{2}} M_1 M_2^j$.

Therefore we have the following:

$$E[\Delta F (\Delta H)^j] = O(N^{-(j+1)/2})$$

Bibliography

- [1] Veená Adlakha. An empirical evaluation of anithetic variates and quasirandom points for simulating stochastic networks. *Simulation*, 58(1):23–31, January 1992.
- [2] James B. Anderson. A random walk simulation of the Schrödinger Equation: H_3^+ . *Journal of Chemical Physics*, 63(4):1499–1503, August 1975.
- [3] James B. Anderson. Simplified sampling in Quantum Monte Carlo: Application to H_3^+ . *Journal of Chemical Physics*, 86(5):2839–2843, March 1987.
- [4] James B. Anderson and Carol A. Traynor. Quantum chemistry by random walk: Exact treatment of many-electron systems. Research Report, May 1991.
- [5] Michael Berblinger and Christoph Schlier. Monte Carlo integration with quasi-random numbers: Some experience. *Computer Physics Communications*, 66:157–166, 1991.
- [6] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.

- [7] Paul Bratley and Bennett L. Fox. Implementing Sobol's quasirandom sequence generator. *ACM Transactions of Mathematical Software*, 14(1):88–100, March 1988.
- [8] David Ceperly and Berni Alder. Quantum Monte Carlo. *Science*, 231:555–560, February 1986.
- [9] D.M. Ceperly and B.J. Alder. Quantum Monte Carlo for molecules: Green's function and nodal release. *Journal of Chemical Physics*, 81(12):5833–5844, 1984.
- [10] Siu A. Chin. Quadratic diffusion Monte Carlo algorithms for solving atomic many-body problems. *Physical Review A*, 42(12):6991–7005, December 1990.
- [11] Luc Devroye and László Györfi. *Nonparametric Density Estimation*. Wiley, New York, 1985.
- [12] H. Engels. *Numerical Quadrature and Cubature*. Academic Press, New York, 1980.
- [13] Bennett L. Fox. Implementation and relative efficiency of quasirandom sequence generators. *ACM Transactions of Mathematical Software*, 12(4):362–376, December 1986.
- [14] Avner Friedman. *Stochastic Differential Equations and Applications, Volume 1*. Academic Press, Inc., New York, 1975.

- [15] Thomas C. Gard. *Introduction to Stochastic Differential Equations*. Marcel Dekker Inc., New York, 1988.
- [16] C.W. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences*. Springer-Verlag, Berlin; New York, 1985.
- [17] Daniel T. Gillespie. *Markov Processes: An Introduction for Physical Scientists*. Academic Press, Inc., San Diego, California, 1992.
- [18] H.S. Greenside and E. Helfand. Numerical integration of stochastic differential equations - II. *The Bell System Technical Journal*, pages 1927–1941, October 1981.
- [19] R. Grimm and R.G. Storer. A new method for the numerical solution of the Schrödinger equation. *Journal of Computational Physics*, 4:230–249, 1969.
- [20] R. Grimm and R.G. Storer. Monte Carlo solution of Schrödinger's equation. *Journal of Computational Physics*, 7:134–156, 1971.
- [21] J.H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.
- [22] J.M. Hammersley and D.C. Handscomb. *Monte Carlo Methods*. John Wiley and Sons, New York, 1965.

- [23] Morris H. Hansen, William N. Hurwitz, and William G. Madow. *Sample Survey Methods and Theory, Volumes I and II*. John Wiley and Sons, Inc., New York, 1953.
- [24] Robert J. Harrison, 1991. private communication by email.
- [25] W.H. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [26] E. Helfand. Numerical integration of stochastic differential equations. *The Bell System Technical Journal*, pages 2289–2299, December 1979.
- [27] F. James. A review of pseudorandom number generators. *Computer Physics Communications*, 60(3):329–344, 1990.
- [28] Chris Joslin and Saul Goldman. Quantum Monte Carlo studies of two-electron atoms constrained in spherical boxes. *Journal of Physics B - Atomic, Molecular, and Optical Physics*, 25(9):1965–1975, 1992.
- [29] Mark Kac. *Probability and Related Topics in Physical Sciences*. Interscience Publishers, New York, 1959.
- [30] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods, Volume I: Basics*. J. Wiley and Sons, New York, 1986.
- [31] M.H. Kalos. Monte Carlo calculations of the ground state of three- and four-body nuclei. *Physical Review*, 128(4):1791–1795, November 1962.

- [32] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, Berlin; New York, 1992.
- [33] D.E. Knuth. *The Art of Computer Programming, Volume II : Seminumerical Algorithms*, pages 34–97. Addison Wesley, Reading, Mass., 1969.
- [34] Steven E. Koonin. *Computational Physics*. Benjamin/Cummings, Menlo Park, California, 1986.
- [35] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences*. Wiley, New York, 1974.
- [36] Christian Lecot. A quasi-Monte Carlo method for the Boltzmann equation. *Mathematics of Computation*, 56(194):621–644, April 1991.
- [37] Michael E. Lee and Kevin E. Schmidt. Green’s Function Monte Carlo. *Computers in Physics*, pages 192–197, March/April 1992.
- [38] Yu L. Levitan, N.I. Markovich, S.G. Rozin, and I.M. Sobol’. Short communications on quasirandom sequences for numerical computations. *Zh. vychisl. Mat. mat. Fiz.*, 28(5):755–759, 1988.
- [39] Hua Loo-Keng and Wang Yuan. *Applications of Number Theory to Numerical Analysis*. Springer-Verlag, Berlin; New York, 1981.
- [40] George Marsaglia. Normal (Gaussian) random variables for supercomputers. *The Journal of Supercomputing*, 5:49–55, 1991.

- [41] William J. Morokoff. *Quasi-Monte Carlo Methods for Numerical Integration and Simulation*. PhD thesis, New York University, May 1990.
- [42] Harald Niederreiter. Quasi-Monte Carlo methods and pseudo-random numbers. *Bulletin of the AMS*, 84(6), November 1978.
- [43] Harald Niederreiter. Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30:51–70, 1988.
- [44] Harald Niederreiter. Quasi-Monte Carlo methods for multidimensional integration. *International Series of Numerical Mathematics*, 85:157–171, 1988.
- [45] D.M. O’Brien. Accelerated quasi Monte Carlo integration of the radiative transfer equation. *J. Quant. Spectrosc. Radiat. Transfer*, 48(1):41–59, 1992.
- [46] M.J.D. Powell and J. Swann. Weighted uniform sampling – a Monte Carlo technique for reducing variance. *J.Inst.Maths Applics*, 2:228–236, 1966.
- [47] William H. Press and Saul A. Teukolsky. Quasi- (that is, sub-) random numbers. *Computers in Physics*, pages 76–79, Nov/Dec 1988.
- [48] Peter J. Reynolds, Jan Tobochnik, and Harvey Gould. Diffusion quantum Monte Carlo. *Computers in Physics*, pages 662–668, Nov/Dec 1990.
- [49] P.K. Sarkar and M.A. Prasad. A comparative study of pseudo and quasi random sequences for the solution of integral equations. *Journal of Computational Physics*, 68:66–88, 1987.

- [50] Leonard I. Schiff. *Quantum Mechanics*. McGraw-Hill, New York, 1949.
- [51] K.E. Schmidt and J.W. Moskowitz. Monte Carlo calculations of atoms and molecules. *Journal of Statistical Physics*, 43:1027–1041, 1986.
- [52] Zeev Schuss. *Theory and Applications of Stochastic Differential Equations*. Wiley, New York, 1980.
- [53] I.U.A. Shreider. *The Monte Carlo Method: the Method of Statistical Trials*. Pergamon Press, New York, 1966.
- [54] B.V. Shuhman. Application of quasirandom points for simulation of gamma radiation transfer. *Progress in Nuclear Energy*, 24:89–95, 1990.
- [55] I.M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zh. vychisl. Mat. mat. Fiz.*, 7(4):784–802, 1967.
- [56] I.M. Sobol'. Pseudo-random numbers for constructing discrete Markov chains by the Monte Carlo method. *Zh. vychisl. Mat. mat. Fiz.*, 14(1):36–44, 1974.
- [57] I.M. Sobol'. Quasi-Monte Carlo methods. *Progress in Nuclear Energy*, 24:55–61, 1990.
- [58] J.F. Traub and H. Woźniakowski. The Monte Carlo algorithm with a pseudo-random generator. *Mathematics of Computation*, 58(197):323–339, January 1992.

- [59] Carol A. Traynor, James B. Anderson, and Bruce M. Boghosian. A quantum Monte Carlo calculation of the ground state energy of the hydrogen molecule. *Journal of Chemical Physics*, 94(5):3657–3664, March 1991.
- [60] S.S. Wilks. *Mathematical Statistics*. Wiley, New York, 1962.
- [61] H. Woźniakowski. Average case complexity of multivariate integration. *Bulletin (New Series) of the AMS*, 24(1):185–194, January 1991.
- [62] S.K. Zaremba. The mathematical basis of Monte Carlo and quasi-Monte Carlo methods. *SIAM Review*, 10(3), July 1968.
- [63] S.W. Zhang and M.H. Kalos. Bilinear quantum Monte-Carlo - expectations and energy differences. *Journal of Statistical Physics*, 70(3-4):515–533, February 1993.