UNIVERSITY OF CALGARY

A Study in Hybrid Monte Carlo Methods in Computing Derivative Prices

by

Binbin Wang

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF MATHEMATICS AND STATISTICS

CALGARY, ALBERTA

December, 2012

Abstract

Hybrid Monte Carlo (HMC) method is defined in this thesis as Monte Carlo method that utilizes conditional expectation so that the regular Monte Carlo method and other computational methods can be combined to price financial derivatives. This thesis introduces several hybrid Monte Carlo methods and studies the algorithm and efficiency of these methods, which include three methods combining Monte Carlo with fast Fourier transform, cosine series, and Black-Scholes formula respectively.

These methods can be considered as ways of variance reduction. The thesis also introduces a new variance reduction method using orthogonal transformation which further reduces the variance. It is shown in this thesis that the HMC methods can significantly improve the efficiency when compared to the regular Monte Carlo method. A basket option example is used throughout this thesis for implementation and efficiency comparison.

Acknowledgements

I would like to first express my deepest gratitude to my supervisor, Dr. Antony Ware, for his excellent guidance and patience in our numerous times of discussion. I wouldn't be able to start and finish this thesis without his support.

I wish to thank my dear wife Jia for her support and encouragement during my study.

I also wish to thank Dr. Michael Lamoureux and Dr. Philip Chang for carefully reviewing my thesis. My sincere gratitude also goes to the Department of Math & Stats, and the Faculty of Graduate Studies for the financial support during my graduate program.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations and Nomenclature

Symbol	Definition
HMC	Hybrid Monte Carlo
GBM	Geometric Brownian Motion
p.d.f.	probability density function
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
c.d.f.	cumulative distribution function
COS HMC	COS-based Hybrid Monte Carlo
FFT HMC	FFT-based Hybrid Monte Carlo
BS HMC	Black-Scholes Hybrid Monte Carlo
var	Variance
cov	Covariance

Chapter 1

Introduction

When computing price of derivatives on multiple assets using Monte Carlo method, one needs to generate samples $Z_i,\ i = 1, \ldots, M$ of some random vector Z. Then the price is computed by taking the mean of all $f(Z_i)$ for some payoff function f. Instead of computing $E[f(Z)]$ purely using Monte Carlo method, we can consider use Monte Carlo partially on some dimensions of Z and use some other methods on the rest. We call such methods Hybrid Monte Carlo (HMC) methods.

This idea can be realized as follows. Let $Z = (z_1, z_2, \ldots, z_n)$, $X = (z_1, \ldots, z_k)$, and $Y = (z_{k+1}, \ldots, z_n)$ for some $1 \leq k < n$ so that $Z = (X, Y)$. To compute $E[f(Z)]$, we first generate samples of Y, and then we compute the conditional expectation $E[f(Z)|Y]$ which should be a function of Y, denoted by $g(Y)$. For each sample Y_i of Y, $g(Y_i)$ is computed using some method other than Monte Carlo, and then a Monte Carlo method is applied on $g(Y)$ to get $E[g(Y)]$. By the property of conditional expectation we have

$$E[g(Y)] = E\left[E[f(Z)|Y]\right] = E[f(Z)],$$

so $E[f(Z)]$ is computed.

In this thesis we introduce some hybrid Monte Carlo methods which integrate computing methods such as Fourier series, COS series, and Black-Scholes-like for-

mulas. These methods are applied to a basket option example described in Section 1.2.

1.1 A Literature Review

The first HMC method we introduce in this thesis is based on the Fourier Transform. While there is little literature on hybrid Monte Carlo method using Fourier transforms, there is a rich and comprehensive literature on computing financial derivative prices using Fourier transforms and inverse Fourier transforms. In the first subsection we give a brief literature review on Fourier Methods in finance. In the second subsection we introduce the mixed approach in Loeper and Pironneau (2009) [16], which can be considered as a hybrid Monte Carlo method.

1.1.1 Fourier Transform methods in Finance

Fourier transform methods have been one of the major methodologies for computing prices of financial derivatives in recent years. One of the reasons is that more sophisticated financial models are introduced beyond the classic Black-Scholes model. These new models usually involve stochastic processes that are not normally distributed, which results in difficulties in computing expectations. The Fourier transform directly relates the characteristic function of a random variable to its density function, hence can be naturally implemented in the computation.

To illustrate how the Fourier transform can be used, we briefly introduce the idea in the original paper of Carr and Madan (1999) [6].

Given an option with strike K, maturity T, interest rate r, and spot price S_t, let $k = \ln K$, $s_t = \ln S_t$. Then the time zero price of the option can be considered as a function of k and is given by

$$C(k) = \int_k^\infty e^{-rT}(e^x - e^k)q(x)dx,$$

where $q(x)$ is the risk-neutral density function of s_T. One problem of computing the integral is that $q(x)$ usually does not have a closed form. Instead we usually have a closed form for the characteristic function $\phi(u)$. Then choose some $\alpha > 0$ such that $c(k) = e^{\alpha k}C(K)$ is square integrable and apply Fourier transform to $c(k)$ to get

$$\psi(v) = \int_{-\infty}^{\infty} e^{ivk} c(k) dk.$$

Applying the inverse Fourier transform we can get

$$C(k) = \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{\infty} e^{-ivk} \psi(v) dv. \tag{1.1}$$

Using the fact that the characteristic function $\phi(u)$ of s_T is the Fourier transform of $q(x)$, one can show that $\psi(v)$ is linked to $\phi(u)$ by the equation

$$\psi(v) = \frac{e^{-rT}\phi(v - (\alpha + 1)i)}{\alpha^2 + \alpha - v^2 + i(2\alpha + 1)v}. \tag{1.2}$$

Combining (1.1) and (1.2) we can compute the option price. Note that (1.1) is a Fourier transform so the FFT can be used (see for detail). To obtain $\psi(v)$ in equation (1.1) we only need the characteristic function $\phi(u)$.

The Carr and Madan approach is based on the Fourier Transform with respect to the log strike. In the Ph.D Thesis [19] (2000), Raible derived a formula of prices for European options by bilateral Laplace transform with respect to the log spot price. The FFT can be used to evaluate the formula. Raible's basic idea is as follows.

For a European option on a stock with price $S_t = S_0 e^{X_t}$ and payoff function $g(S_T)$ at expiration T, the price of the option at time 0 is $V(c) = e^{-rT} E_Q[g(S_T)] = e^{-rT} E_Q[g(e^{-c+X_T})]$, where $c = -\ln S_0$ and Q is the risk-neutral probability. Let $\rho(x)$ be the density function of X_T, and let $\pi(x) = g(e^{-x})$, then

$$V(c) = e^{-rT} E_Q[\pi(c - X_T)] = e^{-rT} \int_{\mathbb{R}} \pi(c - x)\rho(x) dx,$$

which is e^{-rT} times the convolution of π and ρ.

Let $\mathcal{L}_f(z)$ denoted the bilateral Laplace transform of $f(x)$ defined by $\mathcal{L}_f(z) = \int_{\mathbb{R}} e^{-zx} f(x)dx$. Then, using the property of Laplace transform, we have $\mathcal{L}_V = e^{-rT}\mathcal{L}_\pi\mathcal{L}_\rho$. By taking the inverse bilateral Laplace transform we can get

$$V(c) = \frac{e^{-rT+cR}}{2\pi} \int_{\mathbb{R}} e^{icu}\mathcal{L}_\pi(R+iu)\mathcal{L}_\rho(R+iu)du,$$

where R is a real number so that both $\mathcal{L}_\pi(R+iu)$ and $\phi_{X(T)}(iR-u)$ are defined for $u \in \mathbb{R}$. But note that $\mathcal{L}_\rho(R+iu) = \phi_{X(T)}(iR-u)$ where $\phi_{X(T)}$ is the characteristic function of X_T. Therefore we have

$$V(c) = \frac{e^{-rT+cR}}{2\pi} \int_{\mathbb{R}} e^{icu}\mathcal{L}_\pi(R+iu)\phi_{X(T)}(iR-u)du,$$

which can be considered as a Fourier Transform, so that the FFT can be applied to its calculation.

Besides Carr & Madan and Raible's methods, various methods using FT and FFT to compute derivative prices have been developed in recent years in different areas of financial derivatives. These methods consist of applications to different type of derivatives (e.g. Parisian options in Anderluh and Weide (2009) [1], electricity derivatives in Deng [8], and Asian options in Benhamou (2002) [4] and Fusai (2004) [12]), generalizations of old methods (implied diffusion approach in Andersen and Andreasen (2000) [2], use of the FFT in Dempster and Hong (2000) [7]), and new methods such as the COS method by Fang and Oosterlee (2008) [11] (which is related to our COS HMC method), and the CONV method in Lord et al. (2008) [17].

1.1.2 Loeper and Pironneau's mixed approach

While there is little literature related to our topic of hybrid Monte Carlo, Loeper and Pironneau (2009) [16] have a paper about a mixed PDE/Monte Carlo method which

can be considered as a hybrid Monte Carlo method. Here we briefly introduce their idea.

Assume that under the risk neutral probability measure the underlying stock price S_t follows a Heston stochastic volatility model in Heston (1993) [15], i.e.

$$\frac{dS_t}{S_t} = rdt + \sigma_t dW_t^1$$
$$dv_t = k(\theta - v_t)dt + \delta\sqrt{v_t}dW_t^2$$

where $v_t = \sigma_t^2$, and (W_t^1, W_t^2) is a 2-dimensional instantaneously correlated Brownian motion with correlation coefficient ρ between the increments per unit time.

A standard Monte Carlo simulation of this price process will require a two-dimensional discretized process. One is the volatility and the other one is the stock price. In Loeper and Pironneau (2009) [16] the authors realize that a mixed approach of Monte Carlo and a PDE solver can be applied. More specifically, the volatility process is simulated by its discretization:

$$v_{i+1} = v_i + k(\theta - v_i)\Delta t + \delta\sqrt{v_t \Delta t}Z_2$$

where Z_2 is a standard normal random variable.

Now, conditional on a path of v_t, i.e. a sequence $\{v_i | i = 1, 2, \ldots, N\}$, let the volatility stay constant over each time interval. Then the stock price process can be considered as a generalized geometric Brownian motion with deterministic volatility determined by $\{v_i | i = 1, 2, \ldots, N\}$. Then S_T has a analytic solution and the option price can be computed using Monte Carlo method if the option is not path-dependent. If the option is path-dependent, numerical methods are used.

This is quite different from our approach but it also makes use of the idea of conditional distribution. The purpose of HMC methods is to improve computation efficiency, in terms of computation time and variance reduction. As we will see the later chapters, the efficiency improvement of HMC can be very significant.

One of the nature of HMC is that it is case sensitive and requires the user to analyze the situation, choose the appropriate method, and make necessary modifications to the method. For example, the COS-based HMC introduced in Chapter 4 and the Black-Scholes HMC introduced in Chapter 5 are all based on specific properties of the payoff function and may not be applicable if the payoff function changes.

1.2 The Basket Option

The basket option example is from Milevsky and Posner (1998) [18]. It is a call option on a product linked to the performance of the G-7 stock markets. The interest rate r is 6.3%, and the option expires in 3 months ($T = 0.25$) with strike 1 ($K = 1$).

The payoff of the option is

$$e^{-rT}\left(\sum_{i=1}^{7} w_i S_i(T) - K\right)_+,$$

where w_i are the weights. We assume that the stock prices $(S_1, S_2, \ldots, S_7)$ follow Geometric Brownian Motions (GBM), i.e. the stock price S_i satisfies the stochastic differential equations

$$dS_i = (r - d_i)dt + \sigma_i dW_i(t)$$

under the risk neutral probability measure, where $(W_1(t), W_2(t), \ldots, W_7(t))$ is a 7-dimensional Brownian motion with correlated increments, and the correlation matrix per unit time is a constant matrix C. The weights (w_i), the annualized dividend yields (d_i) and volatilities (σ_i) are given in the Table 1.1. The correlation matrix C is given in the Table 1.2.

Since the stock prices follow GBM process, we have

$$S_i(T) = S_i(0)\exp\{(r - d_i - \frac{\sigma_i^2}{2})T + \sigma_i W_i(T)\}.$$

The correlated Brownian motion $W = (W_1(T), W_2(T), \ldots, W_7(T))^t$ at time T can be simulated by $\sqrt{T}LZ$ where L is the Cholesky decomposition (i.e. a lower triangular

Country	Index	Weight(%)	Annualized Volatilities(%)	Dividend Yield(%)
Canada	TSE 100	10	11.55	1.69
France	CAC 40	15	20.68	2.39
Germany	DAX	15	14.53	1.36
Italy	MIB30	5	17.99	1.92
Japan	Nikkei 225	20	15.59	0.81
U.K.	FTSE 100	10	14.62	3.62
U.S.	S&P 500	25	15.68	1.66

Table 1.1: G-7 Stock data

Country	Canada	France	Germany	Italy	Japan	U.K.	U.S.
Canada	1	0.35	0.1	0.27	0.04	0.17	0.71
France	0.35	1	0.39	0.27	0.5	-0.08	0.15
Germany	0.1	0.39	1	0.53	0.7	-0.23	0.09
Italy	0.27	0.27	0.53	1	0.46	-0.22	0.32
Japan	0.04	0.5	0.7	0.46	1	-0.29	0.13
U.K.	0.17	-0.08	-0.23	-0.22	-0.29	1	-0.03
U.S.	0.71	0.15	0.09	0.32	0.13	-0.03	1

Table 1.2: G-7 Stock correlations

matrix such that $C = LL^t$) of C and Z is a 7-dimensional standard normal random variable. This is true because $\sqrt{T}LZ$ is normal and

$$cov(\sqrt{T}LZ) = TLL^t = TC.$$

We can then rewrite the discounted payoff as a function of Z

$$f(Z) = e^{-rT}\left(\sum_{i=1}^{7} \alpha_i e^{(AZ)_i} - K\right)_+,$$

where $(AZ)_i$ is the i-th row of AZ, α_i is given by

$$\alpha_i = S_i(0)w_i e^{-d_i T + (r - \sigma_i^2/2)T},$$

and A is a seven by seven matrix formed by taking account of volatilities, scales, and the Cholesky decomposition of the correlation matrix, i.e.

$$A = \sqrt{T}\sigma L$$

where σ is the diagonal matrix with σ_i as its diagonal entries. With the data in the table we can get

$$A = \begin{bmatrix} 0.2001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1254 & 0.3355 & 0 & 0 & 0 & 0 & 0 \\ 0.0252 & 0.0954 & 0.2315 & 0 & 0 & 0 & 0 \\ 0.0841 & 0.0584 & 0.1463 & 0.2553 & 0 & 0 & 0 \\ 0.0108 & 0.1401 & 0.1466 & 0.0320 & 0.1751 & 0 & 0 \\ 0.0430 & -0.0377 & -0.0525 & -0.0435 & -0.0339 & 0.2346 & 0 \\ 0.1928 & -0.0286 & 0.0174 & 0.0391 & 0.0437 & -0.0313 & 0.1762 \end{bmatrix}$$

Let B be sub-matrix of A obtained by removing the first column of A, and let $a = [a_1, \ldots, a_7]^t$ be the first column of A. As we will see in Chapter 4, we can write

$$\sum_{i=1}^{7} \alpha_i e^{(AZ)_i} = \sum_{i=1}^{7} \beta_i e^{a_i z_1}$$

where $\beta_i = \alpha_i e^{(B\bar{z})_i}$, and $(B\bar{z})_i$ is the i-th entry of $B\bar{z}$.

1.3 Structure of thesis

This thesis is organized as follows. Chapter 1 introduces the basic idea and the basket option example we use throughout this thesis, and also includes a literature review on related subjects. Chapter 2 includes preliminary material used in this thesis including Monte Carlo method, Fourier Transform, and COS series etc.

In Chapter 3 we introduce the first HMC method using Fast Fourier Transform (FFT), and a detailed discussion on error analysis, variance reduction, and efficiency is included.

Chapter 4 presents the second HMC method using COS series, which is similar to but also different than the FFT-based HMC method. In Chapter 5 the third HMC method using Black-Scholes-like formulas is presented.

All implementations of the methods described in this thesis are done in Matlab. While some of the key codes are presented in the thesis, a complete package of codes may be obtained by contacting the author.

Chapter 2

Preliminary

In this Chapter we review the basics of Monte Carlo methods, Fourier series, COS series, and conditional expectations, all of which will be used in later chapters.

2.1 Introduction to Monte Carlo Method

In a narrow sense, Monte Carlo method is a class of algorithms of computing expectations of random variables using random sampling. While it has a fully developed system of methodologies with numerous application in scientific computing, we only introduce the material we will be using in this thesis.

The theoretic foundation of Monte Carlo method relies on the *Law of Large Numbers* and the *Central Limit Theorem*. But before we get to the detail, let's look at its intuitive background. When trying to evaluate the expectation of a random variable X, one can certainly try to find its *probability density function* (p.d.f.) $f(x)$ (if exists) and integrate to get the expected value

$$E[X] = \int_{\mathbb{R}^n} xf(x)dx.$$

This will not always work due to various reasons. For example, we don't know the p.d.f., or even if we know it, it may be difficult to integrate.

The intuitive idea of taking average to get expected value, which is the most basic idea of Monte Carlo method, is as follows. For the random variable X, if there is a way of generate a large number of independent samples X_i of X, $i = 1, 2, \ldots, N$, then the average of these samples, $\frac{1}{N}\sum_{i=1}^{N} X_i$ "must" be approximately equal to the expected value. It turns out to be right under certain conditions and extremely useful. It is guaranteed by Kolmogorov's strong law of large numbers (see for example Grimmett and Stirzaker (1992) [14]).

Theorem 2.1.1 *Let $\{X_i\}_{i=1}^{\infty}$ be an sequence of identical independently distributed random variables on a probability space $(\Omega, \mathcal{F}, P)$ with expected value $E[X_i] = \mu$, and $E|X_i| < \infty$ for all i, and let $\bar{X}_n = \frac{1}{N}\sum_{i=1}^{n} X_i$, then $\bar{X}_n$ converges to μ almost surely, i.e., there exit $\Omega_0 \in \mathcal{F}$ such that $P(\Omega_0) = 0$ and for any $\omega \in \Omega \setminus \Omega_0$, $\bar{X}_n \to \mu$.*

We should also be concerned about the accuracy of the estimation obtained this way. One would guess that the larger number of samples used, the more accurate estimation we will get. It also turn out be quite right. But we want to know how fast the convergence is. The central limit theorem says that the convergence speed is proportional to the square root of the number of samples N.

Theorem 2.1.2 *Let $\{X_i\}_{i=1}^{\infty}$ be an sequence satisfying the conditions in the theorem above, and X_i has finite variance $Var(X_i) = \sigma^2$, then*

$$\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \to N(0, 1)$$

as $n \to \infty$ in distribution, where $N(0, 1)$ is a standard normal random variable. In other words, the distribution function of $\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}$ (weakly) converges to the standard normal distribution function.

This may not tell us directly how it is related to the error of the Monte Carlo estimation. Observe that $\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \to N(0, 1)$ in distribution implies that $\frac{\bar{X}_n - \mu}{\sigma} -$

$N(0,1)/\sqrt{n} \to 0$ in distribution. Note that we used the fact if $A_n \to A$ in distribution and a sequence of numbers $a_n \to a$, then $a_n A_n \to aA$ in distribution, which can be found in most books of probability theory.

Therefore we can see that the distribution of $\bar{X}_n - \mu$ is approximately the same as the distribution function of $N(0, \sigma^2/n)$ for large n, i.e. $\bar{X}_n - \mu$ is nearly a normal random variable with 0 mean and standard deviation $\sigma/\sqrt{n}$. Based on this fact, we can get an estimation of the error of the Monte Carlo estimation. This also tells us that to get one more significant digit, we need to increase the number of sample by 100 times.

For example, we know the radius of the 95% confidence interval of standard random variable is 1.96, then we know for sufficiently large n, we are (asymptotically) 95% sure that the error $\bar{X}_n - \mu$ of the Monte Carlo estimation is within the confidence interval $(-\frac{1.96\sigma}{\sqrt{n}}, \frac{1.96\sigma}{\sqrt{n}})$, since for large n

$$P(|\bar{X}_n - \mu| < \frac{1.96\sigma}{\sqrt{n}}) \approx P(|Z| < 1.96) = 0.95,$$

where Z is a standard normal random variable.

There is one more problem. In order to get the error estimation, we need the standard deviation, which is usually not known. Just like the mean of X, the standard deviation can also estimated by taking the sample standard deviation

$$\bar{\sigma}_n = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)}.$$

Replacing the σ by $\bar{\sigma}$ we get an estimate of the 95% confidence interval of the form $(-\frac{1.96\bar{\sigma}}{\sqrt{n}}, \frac{1.96\bar{\sigma}}{\sqrt{n}})$

Monte Carlo method has an advantage when the random variable is multi-dimensional, because the standard error term like $\frac{1.96\bar{\sigma}}{\sqrt{n}}$ does not depends on the dimension. To the contrary, some other method such as the trapezoidal rule in d dimensions has an error of $O(n^{-2/d})$ for twice continuously differentiable integrands. Nevertheless,

as d increases, the Monte Carlo method will require sampling of multi-dimensional distributions, and so the computation load will significantly increase.

One other component of the Monte Carlo method is to generate random numbers. While this is a huge topic itself, we are not concerned about it as we are mainly going to encounter normal random variables and all implementation are done in Matlab, which has efficient random number generators for our purpose.

2.2 Variance Reduction

As we have seen in the previous section that the standard error like $\frac{1.96\bar{\sigma}}{\sqrt{n}}$ depends on the number of samples n and also the sample variance $\bar{\sigma}$. Besides increasing n to get better estimate, we can also work on reducing the sample variance $\bar{\sigma}$. Techniques of reducing the sample variance are call *variance reduction* techniques.

Commonly used variance reduction methods include *antithetic variables*, *control variates*, *stratified sampling*, and *importance sampling*, etc.

We give a brief introduction of antithetic variables, control variate and importance sampling here.

2.2.1 Antithetic Variables

When computing the expected value of a random variable $V = f(X)$ using Monte Carlo method where $X \sim U[0,1]^n$, we observe that $1 - X$ has the same distribution as X. So one can consider the new random variable

$$V^a = \frac{1}{2}(f(X) + f(1 - X)),$$

which has the same mean as V but may have smaller variance than V.

Since generating samples of V^a will need approximately twice the amount of the operations of generating the same number of samples of V, we would expect the

variance of V^a to be less than half of that of V. The variance of V^a is

$$\begin{aligned}\mathrm{var}(V^a) &= \frac{1}{4}\mathrm{var}(f(X)) + \frac{1}{4}\mathrm{var}(f(1-X)) + \frac{1}{2}\mathrm{cov}(f(X), f(1-X)) \\ &= \frac{1}{2}\mathrm{var}(V) + \frac{1}{2}\mathrm{cov}(f(X), f(1-X))\end{aligned}$$

Therefore if $\frac{1}{2}\mathrm{cov}(f(X), f(1-X)) < 0$ then the efficiency is improved. For example it is true when f is monotone in each dimension. This can be shown by Chebyshev's algebraic inequality (for example see Prop. 2.1 in Egozcue et al (2009) [9]). Another way to see this is by recognizing the monotonicity of $f(1-x)$ and then use Property 1 in Esary et al (1967) [10].

Antithetic variables also work when X is a n-dimensional standard random variable, in which case we can use

$$V^a = \frac{1}{2}(f(X) + f(-X)).$$

In the normal random variable case, monotonicity also guarantee that

$$\frac{1}{2}\mathrm{cov}(f(X), f(1-X)) < 0.$$

2.2.2 Control Variates

To compute the expected value of a random variable V by Monte Carlo, we can consider adding another random variable that has mean zero to V but can reduce the variance. More specifically we try to find a constant λ and another random variable V^c such that $\mu^c = E[V^c]$ can be easily computed (by analytical formula or other lower cost methods), so that the variance of the random variable defined by

$$W = V - \lambda(V^c - \mu^c)$$

is significantly reduced.

By "significantly reduced" we mean for example if the cost of generating samples of V^c is similar to that of V, then we expect that the variance of W is at most

$\frac{1}{2}\text{var}(V)$. The variance of W is

$$\text{var}(W) = \text{var}(V) + \lambda^2 \text{var}(V^c) - 2\lambda \text{cov}(V, V^c)$$

which is a quadratic function of λ, hence it's minimized when $\lambda = \dfrac{\text{cov}(V, V^c)}{\text{var}(V^c)}$. The min value of $\text{var}(W)$ is

$$\begin{aligned}
\text{var}(W) &= \text{var}(V) + \frac{\text{cov}(V, V^c)^2}{\text{var}(V^c)} \text{var}(V^c) - 2\frac{\text{cov}(V, V^c)}{\text{var}(V^c)} \text{cov}(V, V^c) \\
&= \text{var}(V) - \frac{\text{cov}(V, V^c)^2}{\text{var}(V^c)} \\
&= \text{var}(V)(1 - \text{corr}(V, V^c)^2).
\end{aligned}$$

Thus if $|\text{corr}(V, V^c)| > \frac{1}{\sqrt{2}}$, this technique is worthwhile. The larger the absolute value of the correlation, the more efficient this technique will be.

For example, if $V = (S_T - K)_+$, we can define $V^c = S_T - K$, which is usually strongly correlated to V and its mean can be calculated by analytical formulas.

In practice we usually don't have the exact value of $\text{cov}(V, V^c)$ so we usually use the sample covariance to compute λ.

2.2.3 Importance Sampling

In implementation of Monte Carlo of some random variable $f(X)$, we sometimes come across the problem that a very large sample size is needed to ensure convergence of the results. This maybe caused by the fact that there is not enough "important" samples which gives meaningful value of $f(X)$. For example if $E[f(X)]$ relies greatly on the tails of X but there is not enough samples lying in the tails, the computed mean maybe biased due to lack of "important samples". So instead of sampling from X, we can sample from another random variable Y which has a different density and emphasize more on "important samples". While this is the motive of the so-called importance sampling, the theory is as follows.

Let X, Y be random vectors in $\mathbb{R}^n$, and let $p_X(x)$ and $p_Y(x)$ be the density function of X and Y. If

$$p_Y(x) = 0 \Rightarrow p_X(x) = 0,$$

i.e. the support S_X of p_X is a subset of the support S_Y of p_Y, then we have the following:

$$\begin{aligned} E[f(X)] &= \int_{S_X} f(x)p_X(x)dx \\ &= \int_{S_X} f(x)\frac{p_X(x)}{p_Y(x)}p_Y(x)dx, \text{ because } p_Y(x) \neq 0 \text{ if } x \in S_X \subseteq S_Y \\ &= \int_{S_Y} f(y)\frac{p_X(y)}{p_Y(y)}p_Y(y)dy, \text{ because } p_X(y) = 0 \text{ if } y \in S_Y \setminus S_X \\ &= E\left[f(Y)\frac{p_X(Y)}{p_Y(Y)}\right]. \end{aligned}$$

Therefore we can then apply Monte Carlo to the random variable $f(Y)\frac{p_X(Y)}{p_Y(Y)}$ to get $E[f(X)]$. But in order for this to be advantageous, the variance of $f(Y)\frac{p_X(Y)}{p_Y(Y)}$ need to be smaller than the variance of $f(X)$. Since they have the same mean it suffices to compare the second moment. It is easy to check that

$$E\left[f^2(Y)\frac{p_X^2(Y)}{p_Y^2(Y)}\right] = E\left[f^2(X)\frac{p_X(X)}{p_Y(X)}\right],$$

which is not necessarily less than $E[f^2(X)]$ unless p_Y is appropriately chosen. The strategy is to choose $p_Y(x)$ so that it is roughly proportional to $f(x)p_X(x)$.

2.3 Fourier Series and Fast Fourier Transforms

In this section we review Fourier series and the so-called Fast Fourier Transform.

2.3.1 Fourier Series and its convergence

Let f be a function defined on the interval $[-R, R]$, then the *Fourier series* of f, if exists, is given by

$$\sum_{j=-\infty}^{\infty} \hat{f}_j e^{\frac{i\pi jx}{R}}, \text{ with } \hat{f}_j = \frac{1}{2R}\int_{-R}^{R} f(x)e^{-i\pi jx/R}dx \tag{2.1}$$

It is well know that Fourier series of f converges to f almost everywhere under an integrability condition.

Theorem 2.3.1 *If a function f is square-integrable on $[-R, R]$, then the Fourier series converges to $f(x)$ almost everywhere on the interval $[-R, R]$.*

We need to know more about the convergence, which is used in the error analysis in Section 3.2.

For well-behaved function f, the Fourier Series of f converges very fast as we see in the following theorem.

Theorem 2.3.2 (Truncation Error of Fourier Series) *Let $f(x)$ be a function on $[-R, R]$ and let $S_N f(x)$ be the partial sum $\sum_{j=-N}^{N} \hat{f}_j e^{i\omega_j x}$ ($\omega_j = j\pi/R$). If $f \in C^p([-R, R])$, then the truncation error $e_N(x) = f(x) - S_N f(x)$ satisfies*

$$\|e_N\| \leq \frac{R^p \|D^p f\|}{\pi^p N^p},$$

where $\|\cdot\|$ is a norm defined by

$$\|g\|^2 = \frac{1}{2R}\int_{-R}^{R} |g(x)|^2 dx$$

and $D^p f$ is the p-th derivative of f.

Proof We know that $\{e^{i\omega_j x} | j \in \mathbb{Z}\}$ is an orthogonal basis of $L^2[-R, R]$. By Parseval's

identity we have

$$\begin{aligned}
\|e_N\|^2 &= \sum_{|j|>N} |\hat{f}_j|^2 \\
&= \sum_{|j|>N} \frac{1}{|\omega_j|^{2p}} |\omega_j^p \hat{f}_j|^2 \\
&\leq \frac{1}{|\omega_N|^{2p}} \sum_{|j|>N} |\omega_j^p \hat{f}_j|^2 \\
&\leq \frac{1}{|\omega_N|^{2p}} \sum_{j=\infty}^{\infty} |\omega_j^p \hat{f}_j|^2 \\
&= \frac{\|D^p f\|^2}{|\omega_N|^{2p}}.
\end{aligned}$$

By taking the square root we get the desired inequality in the theorem. ∎

2.3.2 Approximate Fourier Coefficients using the Fast Fourier Transform

The Discrete Fourier Transform (**DFT**) is a map from $\mathbb{C}^N$ to $\mathbb{C}^N$ that maps $(x_0, x_1, \cdots, x_{N-1})$ to $(X_0, X_1, \cdots, X_{N-1})$ with

$$X_j = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi jn}{N}}.$$

Computing the DFT directly using the definition involves N^2 multiplications of complex numbers. The so-called *Fast Fourier Transform* (**FFT**) is an algorithm that computes DFTs more efficiently than computing directly using the definition.

A commonly used FFT algorithm is the Cooley-Tukey algorithm. The basic idea of the Cooley-Tukey algorithm is to use the periodic property of the function e^{ix} and then "divide and conquer". For example, the radix-2 DIT algorithm is as follows. Let N be a power of 2, and let E_k be k-th component of the DFT of the even-indexed vector $(x_0, x_2, \ldots, x_{N-2})$ and let O_k be the k-th component of the DFT of the odd-indexed vector $(x_1, x_3, \ldots, x_{N-1})$, then the following relationship is true:

$$X_k = \begin{cases} E_k + e^{-\frac{2\pi i}{N}k} O_k, & \text{if } k < N/2 \\ \\ E_{k-N/2} - e^{-\frac{2\pi i}{N}(k-N/2)} O_{k-N/2}, & \text{if } k \geq N/2 \end{cases}$$

This means that a size N DFT can be obtained by computing two size $N/2$ DFTs. By using this relationship recursively, it all reduces to DFTs of size 2. It is proved that the computation complexity of this algorithm is $O(N\ln(N))$.

There is a famous C subroutine library called FFTW (stands for Fastest Fourier Transform in the West) for computing the discrete Fourier transform (DFT) in one or more dimensions. The built-in functions in Matlab such as "fft", "fft2", and "fftn" are based on this library. Implementations in this thesis will be using these Matlab functions.

Fourier series and the DFT are closely related. The Fourier coefficients $\hat{f}_j$ in (2.1) can be approximated by a DFT through the following algorithm.

First the coefficients $\hat{f}_j = \frac{1}{2R}\int_{-R}^{R} f(x)e^{-i\pi jx/R}dx$ can be approximated by a finite sum

$$\tilde{f}_j = \frac{1}{2N}\sum_{n=-N}^{N} g_n e^{-i\omega_j x_n},$$

where $\omega_n = \frac{\pi j}{R}$ and g_n can be chosen according to the method of approximation. The $\tilde{f}_j$ is actually called the *discrete Fourier coefficient*. In this thesis we use the trapezoidal rule so $g_n = f(x_n)$ for $-N+1 \le n \le N-1$ and $g_n = f(x_n)/2$ for $n = N, -N$. Now choose $x_n = \frac{Rn}{N}$ so that $\omega_j x_n = \pi jn/N$, and define $h_{-N} = (f(x_{-N}) + f(x_N))/2$ and $h_n = g_n$ for $n = -N+1, \cdots, N-1$, then the sum becomes

$$\frac{1}{2N}\sum_{n=-N}^{N-1} h_n e^{-\frac{i2\pi jn}{2N}}.$$

To make it look like a DFT, we can arrange the order and define a new vector $u = [u_0, u_1, \cdots, u_{2N-1}]$ such that $u_n = h_n$ for $n = 0, 1, \cdots, N-1$ and $u_n = h_{n-2N}$ for $n = N, \cdots, 2N-1$, then one can show that the sum becomes

$$\frac{1}{2N}\sum_{n=0}^{2N-1} u_n e^{-\frac{i2\pi jn}{2N}},$$

which is exactly $\frac{1}{2N}$ times the j-th component of the DFT of u for $j = 0, 1, \cdots, 2N-1$. Then observe that $\hat{f}_j = \hat{f}_{j+2N}$ for $-N \le j \le -1$, so by rearrange the DFT of u and

then add the first entry to the end we can get an approximation of $[\hat{f}_{-N}, \cdots, \hat{f}_N]$.

A similar algorithm can be constructed for 2-dimensional Fourier series. The 2-dimensional Fourier series of a function $f(x, y)$ defined on $[-R, R]^2$ has the form

$$f(x, y) = \sum_{j,k} \hat{f}_{jk} \cdot e^{i\pi jx/R} e^{i\pi ky/R},$$

with

$$\hat{f}_{jk} = \frac{1}{4R^2} \int_{-R}^{R} \int_{-R}^{R} f(x, y) e^{-i\pi jx/R} e^{-i\pi ky/R} dxdy.$$

By choosing a proper 2-D lattice and rearranging the indices, the coefficients $\hat{f}_{jk}$ can be computed using 2-D DFT. For example, Let $x_n = \frac{Rn}{N_1}$, $y_m = \frac{Rm}{N_2}$, for $n = -N_1, \ldots, N_1 - 1$, $m = -N_2, \ldots, N_2 - 1$. Then define $f_{nm} = f(x_n, y_m)$. Then the 2-D Fourier coefficients $\hat{f}_{jk}$ can be approximated by

$$\tilde{f}_{jk} = \frac{1}{4N_1N_2} \sum_{n=-N_1}^{N_1-1} \sum_{m=-N_2}^{N_2-1} f_{jk} e^{-i\pi jx_n/R} e^{-i\pi ky_m/R}.$$

To make it look like the 2-D DFT, we can define

$$f_{jk} = \begin{cases} f_{j-2N_1,k}, \text{ if } j > N_1, k \leq N_2 \\ f_{j,k-2N_2}, \text{ if } j \leq N_1, k > N_2 \\ f_{j-2N_1,k-2N_2}, \text{ if } j > N_1, k > N_2 \end{cases}$$

Then using the periodic property we can get

$$\tilde{f}_{jk} = \frac{1}{4N_1N_2} \sum_{n=0}^{2N_1-1} \sum_{m=0}^{2N_2-1} f_{jk} e^{-i2\pi jn/2N_1} e^{-i2\pi km/2N_2}.$$

Hence the 2-D FFT can be used to compute the 2-D DFT of f_{jk}, and then $\tilde{f}_{jk}$ can be obtained by rearranging the indices.

2.4 The Fourier Cosine Series

For a function $f(x)$ defined on an interval $[a, b]$ that is square-integrable, the Fourier cosine series of $f(x)$ is

$$\frac{1}{2}c_0 + \sum_{k=1}^{\infty} c_k \cos\left(k\pi \frac{x-a}{b-a}\right) \tag{2.2}$$

with

$$c_k = \frac{2}{b-a}\int_a^b f(x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx,\ k = 0, 1, \ldots \tag{2.3}$$

As a Fourier cosine series is essentially a Fourier series, the convergence is the same as we described in Theorem 2.3.1 and 2.3.2, i.e. the Fourier cosine series converges to the function $f(x)$ almost everywhere and the truncation error decays very fast under some conditions of f. The cosine series is used in Chapter 4.

2.5 Conditional Expectation

All the hybrid Monte Carlo methods in this thesis are based on conditional expectation. In this section we review the definition and properties of conditional expectation.

Definition Let X be an integrable random variable on a probability space $(\Omega, \mathcal{F}, P)$. If $\mathcal{G}$ is a sub-σ-algebra of $\mathcal{F}$, then by the Radon-Nikodym theorem (for example see on Rudin (1986) [20, pp. 116-134]) there is a unique (up to a.s) $\mathcal{G}$-measurable random variable $\bar{X}$ such that $E[X1_B] = E[\bar{X}1_B]$ for any $B \in \mathcal{G}$. This random variable $\bar{X}$ is called the conditional expectation of X given $\mathcal{G}$, or expectation of X conditional on $\mathcal{G}$, and is denoted by $E[X|\mathcal{G}]$.

Definition Given two random variables X and Y, the conditional expectation $E[X|Y]$ is defined to be $E[X|\sigma(Y)]$, where $\sigma(Y)$ is the σ-algebra generated by all inverse images of measurable subsets (of $\mathbb{R}^n$ or $\mathbb{C}^n$) under Y.

The conditional expectation $E[X|\sigma(Y)]$ is a random variable and is a function of Y. In practice, when X is a function of the form $X = f(Y, Z)$ for independent random variables Y and Z, we can compute $E[X|Y]$ as if Y is a parameter.

In this thesis we may use the following properties of conditional expectation without specifying them (for reference of some of the properties see for example [21, Theorem 2.3.2]).

Theorem 2.5.1 *Let $(\Omega, \mathcal{F}, P)$ be a probability space and let $\mathcal{G}$ be a sub-σ-algebra of $\mathcal{F}$. Then*

1). $E[X|\mathcal{G}]$ is a unbiased estimator of X, i.e.

$$E\left[E[X|\mathcal{G}]\right] = E[X].$$

*2). (**Linearity**) If X and Y are integrable random variables, then*

$$E[aX + bY|\mathcal{G}] = aE[X|\mathcal{G}] + bE[Y|\mathcal{G}],$$

for any constant a and b.

*3). (**Taking out what is known**) If X and Y are integrable random variables, Y is $\mathcal{G}$-measurable, then*

$$E[XY|\mathcal{G}] = YE[X|\mathcal{G}].$$

*4). (**Conditional Jensen's Inequality**) If $f(x)$ is a convex function, i.e. f satisfies $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$ for any $t \in [0,1]$, then*

$$f(E[X|\mathcal{G}]) \leq E[f(X)|\mathcal{G}].$$

We can also define conditional variance.

Definition Let $(\Omega, \mathcal{F}, P)$ be a probability space and let X be an integrable random variable on this space. If $\mathcal{G}$ is a sub-σ-algebra of $\mathcal{F}$, the conditional variance $\mathrm{var}(X|\mathcal{G})$ is defined by

$$\mathrm{var}(X|\mathcal{G}) = E\left[\left(E[X|\mathcal{G}] - E[X]\right)^2 |\mathcal{G}\right].$$

It follows immediately from the definition of conditional variance and property of conditional expectation that

$$\mathrm{var}(X|\mathcal{G}) = E[X^2|\mathcal{G}] - (E[X|\mathcal{G}])^2. \tag{2.4}$$

Conditional variance and conditional expectation are related by the so-called *law of total variance*.

Theorem 2.5.2 *var*$(X) =$ *var*$(E[X|\mathcal{G}]) + E\left[$*var*$(X|\mathcal{G})\right]$.

Proof

$$\begin{aligned}\text{var}(X) - \text{var}(E[X|\mathcal{G}]) &= E[X^2] - (E[X])^2 - \left(E\left[(E[X|\mathcal{G}])^2\right] - (E[X])^2\right) \\ &= E[X^2] - E\left[(E[X|\mathcal{G}])^2\right],\end{aligned}$$

while by equation (2.4)

$$\begin{aligned}E\left[\text{var}(X|\mathcal{G})\right] &= E\left[E[X^2|\mathcal{G}] - (E[X|\mathcal{G}])^2\right] \\ &= E[X^2] - E\left[(E[X|\mathcal{G}])^2\right]. \qquad \blacksquare\end{aligned}$$

As a corollary of Jensen's Inequality we can show the following inequality.

Corollary 2.5.3 *var*$(E[X|\mathcal{G}]) \leq$ *var*(X).

Proof Since $\text{var}(X) - \text{var}(E[X|\mathcal{G}]) = E\left[\text{var}(X|\mathcal{G})\right]$, so we only need to show

$$E\left[\text{var}(X|\mathcal{G})\right] \geq 0.$$

As we see in the proof of Theorem 2.5.2,

$$E\left[\text{var}(X|\mathcal{G})\right] = E[X^2] - E\left[(E[X|\mathcal{G}])^2\right],$$

we only need to show

$$E\left[(E[X|\mathcal{G}])^2\right] \leq E[X^2],$$

which is true by Jensen's inequality applying to the convex function $f(x) = x^2$. $\blacksquare$

Chapter 3

The FFT-Monte Carlo Method

Starting from this chapter, we present three HMC methods. This is the key chapter in this thesis, and it includes a detailed description of the algorithms, an comprehensive error analysis, variance reduction, and its implementation and efficiency.

3.1 The algorithms

In this section we describe the algorithms of the FFT based hybrid Monte Carlo method (FFT HMC).

3.1.1 The FFT HMC

Suppose we want to compute $E[f(Z)]$ where $Z = (z_1, z_2, \cdots, z_d)$ and $(z_1, z_2, \cdots, z_d)$ are independent standard normal random variables. Let $\bar{z} = (z_2, \cdots, z_d)$, then given $\bar{z}$ by Equation (2.1) we can find the Fourier series of $f(Z) = f(z_1, \bar{z})$ with respect to z_1, i.e.

$$f(z_1, \bar{z}) = \sum_{j=-\infty}^{\infty} \hat{f}_j(\bar{z}) e^{i\omega_j z_1} \text{ with } \hat{f}_j(\bar{z}) = \frac{1}{2R} \int_{-R}^{R} f(z_1, \bar{z}) e^{-i\pi j z_1/R} dz_1. \tag{3.1}$$

Remark Note that Equation (3.1) will not make sense unless we restrict z_1 on the finite interval $[-R, R]$. So we are actually computing $E[f(z_1, \bar{z})|z_1 \in [-R, R]]$. But

we should notice that $E[f(z_1, \bar{z})|z_1 \in [-R, R]] \approx E[f(z_1, \bar{z})]$ for moderately large R. This truncation error is expected to decay very fast as R increases (see Section 3.2 for a detailed discussion). In the basket option example we will be discussing, $R = 5$ is large enough to guarantee the truncation error is insignificant compared to errors caused by other reasons.

Then consider the conditional expectation $\bar{f}(\bar{z}) \triangleq E[f(Z) \mid \bar{z}]$. First, since the characteristic function of a standard normal random variable is $E[e^{itz}] = e^{-t^2/2}$, we have

$$E[e^{i\omega_j z_1}] = e^{-\omega_j^2/2},$$

so that

$$\bar{f}(\bar{z}) \approx \sum_{j=-N}^{N} \hat{f}_j(\bar{z}) E[e^{i\omega_j z_1}] = \sum_{j=-N}^{N} \hat{f}_j(\bar{z}) e^{-\omega_j^2/2}. \tag{3.2}$$

By Property 1) of Theorem 2.5.1 we have $E[f(Z)] = E[\bar{f}(\bar{z})]$, so we can apply Monte Carlo method on $\bar{f}(\bar{z})$ to find $E[f(Z)]$.

Therefore the basic 1-D algorithm is as follows:

a). Generate M samples of $\bar{z}$, i.e. $\{\bar{z}_1, \ldots, \bar{z}_M\}$;
b). For each $\bar{z}_n$, find the Fourier coefficients $[\hat{f}_{-N}(\bar{z}_n), \cdots, \hat{f}_N(\bar{z}_n)]$ using the FFT and the 1-D DFT algorithm as described in Section 2.3.2;
c). Find $\bar{f}(\bar{z}_n)$ using equation (3.2);
d). Take the mean of all such $\bar{f}(\bar{z})$, i.e. $\frac{1}{M} \sum_{n=1}^{M} \bar{f}(\bar{z}_n)$.

Algorithm 1: The 1-D FFT HMC algorithm

In the 2-D case the algorithm is similar.

3.2 Error analysis

Approximation in the algorithms described in the last section brings errors. We give an analysis of all possible errors for the 1-D algorithm. There are four sources of error in the 1-D algorithm:

a). Generate samples of $\bar{z} = (z_3, \ldots, z_d)$;
b). For each sample of $\bar{z}$, find the 2-D Fourier series coefficients $\hat{f}_{jk}(\bar{z})$ using 2-D FFT and the 2-D DFT algorithm as described in Section 2.3.2;
c). Find $\bar{f}(\bar{z}) = E[f(Z) \mid \bar{z}]$ using

$$E[f(Z) \mid \bar{z}] \approx \sum_{j=-N_1}^{N_1} \sum_{k=-N_2}^{N_2} \hat{f}_{jk}(\bar{z}) e^{-(\omega_j^2+\omega_k^2)/2};$$

d). Take the mean of all such $\bar{f}(\bar{z})$.

Algorithm 2: The 2-D FFT HMC algorithm

1). The truncation of the payoff function by restricting the domain of z_1 to $[-R, R]$;

2). The truncation of its Fourier series to a finite sum $\sum_{j=-N}^{N} \hat{f}_j(\bar{z}) \cdot e^{i\omega_j z_1}$;

3). The trapezoidal rule approximation with a $2N$-subinterval partition to the Fourier coefficients $\hat{f}_j(\bar{z})$ as described in Section 2.3.2;

4). The bias of Monte Carlo method, i.e. the variance of the Monte Carlo estimator. As this source of error is different in nature from the others, it will be discussed in Section 3.3 and Section 3.4.2.

We will be interested on how significant these errors are and how to reduce them. These errors are close related to the behaviors of the payoff function $f(z_1, \bar{z})$, but we will be primarily concerned about payoff function of the the basket option given in Section 1.2.

3.2.1 Truncation of the payoff function

For each given $\bar{z}$, the truncation of the payoff function produces an error

$$\bar{f}(\bar{z}) - E\left[f(z_1, \bar{z}) 1_{z_1 \in [-R,R]} | \bar{z}\right],$$

In integral, it is equal to

$$\int_{|z_1|>R} f(z_1, \bar{z}) e^{-z_1^2/2} dz_1.$$

As $f(z_1, \bar{z})$ is typically an exponential function of z_1 such as

$$f(z_1, \bar{z}) = e^{-rT} \left(\sum_{i=1}^{7} \beta_i e^{a_i z_1} - K \right)_+$$

in our basket option example, the product $f(z_1, \bar{z}) e^{-z_1^2/2}$ decays as fast as $e^{-(z_1-\mu)^2/2}$ for some μ, so the error is proportional to $1 - N(R - \mu)$ where N is the cumulative distribution function (c.d.f.) of a standard normal random variable. The total error is the expected value of these errors, which is

$$E[f(Z)] - E\left[f(Z) 1_{z_1 \in [-R,R]}\right].$$

In our basket option example, μ can be set to be a fixed number $\texttt{max}\{a_{i1} | i = 1, \ldots, 7\}$. So we can expect that a relatively small R such as $5 + \mu$ or just 5 since μ is small gives an accurate approximation, as it is extremely rare (with probability 2.87×10^{-7}) that a standard normal variable is has absolute value greater than 5.

We can also quantitatively verify the error by Matlab. It turns out if we set $R = 5$ the error is always 0 in our experiments using 10^4 samples of Z, since $|z_1| > 5$ never happens in the experiments.

3.2.2 Truncation of the Fourier series

The second error comes from the truncation of the Fourier series at $-N$ and N, which depends on the convergence rate of the Fourier series.

Theorem 2.3.2 implies that if f is p times differentiable, then the truncation error decays as fast as $1/N^p$ times some constant which depends on p and the p-th derivative of f.

In our example of basket option, for an given $\bar{z}$ the payoff function is continuous with infinite order of derivatives except at the at-the-money point. To be able to apply the theorem, observe that we compute $E[f(z_1, \bar{z})]$ by computing the expectation of

$\sum_{j=-n}^{n} E[\hat{f}_j(\bar{z})]e^{i\omega_j z_1}$, which is the truncated Fourier series of the function

$$g(z_1) = E[f(z_1, \bar{z})|z_1]. \tag{3.3}$$

Despite that $f(z_1, \bar{z})$ is not a smooth function of z_1, the function $g(z_1)$ is smooth. In fact, $g(z_1)$ is n-th order differentiable for any n. See Appendix A for a complete proof.

Figure 3.1 is a Monte Carlo estimation of $g(z_1)$ with 1000 samples at each z_1.

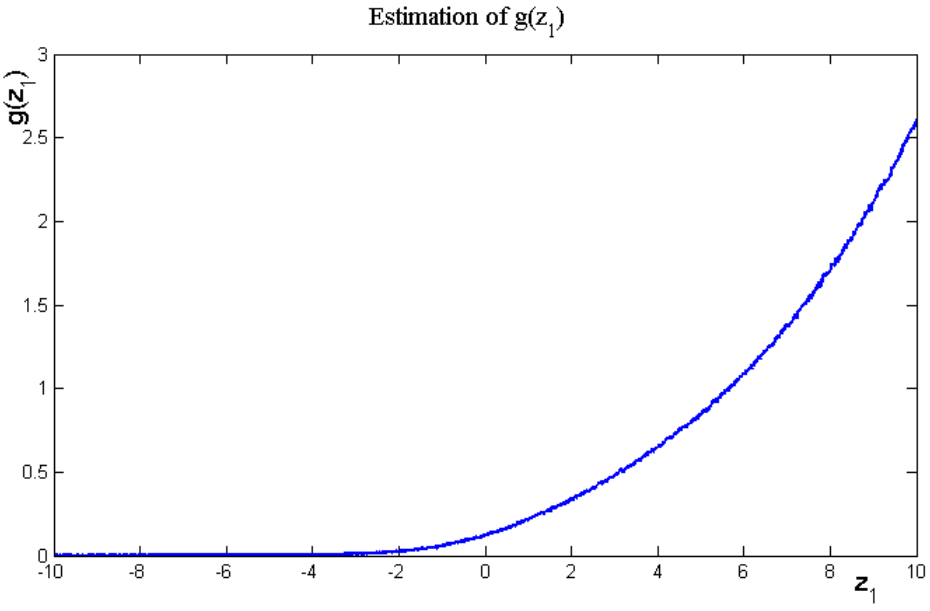

Figure 3.1: A Monte Carlo estimation of $g(z_1)$ with 1000 samples at each z_1

Thus we can apply Theorem 2.3.2 and know that the truncation error decreases rapidly as N become large. In fact in our experiment, $N = 16,\ 32$ are already large enough to ensure the truncation error is insignificant.

For example we can fix a set of samples of $\bar{z}$ and increase N to see how the increase of N change the value of the final results. Figure 3.2 shows the change of the computed option prices when N increase from 4 to 64 (with $n = 1000$, $R = 5$).

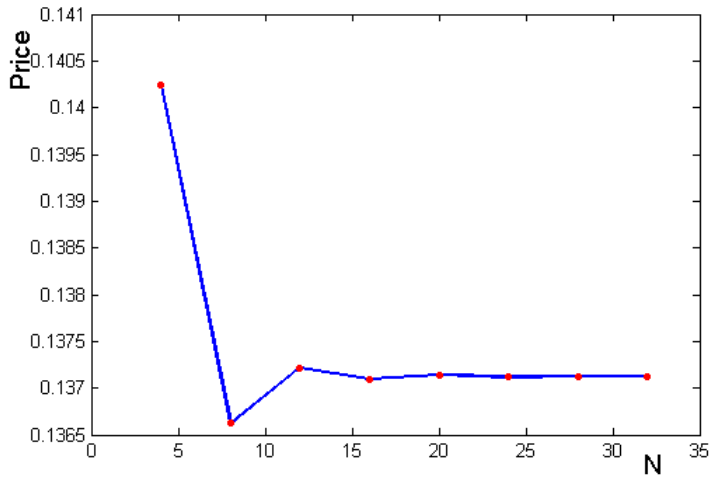

Figure 3.2: Change of option prices when N increases from 4 to 64 (with $n = 1000$, $R = 5$)

Figure 3.2 shows a plot of computed prices as a function of N. As we can see from Figure 3.1, starting from $N = 16$, the results become stable.

3.2.3 Trapezoidal rule approximation

In our algorithm, when computing $\hat{f}_j(\bar{z})$, we use the discrete Fourier coefficients $\tilde{f}_j(\bar{z})$ with a $2N$-points partition for a given $\bar{z}$ and then take the expected value to get $\tilde{g}_j$. One would expect a large N to control the trapezoidal rule error. But we observe that there is an aliasing relation (see for example Equation (8.1.32) in Canuto et.al (2007) [5])

$$\tilde{g}_j - \hat{g}_j = \sum_{k \neq 0} \hat{g}_{j+2kN}.$$

Using this relationship we see that the approximation error is bounded by $|\sum_{k\neq 0} \hat{g}_{j+2kN}|$, which decays at least as fast as e_{2N}. As we can see from the previous section that e_{2N} decreases rapidly, the approximation error can be controlled using a large number N. In our basket option example, $N = 16$ or 32 is good enough to ensure the error is insignificant.

3.3 Variance reduction by orthogonal transformation

In this section we discuss the fourth type of error: the error of Monte Carlo simulation, and present a new variance reduction technique for our HMC. First we note that $\bar{f}(\bar{z})$ has smaller variance than $f(Z)$, so if the cost of FFT is negligible, by Corollary 2.5.3 we have already reduced the error by using $\bar{f}(\bar{z})$ instead of $f(Z)$.

We know that if X is a d by d orthogonal matrix, then XZ has the same distribution as Z. Now define $f_X(Z) = f(XZ)$, then $E[f_X(Z)] = E[f(Z)]$, also $f_X(Z)$ and $f(Z)$ has the same variance. But $\bar{f}_X(\bar{z})$ and $\bar{f}(\bar{z})$ will have different variance. So by choosing a proper matrix X, we can further reduce the variance.

Here we propose two ways of finding an orthogonal matrix X that may significantly reduce the variance.

First we can look at the Maclaurin series of $f_X(Z) = f_X(z_1, \bar{z})$ with respect to z_1:

$$f_X(Z) = \sum_{n=0}^{\infty} \frac{z_1^n}{n!} \frac{\partial^n f_X}{\partial z_1^n}\bigg|_{z_1=0},$$

hence

$$\begin{aligned} \bar{f_X}(\bar{z}) &= E[f_X(Z)|\bar{z}] \\ &= \sum_{n=0}^{\infty} \frac{E[z_1^n]}{n!} \frac{\partial^n f_X}{\partial z_1^n}\bigg|_{z_1=0} \\ &= \sum_{n=0}^{\infty} \frac{1}{2^n n!} \frac{\partial^{2n} f_X}{\partial z_1^{2n}}\bigg|_{z_1=0} \end{aligned}$$

since the odd moments of z_1 are zero and the $2n$-th moment is equal to $1 \cdot 3 \cdots (2n-1)$.

Then we can find that the first few terms of $\mathrm{Var}(f_X(Z)) - \mathrm{Var}(\bar{f_X}(\bar{z})) = E[{f_X}^2(Z)] - E[{\bar{f_X}}^2(\bar{z})]$ are

$$E\left[\left(\frac{\partial f_X}{\partial z_1}\bigg|_{z_1=0}\right)^2\right] + \frac{1}{2}E\left[\left(\frac{\partial^2 f_X}{\partial z_1^2}\bigg|_{z_1=0}\right)^2\right] - \frac{1}{2}E\left[f_X(0,\bar{z})\frac{\partial^2 f_X}{\partial z_1^2}\bigg|_{z_1=0}\right],$$

and a big contributing part of $E\left[\left(\left.\frac{\partial f_X}{\partial z_1}\right|_{z_1=0}\right)^2\right]$ is $\left(\left.\frac{\partial f_X}{\partial z_1}\right|_{Z=0}\right)^2$, so we can try to maximize $\left(\left.\frac{\partial f_X}{\partial z_1}\right|_{Z=0}\right)^2$, or the absolute value of $\left.\frac{\partial f_X}{\partial z_1}\right|_{Z=0}$. We will see that this idea works very well in the basket option example.

Another approach is to find an X by experiment. Fix a set of sample of $\bar{z}$, compute the variance of $\bar{f_X}(\bar{z})$ for different choices of X and find the one that gives the smallest variance. In Matlab it can be done using the function *fminsearch*. For example, we can find a better X by using fminsearch near the X we find by the first approach.

One problem of this approach is that the limitation of fminsearch may not lead us to the best X. It is because when the size X is big (e.g. 7×7), the number of function evaluations becomes large. Also if we find the best X for a given sample set, it is usually not the best for other sample sets. But still it gives us some ideas about the choices of X.

3.4 Implementation and efficiency

In this section we implement the ideas in Section 3.1 and Section 3.3 on the basket option described in Section 1.2, and compare the efficiency with regular Monte Carlo method.

3.4.1 Finding an orthogonal matrix X

To use the variance reduction method by orthogonal transformation, as described in Section 2.2, i.e. to find an orthogonal matrix X so that the variance of $\bar{f}_X(\bar{Z})$ is smaller than that of $f_X(Z)$, we try the first approach in Section 3.3. Let $\alpha = [\alpha_1, \cdots, \alpha_7]$ and let a be the first column of X. We first find that when $f_X(Z) =$

```
function X = createX(a);
% given a column vector a, construct an orthogonal matrix
% with a/norm(a) as its first column

a = a/norm(a);
X = [a null(a')]';
end
```

Figure 3.3: Code for creating an orthogonal matrix X with the first column given

$e^{-rT}\left(\sum_{i=1}^{7}\alpha_i e^{(AXZ)_i}-K\right)_+ > 0$

$$\left.\frac{\partial f_X}{\partial z_1}\right|_{Z=0} = e^{-rT}\sum_{i=1}^{7}\alpha_i(Aa)_i e^{(AXZ)_i}\Bigg|_{Z=0}$$
$$= e^{-rT}\alpha Aa.$$

The absolute value of $e^{-rT}\alpha Aa$ is maximized when

$$a = \frac{A^T\alpha^T}{\parallel A^T\alpha^T \parallel}.$$

Thus we can find X by creating an orthogonal matrix with $a = \frac{A^T\alpha^T}{\parallel A^T\alpha^T \parallel}$ as its first column. This can be done using the Matlab function "null" as indicated in the code[1] shown in Figure 3.3.

It turns out that in this example, the X obtained this way actually works surprisingly well. With such an orthogonal transformation, the radius of the 95% confidence interval is reduced by an average of 87%, which is a dramatic improvement.

3.4.2 Efficiency Improvement

With the X we found in the previous section, we can implement the 1-D algorithm in Section 3.1 in Matlab. One may find that the hybrid Monte Carlo algorithm runs slower that regular Monte Carlo if the code is not properly optimized. The running time of the hybrid Monte Carlo code can be improved in some ways:

[1] original idea by Dr. A Ware

```
function B=basketpayoff4all(x,z)

load basketinfo a b alpha r T K
B=zeros(length(x),size(z,2));
for k=1:7
    B=B+alpha(k)*exp(a(k)*x)'*exp(b(k,:)*z);
end
B=exp(-r*T)*max(B-K,0);
end
```

Figure 3.4: Payoff function code that return a matrix of payoffs with size $\text{length}(x) \times \text{size}(z, 2)$

1). When finding the Fourier coefficients, for each sample of $\bar{z}$ we need to evaluate the payoff function $f(z_1, \bar{z})$ at $z_1 = \frac{Rn}{N}$ for $n = -N, -N+1, \ldots, N-1$. Instead of writing a function that only accept scalar inputs, we can build a function $f(z_1, \bar{z})$ that accept vector input of z_1 and $\bar{z}$. For input $z_1 = [z_1^1, \ldots, z_1^k]$ and $\bar{z} = [\bar{z}_1, \ldots, \bar{z}_l]$, the function should return a $k \times l$ matrix with $f(z_1^i, \bar{z}_j)$ as its entries. The function code shown in Figure 3.4 can achieve this.

In the code, "basketinfo" is a .mat file that contains all data of the basket option. For instance, "alpha" is the vector α, and r, T, K are the interest rate, expiration, strike respectively. "a" is the first column of A (or AX if we want to incorporate an orthogonal matrix X), and "b" is the sub-matrix of A (or AX) formed by removing a from A (or AX).

2). For each $\bar{z}$, we can compute $\hat{f}_j(\bar{z})$ for $j = -N, \cdots, N$ by a FFT on the vector u which depends on $\bar{z}$ as we can see from the algorithm described in Section 2.3.2. The "fft" function in Matlab can operate on matrices so we can compute a matrix with entries $\hat{f}_j(\bar{z}_n)$ in a single FFT by first constructing a matrix with the u's as columns. The code is in Figure 3.5.

In the 2-D DFT case, we can construct a 3-D array and them apply the Matlab function fft2. The code of finding 2-D Fourier coefficients is in Figure 3.6.

3). The variance reduction techniques described in Section 2.2 can also be applied.

```
function F=FourierCoef4all(f,zbar,N,R)

x=[(-N+1):(N-1)]*R/N;
L=size(zbar,2);
u=zeros(2*N,L);
u=[(f(-R,zbar)+f(R,zbar))/2;f(x,zbar)];
ind = [(N+1):2*N, 1:N];
u =u(ind,:);
w=fft(u)/(2*N);
w=w(ind,:);
F=[w;w(1,:)];
end
```

Figure 3.5: Fourier Coefficients Code that returns Fourier coefficients for all $\bar{z}$ at the same time

The author has tried antithetic variables and control variates, and it turns out that antithetic variables technique does not reduce the variance but the control variates technique does significantly reduce the variance.

Using the notation in Section 2.2.2, in our problem we set $V = E[f_X(Z)|\bar{z}]$, which is computed using the FFT method. We choose the control variate as

$$V^c = E[h_X(Z)|\bar{z}],$$

where

$$h_X(Z) = e^{-rT}\left(\sum_{i=1}^{7} \alpha_i e^{(AXZ)_i} - K\right)$$

since it is strongly correlated to V.

The control variate V^c can be generated using both FFT and an analytical method. If we use the FFT, generating V^c is exactly the same as generating V. If we use the analytical method, we first write

$$\sum_{i=1}^{7} \alpha_i e^{(AXZ)_i} = \sum_{i=1}^{7} \beta_i e^{a_i z_1}$$

like we did in Section 1.2 but with A replaced by AX, so $\beta_i = \alpha_i e^{(B_X \bar{z})_i}$ and B_X is the sub-matrix of AX obtained by removing the first column of AX, and $a = [a_1, \ldots, a_7]^t$

```
function F=DoubleFourierCoef4all(f,zbar,N1,N2,R)

L=size(zbar,2);
u=zeros(2*N1,2*N2,L);
x=[(-N1):(N1-1)]*R/N1;
y=[(-N2):(N2-1)]*R/N2;;
for l=1:L
u(:,:,l)=f(x,y,zbar(:,l));
end

ind1=[N1+1:2*N1,1:N1];
ind2=[N2+1:2*N2,1:N2];
u=u(ind1,ind2,:);
w=fft2(u)/(4*N1*N2);
F=w(ind1,ind2,:);
end
```

Figure 3.6: Double Fourier Coefficients Code

is the first column of AX. Then we can calculate

$$\begin{aligned} V^c = E[h_X(Z)|\bar{z}] &= e^{-rT}\left(\sum_{i=1}^{7}\beta_i E[e^{a_i z_1}] - K\right) \\ &= e^{-rT}\left(\sum_{i=1}^{7}\beta_i e^{\frac{a_i^2}{2}} - K\right). \end{aligned} \tag{3.4}$$

In the case of the analytical control variate, the cost of the control variate is negligible compared to the cost of generating V, therefore is better than generating V^c using FFT. One can use the code shown in Figure 3.7.

The estimated radius of the 95% confidence interval using a control variate is less than $\frac{1}{9}$ of the radius without the control variate.

3.4.3 Efficiency Comparison

If we use a $2N$-grid for the FFT and generate M sample of $\bar{f}(\bar{z})$, the desired efficiency is that it is at least as efficient as a regular Monte Carlo that uses $2NM$ samples. Figure 3.8 shows the comparison between the estimated errors of the hybrid Monte Carlo using M samples and regular Monte Carlo using $64M$ sample (with $R = 5$, $N = 32$).

```
function P=control_variate(R,N,n)

load basketinfo alpha A X r T K;
Z=randn(6,n);
f=@(x,z)basketpayoff4all_X(x,z,X);
Coef=FourierCoef4all(f,Z,N,R);
E=exp(-1/2*(pi*[-N:N]/R).^2);
A=A*X;
s=sum(A.^2,2);
mu=exp(-r*T)*(dot(alpha,exp(s/2))-K);
V=E*Coef;
a=A(:,1);
beta=diag(alpha)*exp(A(:,2:7)*Z);
VC=exp(-r*T)*((exp(a.^2/2))'*beta-K);
c=cov(V,VC); theta=c(1,2)/c(2,2);
CondExp=V-theta*(VC-mu);
P=mean(CondExp);
P=real(P);
err=1.96*std(CondExp)/sqrt(n);
P=[P err];
```

Figure 3.7: Control Variate Code that uses V^c in Equation (3.4)

We can see that in our example, the hybrid Monte Carlo performs better than a regular Monte Carlo. After the improvements as mentioned above, the hybrid Monte Carlo also runs much faster than a regular Monte Carlo. Figure 3.9 shows the cpu time run by the two methods to achieve the same level of accuracy.

The second way of finding X described in Section 3.3 is also tested. It appears that the X we had from the first approach is already quite optimal, and doing fminsearch does not significantly reduce the variance.

3.4.4 More about the 2-D case

Finding an orthogonal matrix X to reduce the variance in the 2-D case is different from the 1-D case. We can seek to maximize $\left(\frac{\partial f_X}{\partial z_1}\Big|_{Z=0}\right)^2 + \left(\frac{\partial f_X}{\partial z_2}\Big|_{Z=0}\right)^2$ instead. In our example, it is equal to $(\alpha Aa)^2 + (\alpha Ab)^2$ where a, b are the first two columns of X. It is maximized when the $\alpha^T A^T$ is a linear combination of a and b. So we can construct such an orthogonal matrix X, and implement the algorithm as described

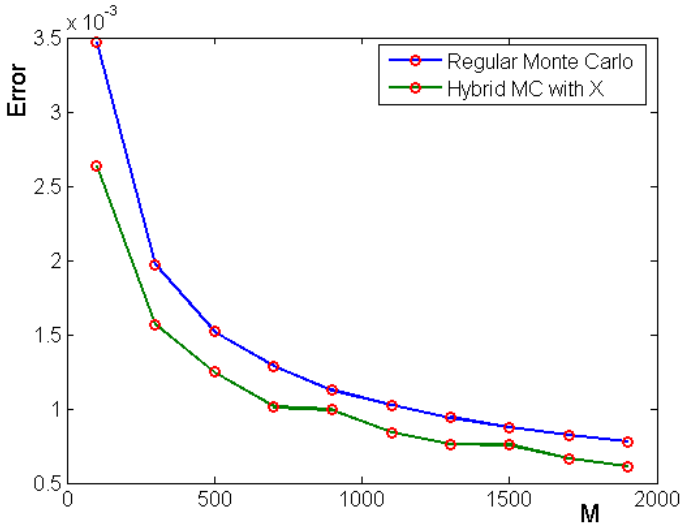

Figure 3.8: Error comparison between the estimated errors of the hybrid Monte Carlo using M samples and regular Monte Carlo using $64M$ sample (with $R = 5$, $N = 32$)

in Section 2.1.

Using 2-D DFT still works to get the correct price but it turns out to be not so satisfying compared to regular Monte Carlo method in terms of efficiency.

3.5 Possible further development of this method

There are several aspects of this idea that worths exploring.

- As one can see that the way of finding the orthogonal matrix X is based on a rough idea. There may be ways of finding better Xs.
- If we look at the efficiency of the hybrid Monte Carlo method, we can see that there is a bottle-neck to this method. Suppose we use a $2N$-grid for the FFT and generate M samples of $\bar{f}(\bar{z})$, then the asymptotic error of this method is $E(N) + \frac{k_1}{\sqrt{M}}$ where $E(N)$ is the error of computing Fourier series using $2N$ terms and $2N$-grid for the coefficients. The asymptotic error of a regular Monte Carlo method using $2NM$ is $\frac{k_2}{\sqrt{2NM}}$.

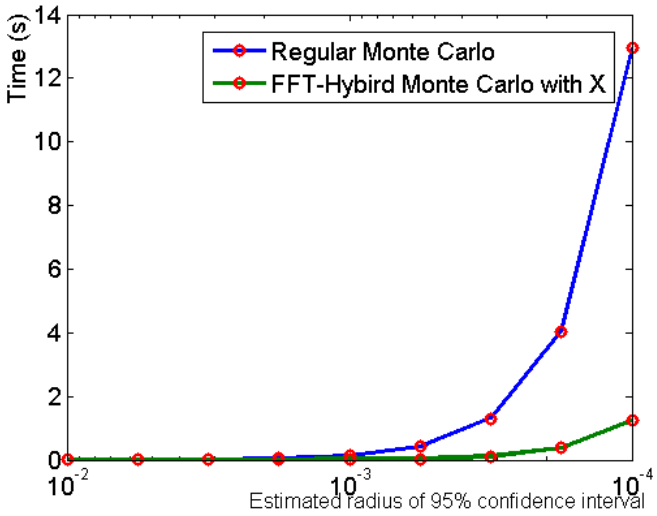

Figure 3.9: cpu time run by FFT HMC with orthogonal transformation by X and regular MC to achieve the same level of accuracy, x-axis in log scale. Run on PC with Intel Core 2 Duo T5250 1.5GHz, 2GB Memory

The error $E(N)$ usually decays much faster than the order $\frac{1}{\sqrt{N}}$. To make $E(N)$ small we need to increase N. But when N is increasing, the regular Monte Carlo method will eventually outperform the hybrid method.

Dr. A. Ware proposed a possible solution to this problem that may be called the "wedding cake" algorithm with aliasing correction, which is not included in this thesis.

Nevertheless we expect N does not need to be very large (for example in the basket option example) to guarantee that $E(N)$ is sufficiently small.

Chapter 4

The Cos-Monte Carlo Method

In this chapter we present a HMC method similar to the FFT HMC in the previous chapter. This method is based on cosine expansion, so its called COS HMC. But due to the structure of the payoff function, in the algorithm we can use some analytical formulas to compute the conditional expectation.

4.1 The Basic Idea

The COS HMC is similar to the FFT HMC method. The difference is that in the Cos-Monte Carlo method, the payoff function $f(z_1, \bar{z})$ is approximated by its cosine expansion, and the cosine expansion coefficients are computed using closed-form formulas.

By Equation (2.2) and (2.3) we can express the cosine expansion of $f(z_1, \bar{z})$ as

$$\frac{1}{2}c_0 + \sum_{k=1}^{\infty} c_k \cos\left(k\pi \frac{z_1 - a}{b-a}\right),$$

with

$$c_k = \frac{2}{b-a}\int_a^b f(z_1, \bar{z}) \cos\left(k\pi \frac{z_1 - a}{b-a}\right) dz_1, \ k = 0, 1, \ldots$$

The conditional expectation is

$$E[f(z_1, \bar{z})|\bar{z}] = \frac{1}{2}c_0 + \sum_{k=1}^{\infty} c_k E\left[\cos\left(k\pi \frac{z_1 - a}{b-a}\right)\right].$$

In this equation, $E\left[\cos\left(k\pi\frac{z_1-a}{b-a}\right)\right]$ has a closed form expression, and in some cases the Fourier coefficients a_k also have closed form formulas. Therefore given a sample of $\bar{z}$, we can get a sample of $E[f(z_1,\bar{z})|\bar{z}]$ using these formulas and then Monte Carlo method can be applied. We apply this idea to the basket option pricing in the following sections.

4.2 Applying to the Basket option

As described in Section 1.2, the basket option has payoff function

$$e^{-rT}\left(\sum_{i=1}^{7}\alpha_i e^{(AZ)_i}-K\right)_+.$$

Given $\bar{z}$, to calculate the expectation of the payoff function conditional on $\bar{z}$, we need to know the value of z_1 so that the payoff is zero. It happens that the first column of A and α_i are all positive[1], therefore given $\bar{z}$, $\sum_{i=1}^{7}\alpha_i e^{(AZ)_i}-K=0$ has a unique solution $z_1=\gamma(\bar{z})$. While there is usually no closed form solution to r, the root can be found quite efficiently using Newton's method.

First we write

$$\sum_{i=1}^{7}\alpha_i e^{(AZ)_i}=\sum_{i=1}^{7}\beta_i e^{a_i z_1}$$

as we did in Section 1.2. The the iteration formula for Newton's method is

$$\gamma_{n+1}=\gamma_n-\frac{\sum_{i=1}^{7}a_i\beta_i e^{a_i z_1}}{\sum_{i=1}^{7}\beta_i e^{a_i z_1}-K}. \tag{4.1}$$

Using Matlab we can build a function that can take input matrix in which every column is a sample of $\bar{z}$ and returns a vector of estimated roots at the same time. It turns out that Newton's method converges very fast that in most cases it only needs less than 5 iterations with a tolerance of 10^{-4}.

Now let $a=-R$, $b=R$ for some chosen $R>0$, and $f(z_1,\bar{z})=\left(\sum_{i=1}^{7}\alpha_i e^{(AZ)_i}-K\right)_+$. Then $E\left[\cos\left(k\pi\frac{z_1-a}{b-a}\right)\right]=E\left[\cos\left(\frac{k\pi z_1}{2R}+\frac{k\pi}{2}\right)\right]$. To find a formula for $E\left[\cos\left(\frac{k\pi z_1}{2R}+\frac{k\pi}{2}\right)\right]$,

[1]The method described in this Chapter relies on this fact

first note that

$$\cos\left(\frac{k\pi z_1}{2R}+\frac{k\pi}{2}\right)=\cos\frac{k\pi z_1}{2R}\cos\frac{k\pi}{2}-\sin\frac{k\pi z_1}{2R}\sin\frac{k\pi}{2}.$$

Then since the characteristic function of the standard normal random variable z_1 is

$$E[e^{itz_1}]=E[\cos tz_1+i\sin tz_1]=e^{-\frac{t^2}{2}},$$

so we have $E[\cos tz_1]=e^{-\frac{t^2}{2}}$ and $E[\sin tz_1]=0$. Then we can get

$$E\left[\cos\left(\frac{k\pi z_1}{2R}+\frac{k\pi}{2}\right)\right]=\begin{cases}0, & \text{if k is odd}\\ (-1)^{k/2}e^{-\frac{1}{2}(\frac{k\pi}{2R})^2}, & \text{if k is even}\end{cases}$$

The COS coefficients are

$$c_0=\frac{1}{R}\int_\gamma^R(\sum_{i=1}^7\beta_i e^{a_i z_1}-K)dz_1$$
$$c_k=\frac{1}{R}\int_\gamma^R\left(\sum_{i=1}^7\beta_i e^{a_i z_1}-K\right)\cos\left(\frac{k\pi z_1}{2R}+\frac{k\pi}{2}\right)dz_1.$$

Integration gives

$$c_0=\frac{1}{R}\sum_{i=1}^7\beta_i\frac{e^{a_iR}-e^{a_i\gamma}}{a_i}-\frac{K}{R}(R-\gamma) \tag{4.2}$$

and

$$c_k=\sum_{i=1}^7\left[\frac{2\beta_i\left[c_{k,i}e^{a_iR}(-1)^k-e^{a_iR}(\sin\theta_k+c_{k,i}\cos\theta_k)\right]}{k\pi(1+c_{k,i}^2)}\right]+\frac{2K}{k\pi}\sin\theta_k \tag{4.3}$$

with

$$c_{k,i}=\frac{2a_iR}{k\pi}\text{ and }\theta_k=\frac{k\pi}{2R}(R+\gamma). \tag{4.4}$$

Now for some large N we can compute $E[f(z_1,\bar{z})]$ by

$$E[f(z_1,\bar{z})|\bar{z}]\approx\frac{1}{2}c_0+\sum_{k=1}^N c_{2k}(-1)^k e^{-\frac{1}{2}(\frac{k\pi}{R})^2}. \tag{4.5}$$

In summary the algorithm is as follows.

a). Generate samples of $\bar{z}$;
b). Find the roots $\gamma(\bar{z})$ by Newton's method using formula (4.1)(by experiment we found that most roots are close to 0 so the initial guess is set to be 0).
c). Find c_0 and c_k using equations (4.2), (4.3) and (4.4).
d). Compute $E[f(z_1, \bar{z})|\bar{z}]$ using Equation (4.5).
e). Compute the mean of all such $E[f(z_1, \bar{z}_i)|\bar{z}_i]$.

Algorithm 3: The COS HMC algorithm

4.3 Implementation

Using Matlab we can implement the algorithm and compare it to regular Monte Carlo and the FFT based hybrid Monte Carlo described earlier.

Once again to improve efficiency we need to find a way of computing all a_0 and a_k for all samples of $\bar{z}$ at the same time using matrix operation. The code in Figure 4.1 incorporates these ideas.

The variance reduction by orthogonal transformation X still has significant effect in variance reduction. The same control variate in (3.4) can be used here. Similar to the case of FFT-based hybrid Monte Carlo, the estimated radius of 95% confidence interval using control variate is less than $\frac{1}{9}$ of the radius without control variate.

We would expect that the COS-based hybrid Monte Carlo (COS HMC) runs faster than the FFT-based Monte Carlo (FFT HMC) since the COS HMC uses analytic formulas to compute the COS series coefficients. But this is not the case in reality, at least for the codes written by the author. It is mainly because of the computation of the roots by Newton's method and the complexity of the formula (4.3).

For example, if we use $R = 5$, $N = 16$, $n = 1000$, and the orthogonal matrix X found in Section 3.4.1, the COS HMC takes about twice the time of FFT HMC. But the COS HMC is still better than the regular Monte Carlo in terms cpu time needed to achieve a given accuracy. See Chapter 6 for the comparison.

```
function [p,err]=CosBasketFaster(R,N,n,Y)

load basketinfo A alpha K r T X;
if nargin<4, Y=X; end
A=A*Y;
payoff4all=@(z,zbar) alpha*exp(A*[z;zbar])-K;
a=A(:,1);
dif4all=@(z,zbar) alpha.*a'*exp(A*[z;zbar]);
zbar=randn(6,n); maxiter=15; tol=1e-4;
x0=zeros(1,M); k=1; err=1; x=x0;
    while (err >= tol) && (k <= maxiter)
        y = x - payoff4all(x,zbar)./dif4all(x,zbar);
        err = max(abs(x-y));
        x=y;
        k = k+1;
    end
% using newton's method to find all roots at once
root=x;
beta=diag(alpha)*exp(A(:,2:7)*zbar); % beta is 7 by n
A0=sum(diag(1./a)*((exp(a*root).*(exp(a*(R-root))-1)).*beta))/R-...
K/R*(R-root);
Condexp=A0/2; theta=pi*(1+root/R); b=a*R/pi;
    for k=1:N
        kb=b/k;
        coeff=exp(a*R)./(1+kb.^2);
        A2k=1/(k*pi)*sum(beta.*(diag(coeff)*(repmat(kb,1,n)...
        -exp(a*(root-R)).*(repmat(sin(k*theta),7,1)...
        +kb*cos(k*theta)))))+K/(k*pi)*sin(k*theta);
        Condexp=Condexp+A2k*(-1)^k*exp(-(k*pi/R)^2/2);
    end
p=exp(-r*T)*mean(Condexp);
err=1.96*exp(-r*T)*std(Condexp)/sqrt(n);
```

Figure 4.1: COS expansion based Hybrid Monte Carlo Code, the input Y is the orthogonal matrix for variance reduction. If only the first three input arguments are given, X is by default the X we found in Section 3.4.1

Chapter 5

The Black Scholes Hybrid Monte Carlo Method

The hybrid Monte Carlo methods in this thesis follow one idea, i.e. first find the conditional expectation with respect to all but one or two dimension of Z using methods such as Fourier transform, COS transform, then use Monte Carlo on the conditional expectation. In this Chapter we introduce another hybrid Monte Carlo method in which the conditional expectation has a closed form expression which is similar to the Black-Scholes formula. We call this method the BS HMC.

5.1 The Algorithm

This method is demonstrated using our basket option example and is only applicable when the payoff function satisfies some conditions. This method exploits the fact that in the payoff function $e^{-rT}\left(\sum_{i=1}^{7}\alpha_i e^{(AZ)_i}-K\right)_+$ the matrix A is lower triangular (which is in general true in the setting of European style options). In fact, in the basket option example, as we see in Section 1.2

$$A=\sqrt{T}\sigma L$$

where L is the Cholesky decomposition of the covariance matrix C, i.e. a lower triangular matrix such that $C = LL^t$, and σ is the diagonal matrix with volatilities σ_i as its diagonal entries. Now write

$$A = \begin{bmatrix} a_{11} & & 0 & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{71} & a_{72} & \cdots & a_{77} \end{bmatrix}$$

and let $a = a_{77}$, $\bar{z} = [z_1,\ z_2,\ \cdots,\ z_6]^t$, $f(Z) = \left(\sum_{i=1}^{7} \alpha_i e^{(AZ)_i} - K\right)_+$. Observing that z_7 only appear once in the expression, we can write

$$\begin{aligned} f(Z) &= (\alpha_1 e^{a_{11}z_1} + \alpha_2 \exp(a_{21}z_1 + a_{22}z_2) + \cdots + \alpha_7 \exp(a_{71}z_1 + a_{72}z_2 + \cdots + a_{77}z_7) - K)_+ \\ &= \left(\bar{\beta} e^{az_7} + \bar{K}\right)_+ \end{aligned}$$

with

$$\bar{\beta} = \alpha_7 \exp(a_{71}z_1 + a_{72}z_2 + \cdots + a_{76}z_6) \tag{5.1}$$

and

$$\bar{K} = \alpha_1 e^{a_{11}z_1} + \alpha_2 \exp(a_{21}z_1 + a_{22}z_2) + \cdots + \alpha_6 \exp(a_{61}z_1 + a_{62}z_2 + \cdots + a_{66}z_6) - K \tag{5.2}$$

so $\bar{\beta}$ and $\bar{K}$ are functions of $\bar{z}$.

We have to consider the cases when $\bar{K}$ is nonnegative and negative. When $\bar{K} \geq 0$, there is no z making $\bar{\beta} e^{az_7} + \bar{K} = 0$ since $\bar{\beta}$ is positive. If $\bar{K} < 0$, the root is $r = \frac{1}{a} \ln \frac{-\bar{K}}{\bar{\beta}}$.

Now we can compute the conditional expectation $E[f(Z)|\bar{z}]$ just like the Black-Scholes formula:

$$\begin{aligned} E[f(Z)|\bar{z}] &= E[\left(\bar{\beta} e^{az_7} + \bar{K}\right)_+ |\bar{z}] \\ &= \begin{cases} \bar{\beta} e^{\frac{a^2}{2}} N(a - r) + \bar{K} N(-r), \text{ if } \bar{K} < 0 \\ \bar{\beta} e^{\frac{a^2}{2}} + \bar{K}, \text{ if } \bar{K} \geq 0 \end{cases} \end{aligned} \tag{5.3}$$

where N is the CDF of standard normal random variable.

So the algorithm is as follows.

a). Generate samples of $\bar{z}$;
b). Find $\bar{\beta}$ and $\bar{K}$ by Equations (5.1) and (5.2).
c). Compute $E[f(Z)|\bar{z}]$ by Equation (5.3).
d). Compute the mean of all such $E[f(Z)|\bar{z}_i]$.

Algorithm 4: BS HMC algorithm

5.2 Implementation

As before, after samples of $\bar{z}$ have been generated, we should find $\bar{\beta}$ and $\bar{K}$ all at once for all $\bar{z}$ by using matrix operations. Then we need to find all indices of $\bar{K}$ that are negative and nonnegative, and then use the corresponding formula in Equation (5.3). The control variate used in Chapter 3 and 4 can still be used here to achieve a significant reduction of variance. The code shown in Figure 5.1 can be used to implement these ideas.

```
function P=Basket_AnalyticZn_VC(M)

load basketinfo A alpha K r T
zbar=randn(6,M); A_bar=A(1:6,1:6);
a=A(7,7); beta_bar=alpha(7)*exp(A(7,1:6)*zbar);
K_bar=alpha(1:6)*exp(A_bar*zbar)-K;
ind_p=find(K_bar>0); % indices with positive entries
ind_n=find(K_bar<=0); % indices with negative entries
K_bar_p=K_bar(ind_p); beta_bar_p= beta_bar(ind_p);
K_bar_n=K_bar(ind_n); beta_bar_n= beta_bar(ind_n);
root=log(-K_bar_n./beta_bar_n)/a; % the roots
condexp_n=beta_bar_n*exp(a^2/2).*normcdf(a-root)+...
K_bar_n.*normcdf(-root);
condexp_p=beta_bar_p*exp(a^2/2)+K_bar_p;
VC=beta_bar*exp(a^2/2)+K_bar; condexp=[condexp_p, condexp_n];
VC=VC([ind_p, ind_n]); c=cov(condexp,VC); theta=c(1,2)/c(2,2);
s=sum(A.^2,2); mu=dot(alpha,exp(s/2))-K;
condexp=condexp-theta*(VC-mu); p=exp(-r*T)*mean(condexp);
err=1.96*exp(-r*T)*std(condexp)/sqrt(M); P=[p err];
```

Figure 5.1: Black-Scholes-like Hybrid Monte Carlo method code

Note that the orthogonal matrix X we used in Chapter 3 and 4 can not be used here. If we want to apply the idea of orthogonal transformation here, the orthogonal

matrix X has to satisfy two conditions: 1) AX has to have a column with only one nonzero entry for this algorithm to work; 2) X has to be chosen so that the variance is reduced.

For example, if we want to find an orthogonal matrix X so that the 6-th column of AX has the only nonzero entry at $(6, 6)$ entry, then the 6-th column of X must be

$$X_6 = \frac{A^{-1}[0\ 0\ 0\ 0\ 0\ 1\ 0]^t}{\|A^{-1}[0\ 0\ 0\ 0\ 0\ 1\ 0]^t\|}.$$

This condition will also guarantee that 6-th column of AX has the only nonzero entry at $(6, 6)$ entry.

Then we can create an orthogonal matrix X with X_6 as its 6-th column. But there are many ways to do this. The author believe that there exist X's that will significantly reduce the variance but failed to find one after several attempts.

Chapter 6

Conclusions

In this chapter we present comparisons of the efficiency of all HMC methods and the regular Monte Carlo followed by a discussion on possible further work.

6.1 Comparison of all methods

So far we have introduced three HMC methods: the FFT HMC, the COS HMC, and the BS HMC. These methods are applied to the basket option described in Section 1.2. In this Chapter we conclude the thesis with more comparison among these methods and regular Monte Carlo.

With an orthogonal transformation, the first two methods are much more efficient than a regular Monte Carlo. Similar to Figure 3.8, we can compare the estimated errors of MC, FFT HMC, and COS HMC. Control variate techniques can be applied to all these methods. Figure 6.1 shows the results.

For the BS HMC, the author didn't find an orthogonal matrix X to significantly reduce the variance. While no such X is found, the BS HMC have a similar level of efficiency as regular Monte Carlo. Figure 6.2 compares BS HMC and MC.

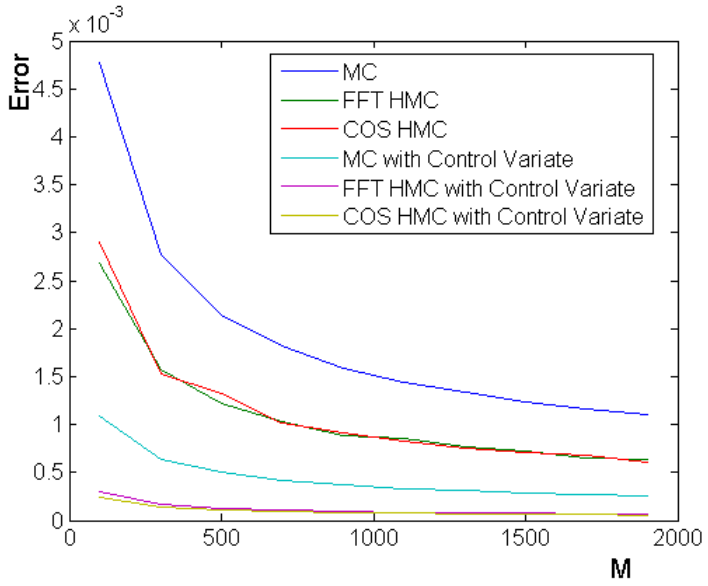

Figure 6.1: Error comparison of the estimated radius of 95% confidence interval for the FFT HMC, COS HMC using M samples and regular Monte Carlo using $32M$ sample; the control variate version of all three methods are also included. ($R = 5$, $N = 16$)

6.2 Possible Further Work

The ideas in this thesis can be extended or improved in the following directions.

- As mentioned in the end of Chapter 3, Dr. A. Ware proposed an algorithm called the "wedding cake" algorithm that can further improve the efficiency of the FFT HMC.

- In the BS HMC we are not able to find an orthogonal matrix to significantly reduce the variance. However the author believes that there should be a way of finding such a matrix that will make the BS HMC be as efficient as the other HMC introduced in this thesis.

- The HMC methods introduced in this thesis are intended for valuation of multi-assets derivatives. Besides basket options such as the example in Section 1.2, we can also apply the methods to spread options and

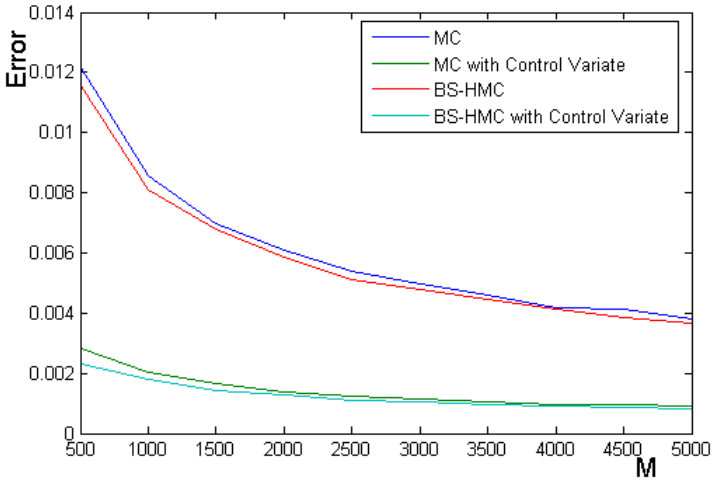

Figure 6.2: Error comparison of the estimated radius of 95% confidence interval for the BS HMC and regular Monte Carlo using the same number of samples; the control variate version them are also included

other multi-assets options which are common in energy market valuation problems.

Bibliography

[1] J. Anderluh, J. Weide, *Double-sided Parisian option pricing*, Finance Stoch. 13(2) 205–238, (2009).

[2] L. Andersen, J. Andreasen, *Jump-diffusion processes: volatility smile fitting and numerical methods for option pricing*, Review of Derivatives Research, 4 231–262, (2000).

[3] D. Applebaum, *Lévy Processes and Stochastic Calculus*, Cambridge studies in advanced mathematics 93, Cambridge University Press (2004).

[4] E. Benhamou, *Fast Fourier transform for discrete Asian options*, Journal of Computational Finance, 6(1) 49–68, (2002).

[5] C. Canuto and M.Y. Hussaini and A. Quarteroni and T.A. Zang, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*, Springer (2007).

[6] P. Carr, D. Madan *Option valuation using the fast Fourier transform*, Journal of Computational Finance, (2) 61–73, (1999).

[7] M. Dempster, S. Hong, *Spread option valuation and the fast Fourier transform*, Technical Report WP 26, (2000).

[8] S. Deng, *Pricing electricity derivatives under alternative stochastic spot price*

models, Proceedings of the 33rd Hawaii International Conference on System Sciences, (2000).

[9] M. Egozcue, L. Garcia, and W. Wong, *On Some Covariance Inequalities For Monotonic And Non-Monotonic Functions*, Journal of Inequal. of Pure and Applied Math., 3(10), (2009).

[10] J. Esary, F. Proschan, and D. Walkup, *Association of random variables, with applications*, Annals of Mathematical Statistics 38:1466-1474, (1967).

[11] F. Fang, C. Oosterlee, *A novel pricing method for european options based on Fourier-cosine series expansions*, SIAM Journal on Scientific Computing, 31(2)(4) 826–848, (2008).

[12] G. Fusai, *Pricing Asian options via Fourier and Laplace transforms*, Journal of Computational Finance, 7 87–106.(2004).

[13] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, 1st edition, Springer (2003).

[14] G. Grimmett, D. Stirzaker, *Probability and Random Processes*, 2nd Edition, Clarendon Press, Oxford, (1992).

[15] S. Heston, *A closed form solution for options with stochastic volatility with application to bond and currency options*, Review with Financial Studies 6 (2) 327C343, (1993).

[16] G. Loeper, O. Pironneau, *A mixed PDE/Monte-Carlo method for stochastic volatility models*, C. R. Acad. Sci. Paris, Ser. I 347 559C563, (2009).

[17] R. Lord et al, *A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes*, SIAM Journal on Scientific Computing, 30 1678-1705, (2008).

[18] M. Milevsky, S. Posner, *A closed-form approximation for valuing basket options*, Journal of Derivatives, 5(4) 54-61, (1998).

[19] S. Raible, *Lévy processes in finance: Theory, numerics and emperical facts*, PhD Thesis, Albert-Ludwigs-Univ. Freiburg, (2000).

[20] W. Rudin, *Real and Complex Analysis*, New York, McGraw-Hill, (1986).

[21] S. Shreve, *Stochastic Calculus for Finance II Continuous-Time Models*, Springer (2004).

Appendix A

Smoothness of $g(z_1)$

Let $g(z_1)$ be the function defined in (3.3). The purpose of this appendix is to show that $g(z_1)$ is a smooth function.

A.1 A physics proof

Let $f(z_1, \bar{z}) = (h(z_1, \bar{z}))_+$ where

$$h(z_1, \bar{z}) = \sum_{i=1}^{7} \alpha_i \exp((AZ)_i) - K$$

is an infinitely differentiable function of z_1 and $\bar{z}$. We first write

$$g(z_1) = E[h(z_1, \bar{z}) 1_{h>0} | z_1] = \int_0^\infty h\phi(z_1, h) dh,$$

where $\varphi(\bar{z})$ is the density function of $\bar{z}$ and $\phi(z_1, h)$ is the density function of $h(z_1, \bar{z})$ given z_1.

If we can differentiate under the integral sign, we only need to prove that $\phi(z_1, h)$ is differentiable with respect to z_1.

To see that we first define $h_i = \alpha_i e^{(AZ)_i}$ for $i = 1, \ldots, 7$, and let $(a_1, a_2, \ldots, a_7)^t$ be the first column of A, and let $\bar{A}$ be the 6×6 sub-matrix of A formed by removing the first row and column. Then consider the random variable transformation $\bar{z} \mapsto H =$

$(h, h_3, \ldots, h_7)$. Note that since A is lower triangular, $h_1 = \alpha_1 e^{a_1 z_1}$. Considering z_1 as a parameter, one can show that this map is one-to-one with inverse

$$\bar{z} = \bar{A}^{-1} \begin{bmatrix} \ln \dfrac{h_2}{\alpha_2 e^{a_2 z_1}} \\ \ln \dfrac{h_3}{\alpha_3 e^{a_3 z_1}} \\ \vdots \\ \ln \dfrac{h_7}{\alpha_7 e^{a_7 z_1}} \end{bmatrix}$$

where $h_2 = h - h_1 - \sum_{i=3}^{7} h_i + K$.

The domain of the map $\bar{z} \mapsto H$ is $\mathbb{R}^6$ hence the range of the map is

$$R_H = \{H \in \mathbb{R}^6 | h - \alpha_1 e^{a_1 z_1} - \sum_{i=3}^{7} h_i + K > 0,\ h_i > 0,\ i = 3, \ldots, 7\}.$$

One can check that the Jacobian matrix $\dfrac{\partial \bar{z}}{\partial H}$ is

$$\bar{A}^{-1} \begin{bmatrix} \dfrac{1}{h_2} & -\dfrac{1}{h_2} & -\dfrac{1}{h_2} & \cdots & -\dfrac{1}{h_2} \\ & \dfrac{1}{h_3} & & 0 & \\ & 0 & \ddots & & \\ & & & & \dfrac{1}{h_7} \end{bmatrix}$$

so the Jacobian determinant is

$$\left| \frac{\partial \bar{z}}{\partial H} \right| = \frac{1}{|\bar{A}| \prod_{i=2}^{7} h_i} \neq 0.$$

Therefore the probability density function of H is

$$p(z_1, H) = (2\pi^{-3}) \prod_{i=2}^{7} \exp\{-\frac{z_i^2(H)}{2}\} |\bar{A}|^{-1} \left(\prod_{i=2}^{7} h_i \right)^{-1}. \tag{A.1}$$

Now let

$$R(z_1, h) = \{(h_3, \ldots, h_7) \in \mathbb{R}^5 | h - \alpha_1 e^{a_1 z_1} - \sum_{i=3}^{7} h_i + K > 0,\ h_i > 0,\ i = 3, \ldots, 7\},$$

then we have that

$$\phi(z_1, h) = \int_{R(z_1,h)} p(z_1, H)\ dh_3 dh_4 \cdots dh_7. \tag{A.2}$$

To show that $\phi(z_1, h)$ is differentiable with respect to z_1, we have to notice that $\phi(z_1, h)$ depends on z_1 in two ways: the integral range and the integrand.

Observe that $R(z_1, h)$ is a bounded subset of $\mathbb{R}^5$ for every fixed z_1 and h, and strictly decreases when z_1 increases. This is clear by writing $R(z_1, h)$ as

$$\{(h_3, \ldots, h_7) \in \mathbb{R}^5 |\ \sum_{i=3}^{7} h_i < h - \alpha_1 e^{a_1 z_1} + K,\ h_i > 0,\ i = 3, \ldots, 7\}. \tag{A.3}$$

The function $p(z_1, h, \bar{h})$ can be extended continuously to the closure of $R(z_1, h)$ by defining its value as zero on the boundary. This can be seen by looking at the structure of $p(z_1, h, \bar{h})$ in Equation (A.1). For instance, if $h_3 \to 0$, in $p(z_1, h, \bar{h})$ there is a term of the form

$$\frac{\exp\{-a \ln^2 h_3 + b \ln h_3\}}{h_3} = h_3^{-a \ln h_3 + b - 1} \to 0,\ \text{if}\ \ h_3 \to 0$$

and all other terms approach 0 as $h_3 \to 0$.

The same extension can be done to the function $\dfrac{\partial p(z_1, h, \bar{h})}{\partial z_1}$.

To show $\dfrac{\partial \phi}{\partial z_1}$ exists, we consider a neighborhood $[z_1 - \delta, z_1 + \delta]$ and let $E = R(z_1 - \delta, h)$, and $\bar{h} = (h_3, \ldots, h_7)$ then $R(z, h) \subseteq R(z_1 - \delta, h)$ for all $z_1 \in [z_1 - \delta, z_1 + \delta]$. Define $R_\epsilon = R(z_1 + \epsilon, h)$ and $R = R(z_1, h)$, then by definition

$$\begin{aligned}
\frac{\partial \phi}{\partial z_1} &= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_E \left[p(z_1 + \epsilon, h, \bar{h}) 1_{R_\epsilon} - p(z_1, h, \bar{h}) 1_R \right] d\bar{h} \\
&= \lim_{\epsilon \to 0} \int_E \left[\frac{(p(z_1 + \epsilon, h, \bar{h}) - p(z_1, h, \bar{h})) 1_{R_\epsilon}}{\epsilon} - \frac{p(z_1, h, \bar{h}) 1_{R \setminus R_\epsilon}}{\epsilon} \right] d\bar{h} \\
&= \lim_{\epsilon \to 0} \int_E \frac{[p(z_1 + \epsilon, h, \bar{h}) - p(z_1, h, \bar{h})] 1_{R_\epsilon}}{\epsilon} d\bar{h} - \lim_{\epsilon \to 0} \int_E \frac{p(z_1, h, \bar{h}) 1_{R \setminus R_\epsilon}}{\epsilon} d\bar{h}.
\end{aligned}$$

We then show the two limits in the last equation exist. The first one is

$$\lim_{\epsilon \to 0} \int_E \frac{[p(z_1 + \epsilon, h, \bar{h}) - p(z_1, h, \bar{h})] 1_{R_\epsilon}}{\epsilon} d\bar{h}. \tag{A.4}$$

We can use the dominated convergence theorem to show that it converges to

$$\int_R \frac{\partial p}{\partial z_1} d\bar{h}.$$

First of all the integrands converge point-wise to the function $\frac{\partial p}{\partial z_1} 1_R$. Also by the mean value theorem we have

$$\frac{p(z_1+\epsilon, h, \bar{h}) - p(z_1, h, \bar{h})}{\epsilon} = \left.\frac{\partial p}{\partial z_1}\right|_{z_1=z^*}$$

for some $z^* \in [z_1, z_1+\epsilon]$. Now by compactness $\frac{\partial p}{\partial z_1}$ is bounded on $[z_1-\delta, z_1+\delta] \times \bar{E}$, where $\bar{E}$ is the closure of E, therefore the integrands are uniformly bounded by an integrable function, hence the dominated convergence theorem is applicable.

For the second limit, we observe that as $\epsilon \to 0$, the limit only depends on the boundary of R. It can be elaborated as follows.

To better describe the movement of the $R(z_1, h)$ as z_1 changes, consider it as a motion in $\mathbb{R}^5$ with z_1 as the time. At time z_1, the object (region) is at position $R(z_1, h)$. The initial position is $R(0, h)$. Define function

$$\bar{h} = P(z_1, h, x) = \frac{h - \alpha_1 e^{a_1 z_1} + K}{h - \alpha_1 + K} x, \ x \in \overline{R(0, h)},$$

then $\bar{h} = P_{z_1} = P(z_1, h, x)$ is the position of the point that is initially at x at time z_1.

Figure A.1 shows the movement of the boundary from time z_1 to $z_1 + \epsilon$.

As shown in figure A.1, when ϵ is small, the integral $\int_E \frac{p(z_1, h, \bar{h}) 1_{R \setminus R_\epsilon}}{\epsilon}$ can be approximated by

$$\int_{\partial R} p(z_1, h, \bar{h}) \left| \frac{P_{z_1+\epsilon} - P_{z_1}}{\epsilon} \cdot \vec{n} \right| dA$$

where $\vec{n}$ is the outward-pointing unit normal vector of the boundary.

Now since $\frac{P_{z_1+\epsilon} - P_{z_1}}{\epsilon}$ has limit

$$\frac{\partial P(z_1, h, x)}{\partial z_1} = \frac{-\alpha_1 a_1 e^{a_1 z_1}}{h - \alpha_1 + K} x = \frac{-\alpha_1 a_1 e^{a_1 z_1}}{h - \alpha_1 e^{a_1 z_1} + K} \bar{h},$$

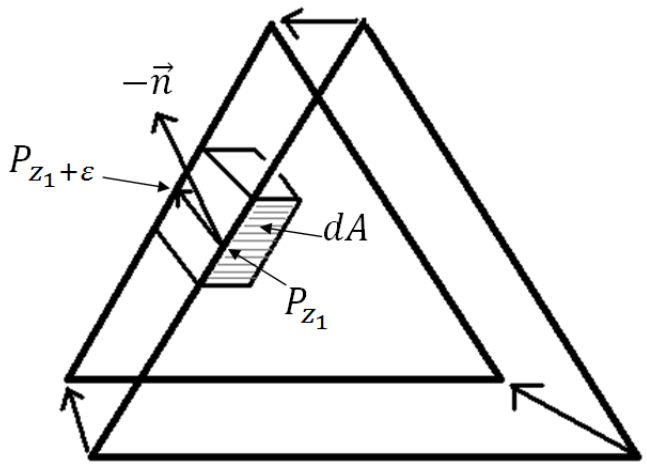

Figure A.1: Illustration for motion near the boundary, not the actual figure which is in $\mathbb{R}^5$

which can be considered as the instantaneous velocity at time z_1 at position $\bar{h}$, and is continuous on z_1 and $\bar{h}$.

Thus we have

$$\lim_{\epsilon \to 0} \int_E \frac{p(z_1, h, \bar{h}) 1_{R \setminus R_\epsilon}}{\epsilon} d\bar{h} = \int_{\partial R} p(z_1, h, \bar{h}) \frac{\alpha_1 a_1 e^{a_1 z_1}}{h - \alpha_1 e^{a_1 z_1} + K} \bar{h} \cdot \vec{n} dA.$$

Hence

$$\frac{\partial \phi}{\partial z_1} = \int_R \frac{\partial p}{\partial z_1} d\bar{h} - \int_{\partial R} p(z_1, h, \bar{h}) \frac{\alpha_1 a_1 e^{a_1 z_1}}{h - \alpha_1 e^{a_1 z_1} + K} \bar{h} \cdot \vec{n} dA. \tag{A.5}$$

Therefore we have shown that $g(z_1)$ is differentiable. The whole idea can be reapplied to show that $\phi(z_1, h)$ is n-th order differentiable with respect to z_1.

If we look at equation A.5 and the region R defined in (A.3) more closely, the boundary of R consists of 6 "faces" in which 5 of them is of the form

$$F_k = \{(x_1, \ldots, x_5) \in \mathbb{R}^5 | \sum_{i=1}^{5} x_i \leq h - \alpha_1 e^{a_1 z_1} + K,\ x_k = 0,\ x_i \geq 0,\ i = 1, \ldots, 5\},$$

and the other one is region S defined by

$$S = \{(x_1, \ldots, x_5) \in \mathbb{R}^5 | \sum_{i=1}^{5} x_i = h - \alpha_1 e^{a_1 z_1} + K,\ \ x_i \geq 0,\ i = 1, \ldots, 5\}.$$

The unit normal vector for F_k is the natural basis vector e_k, which is orthogonal to any $\bar{h} \in F_k$. But the unit normal vector of S is apparently the vector $\vec{u} = (1,1,1,1,1)/\sqrt{5}$, which is in general not orthogonal to any vector $\bar{h} \in S$ since $\bar{h} \cdot \vec{u} = \frac{1}{\sqrt{5}}(h - \alpha_1 e^{a_1 z_1} + K)$ if $\bar{h} \in S$.

Therefore the second term in Equation (A.5) can be rewritten as

$$\int_S p(z_1, h, \bar{h}) \alpha_1 a_1 e^{a_1 z_1} dA,$$

so

$$\frac{\partial \phi}{\partial z_1} = \int_R \frac{\partial p}{\partial z_1} d\bar{h} - \frac{1}{\sqrt{5}} \int_S p(z_1, h, \bar{h}) \alpha_1 a_1 e^{a_1 z_1} dA.$$

A.2 Another approach using divergence theorem

Equation (A.5) can be verified using a different approach which utilizes the divergence theorem. First let's review the divergence theorem.

Theorem A.2.1 *Suppose V is a subset of $\mathbb{R}^n$ which is compact and has a piecewise smooth boundary ∂V. If F is a continuously differentiable vector field defined on a neighborhood of V, then we have*

$$\iiint_V \nabla \cdot \mathbf{F} \; dV = \oiint_{\partial V} F \cdot \vec{n} \; dA,$$

where $\vec{n}$ is the outward-pointing unit normal vector field of the boundary ∂V.

Let $\lambda = h - \alpha_1 e^{a_1 z_1} + K$, then the derivative of λ with respect to z_1 is $\lambda' = -\alpha_1 a_1 e^{a_1 z_1}$. In Equation (A.2) we can do a change of variable by $\bar{h} = \lambda x$, then the Jacobian determinant is $\left|\frac{\partial \bar{h}}{\partial x}\right| = \lambda^5$. After the change of variable we can see the integration region becomes

$$\Delta = \{(x_1, \ldots, x_5) \in \mathbb{R}^5 | \sum_{i=1}^{5} x_i < 1, \quad x_i > 0, \; i = 1, \ldots, 5\},$$

so we have

$$\phi(z_1, h) = \int_\Delta p(z_1, h, \lambda x)\lambda^5 dx.$$

Now since Δ is bounded and the integrand is smooth on $\overline{\Delta}$ with respect to z_1, we can differentiate under the integral sign and get

$$\begin{aligned}
\frac{\partial \phi}{\partial z_1} &= \int_\Delta p(z_1, h, \lambda x) 5\lambda^4 \lambda' dx + \int_\Delta \frac{\partial p(z_1, h, \lambda x)}{z_1} \lambda^5 dx \\
&= \int_\Delta p(z_1, h, \lambda x) 5\lambda^4 \lambda' dx + \int_\Delta \left[\left.\frac{\partial p}{\partial z_1}\right|_{(z_1, h, \lambda x)} + \left.\frac{\partial p}{\partial \bar{h}}\right|_{(z_1, h, \lambda x)} \cdot \lambda' x \right] \lambda^5 dx \\
&= \frac{\lambda'}{\lambda} \int_R \left[5p(z_1, h, \bar{h}) + \frac{\partial p}{\partial \bar{h}} \cdot \bar{h} \right] d\bar{h} + \int_R \frac{\partial p}{\partial z_1} d\bar{h} \\
&= \int_R \frac{\partial p}{\partial z_1} d\bar{h} + \frac{\lambda'}{\lambda} \int_R \nabla \cdot \left(p(z_1, h, \bar{h}) \bar{h} \right) d\bar{h} \\
&= \int_R \frac{\partial p}{\partial z_1} d\bar{h} + \frac{\lambda'}{\lambda} \int_{\partial R} p(z_1, h, \bar{h}) \bar{h} \cdot \vec{n} dA.
\end{aligned}$$

The divergence theorem is used in last equality. Therefore we have

$$\frac{\partial \phi}{\partial z_1} = \int_R \frac{\partial p}{\partial z_1} d\bar{h} + \frac{\lambda'}{\lambda} \int_{\partial R} p(z_1, h, \bar{h}) \bar{h} \cdot \vec{n} dA,$$

which is the same as Equation (A.5).