

**Pricing Path-Dependent Derivative Securities Using Monte Carlo  
Simulation & Intra-Market Statistical Trading Model**

A Thesis  
Presented to  
The Academic Faculty

by

**Sungjoo Lee**

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy

School of Industrial and Systems Engineering  
Georgia Institute of Technology  
November, 2004

# **Pricing Path-Dependent Derivative Securities Using Monte Carlo Simulation & Intra-Market Statistical Trading Model**

Approved by:

Dr. David Goldsman, Committee Chair  
School of Industrial & Systems Engineering

Dr. Carl Spruill  
School of Mathematics

Dr. Shijie Deng, Advisor  
School of Industrial & Systems Engineering

Dr. Keebom Kang  
United States Navy PostGraduate School

Dr. Christos Alexopoulos  
School of Industrial & Systems Engineering

Date Approved: November 17, 2004

*To beloved my parents,  
my brothers and sisters,  
and my fellows in Atlanta.*

## ACKNOWLEDGEMENTS

First, I am very grateful to my academic advisor Professor David Goldsman for his permanent interest as well as financial support through the interesting research projects while I complete my thesis. His valuable comments, helpful suggestions, constant guidance, and friendship are gratefully acknowledged.

I also want to thank my co-advisor Professor Shijie Deng for his deep and sharp intuition. His advice made my work more effective and efficient.

I would like to thank to Professor Christos Alexopoulos, Professor Carl Spruill, Professor Robert Kertz, and Professor Keebom Kang for their interests, kindness and valuable comments during my studies. I also wish to express my thanks to Professor Gary Parker for financial support and encouragement during my Ph.D. program.

Finally, I would like to thank my parents, brothers, and sisters. I could not possibly finish this dissertation without them.

# TABLE OF CONTENTS

<b>DEDICATION</b> . . . . .	<b>iii</b>
<b>ACKNOWLEDGEMENTS</b> . . . . .	<b>iv</b>
<b>LIST OF TABLES</b> . . . . .	<b>vii</b>
<b>LIST OF FIGURES</b> . . . . .	<b>ix</b>
<b>SUMMARY</b> . . . . .	<b>xi</b>
<b>I INTRODUCTION</b> . . . . .	<b>1</b>
<b>PART I : PRICING PATH DEPENDENT DERIVATIVE SECURITIES</b> . . . . .	<b>5</b>
<b>II NUMERICAL METHODS</b> . . . . .	<b>6</b>
2.1 Black-Scholes Equation . . . . .	6
2.2 Lattice methods . . . . .	10
2.2.1 Binomial Tree . . . . .	11
2.2.2 Trinomial Method . . . . .	18
2.3 Finite-Difference Method . . . . .	20
2.3.1 Explicit Method . . . . .	21
2.3.2 Implicit Method . . . . .	25
<b>III MONTE CARLO SIMULATION</b> . . . . .	<b>29</b>
3.1 Boundary with Regression . . . . .	32
3.1.1 Implementation and Numerical Results . . . . .	34
3.2 An Adaptive Simulation Algorithm . . . . .	39
3.2.1 High and Low Biased Estimators . . . . .	45
3.2.2 Convergence of the Optimal Exercising Policy . . . . .	48
3.2.3 Implementation and Numerical Examples . . . . .	53
3.2.4 Conclusion . . . . .	66
3.3 Variance Reduction . . . . .	67
3.3.1 Antithetic Variates (AV) . . . . .	67
3.3.2 Control Variates (CV) . . . . .	69
3.3.3 Importance Sampling (IS) . . . . .	73
3.3.4 Stratified Sampling . . . . .	74

<b>IV INTRODUCTION</b> . . . . .	<b>78</b>
<b>PART II : INTRA-MARKET STATISTICAL TRADING MODEL</b> . . . . .	<b>80</b>
<b>V STATISTICAL TRADING MODEL</b> . . . . .	<b>81</b>
5.1 Modeling . . . . .	81
5.1.1 Numerical Example . . . . .	86
5.2 SPC with Autocorrelated Data . . . . .	90
5.3 Variance Estimation . . . . .	96
5.3.1 Batch-Means Estimator . . . . .	97
5.3.2 Standardized Time Series Weighted Area Estimator (STS) . . . . .	99
5.3.3 Cramér-von Mises Variance Estimator . . . . .	102
5.4 Conclusion . . . . .	105
<b>VI APPENDIX</b> . . . . .	<b>107</b>
<b>REFERENCES</b> . . . . .	<b>112</b>
<b>VITA</b> . . . . .	<b>116</b>

## LIST OF TABLES

Table 1	Price estimate with IEB and OEB. The optimized coefficients are only for a linear function. Option parameters: $K = 100$ , $r = 0.05$ , $\delta = 0.10$ , $T = 1.0$ , and $\sigma = 0.2$ with $\Delta t = 0.25\text{yr}$ and 1M replications. Error unit is %.	35
Table 2	Price estimate comparison with OEB and the stochastic tree method of Glasserman and Broadie. The optimized coefficients are only for a linear function. Option parameters: $K = 100$ , $r = 0.05$ , $\delta = 0.10$ , $T = 1.0\text{yr}$ , and $\sigma = 0.2$ . TRUE(E) represents the true value of European options.	36
Table 3	Price estimates with IEB and OEB. The optimized coefficients are only for a linear function (L-O) and quadratic function (Q-O). Option parameters: $S_0 = 70$ , $K = 100$ , $r = 0.05$ , $\delta = 0.10$ , $T = 1.0\text{yr}$ , and $\sigma = 0.2$ .	36
Table 4	Price estimation with optimized two-piecewise linear early exercise boundary. The optimized coefficients are only for a linear function. Parameters are $K = 100$ , $T = 0.5$ , $r = 0.06$ , and $\sigma_1 = \sigma_2 = 0.6$ .	38
Table 5	Simulation prices of high biased and low biased standard American puts. Option parameters: $K = 100$ , $r = 0.06$ , $T = 0.5$ , and $\sigma = 0.4$ .	55
Table 6	Simulation prices on min-puts on two assets with the same volatility by changing from 3D to 2D. Option parameters: $K = 100$ , $r = 0.06$ , $T = 0.5$ , and $\sigma_1 = \sigma_2 = 0.6$ .	56
Table 7	Simulation prices on min-put options on two assets with different volatilities by reducing from 3D to 2D. Option parameters: $K = 100$ , $r = 0.06$ , $T = 0.5$ , and $\sigma_1 = 0.4$ , $\sigma_2 = 0.8$ .	57
Table 8	Simulation prices of a max call option on 5 assets: $K = 100$ , $r = 0.05$ , $\delta = 0.1$ months, $T = 3$ yrs, and $\sigma_i = 0.2$ . The option can be exercised at any of times $t = iT/d$ , $i = 0, 1, \dots, d$ , where $d = 3, 6, 9$ . CPU-I is for computing the early exercise boundary, CPU-II is for the low and high biased estimator, and CPU-III is quoted from Rogers. CPU-I and CPU-II are computed by Matlab with a P4-2.4Ghz machine and CPU-III is by Scilab with a 600Mhz PC. For Rogers' computation, 1000 sample paths were used for the optimization step and 8000 paths to refine the estimate. All CPU units are seconds. Rogers' CPU times are included reference purposes only.	59
Table 9	Simulation prices of American-Bermudan-Asian option: $K = 100$ , $r = 0.06$ , $t^* = \delta = 3$ months, $T = 2$ years, and $\sigma = 0.2$ . There are 200 exercise opportunities.	63
Table 10	Simulation prices of high biased and low biased standard American puts with jump-diffusion process. All computations are done on a P4 2.4Ghz PC with Matlab. CPU time units are in seconds. Option parameters: $K = 100$ , $r = 0.06$ , and $T = 1$ .	66

Table 11	Simulation prices of low biased standard American puts with Antithetic Variates under GBM. Option parameters: $K = 100$ , $r = 0.06$ , $\sigma = 0.4$ , and $T = 0.5$ . None of the variance reduction techniques are used in obtaining the optimal early exercising policy. . . . .	70
Table 12	Simulation prices of low biased standard American puts with Control Variates under GBM. Option parameters: $K = 100$ , $r = 0.06$ , $\sigma = 0.4$ , and $T = 0.5$ . AV is used to find the optimal early exercising policy. . . . .	70
Table 13	Trading summary based on Figure 12. D and U correspond to the square boxes on the graph. D is the first day of out of the lower threshold and U is the first day of out of the upper threshold, prices are daily closing prices, and the numbers of shares are rounded up to be integer. The numbers of shares are calculated by assuming that we have \$100,000. . . . .	87
Table 14	Trading summary based on Figure 13. D* and U* correspond to square boxes on the graph. D* is the first day out of the lower threshold and U* is the first day out of the upper threshold, D and U correspond to solid circles on the graph. D is the first day out of the lower bound and U is the first day returning from the upper bound. Prices are daily closing prices, and the numbers of shares are rounded up to an integer. The numbers of shares are calculated by assuming that we have \$100,000 for each trading method. . . . .	88
Table 15	Trading summary based on Figure 12. D* and U* correspond to square boxes on the graph. D* is the first day out of the lower threshold and U* is the first day out of the upper threshold, D and U correspond to solid circles on the graph. D is the first day out of the lower bound and U is the first day returning from the upper bound. Prices are daily closing prices, and the numbers of shares are rounded up to an integer. The numbers of shares are calculated by assuming that we have \$100,000 for each trading method. . . . .	89
Table 16	Trading summary on the five different pairs. The first 50 unweighted batch means of data are used to compute the mean and thresholds. Numbers of shares are computed via \$100000/share price. One-minute tick data was collected from 4/28/2004 to 05/26/2004. A batch size of 20 is used for each batch. . . . .	96
Table 17	CvM variance estimators with different numbers of batches. Underlying stochastic process is AR(1) (autoregressive process) with $\phi = 0.9$ and batch size 1024 is used. Frequencies are obtained with 1000000 replications. We use weighting functions from ([47]). . . . .	104
Table 18	Simulation prices on min-put options on two assets with the same volatility by $\min(S_1, S_2)$ and the difference of $S_1$ and $S_2$ . Option parameters: $K = 100$ , $r = 0.06$ , $T = 0.5$ , and $\sigma_1 = \sigma_2 = 0.6$ . . . . .	111
Table 19	Simulation prices on min-put options on two assets with different volatilities by $\min(S_1, S_2)$ and the difference of $S_1$ and $S_2$ . Option parameters: $K = 100$ , $r = 0.06$ , $T = 0.5$ , and $\sigma_1 = 0.4$ , $\sigma_2 = 0.8$ . . . . .	111

## LIST OF FIGURES

Figure 1	One-step binomial tree. . . . .	13
Figure 2	Trinomial tree with one time step from $t$ to $t + \Delta t$ . . . . .	19
Figure 3	Searching for optimal boundary. . . . .	34
Figure 4	Distribution of the boundary for each time step. As we have more iterations, the shape of the distribution becomes narrower, and it converges to the optimal boundary. . . . .	44
Figure 5	Standard put price convergence by updating the free boundary with different initial stock prices. The $X$ -axis represents the iteration number and the $Y$ -axis represents option prices. Parameters are: $K = 100, r = 0.05, T = 0.5$ , and $\sigma = 0.4$ . . . . .	54
Figure 6	Approximated early exercise boundary for American-Bermudan-Asian option at time $t$ . . . . .	60
Figure 7	Approximated boundary for American-Bermudan-Asian option. . . . .	62
Figure 8	Price estimation with three different methods. The dotted line is the true price obtained by the finite-difference method, the dashed asterisk is without any variance reduction technique, the dashed plus sign is with AV, and the dashed point is with CV. . . . .	68
Figure 9	Simulated tree for $b = 3$ . . . . .	75
Figure 10	Stock tree with strata, $N^{Str} = 5, n = 2, T = 2$ . . . . .	76
Figure 11	A sample path of a price difference process. Days 3 and 7 are out of bounds. We need to take an action on these events. . . . .	84
Figure 12	KO and PBG's difference moving average, MA(10), and upper and lower thresholds. Date 1 is August 2, 2002 and Date 260 is July 5, 2001. The solid square boxes are the points out of the bounds where we may need to take a proper action. Table 13 shows overall gains and losses by trading corresponding to this graph. . . . .	86
Figure 13	KO and PBG's difference moving average, MA(10), and upper and lower thresholds. Date 1 is August 2, 2002 and Date 260 is July 5, 2001. In Figure 12, we just had the solid boxes; the current graph includes solid circles to indicate the new trading dates. . . . .	87
Figure 14	Autocorrelation plot for the KO-PBG difference process. Moving average length is 5 minutes. . . . .	92
Figure 15	Normality plot of KO-PBG difference process. Moving average length is 5 minutes. . . . .	92
Figure 16	Autocorrelation plot for batch mean difference process for KO and PBG. Moving average length is 5 minutes and batch size is 60 minutes. . . . .	94

Figure 17	Normality plot for batch mean difference process for KO and PBG. Moving average length is 5 minutes and batch size is 60 minutes. . . . .	94
Figure 18	Percent change of difference process. Mean and standard deviation is calculated with the first 50 data points and trading strategy is applied after that. Overall profit was \$1330. . . . .	95

## SUMMARY

This thesis is composed of two parts. The first part deals with a technique for pricing American-style contingent options. The second part details a statistical arbitrage model using statistical process control approaches.

We propose a novel simulation approach for pricing American-style contingent claims. We develop an adaptive policy search algorithm for obtaining the optimal policy in exercising an American-style option. The option price is first obtained by estimating the optimal option exercising policy and then evaluating the option with the estimated policy through simulation. Both high-biased and low-biased estimators of the option price are obtained. We show that the proposed algorithm leads to convergence to the true optimal policy with probability one. This policy search algorithm requires little knowledge about the structure of the optimal policy and can be implemented naturally using parallel computing methods. As illustrative examples, computational results on pricing regular American options and American-Asian options are reported and they indicate that our algorithm is faster than certain alternative American option pricing algorithms reported in the literature.

Secondly, we investigate arbitrage opportunities arising from continuous monitoring of the price difference of highly correlated assets. By differentiating between two assets, we can separate common macroeconomic factors that influence the asset price movements from an idiosyncratic condition that can be monitored very closely by itself. Since price movements are in line with macroeconomic conditions such as interest rates and economic cycles, we can easily see abnormal behavior on the price changes. We apply a statistical process control approach for monitoring time series with serially correlated data. We use various variance estimators to set up and establish trading strategy thresholds.

# CHAPTER I

## INTRODUCTION

One of the most important problems in option pricing theory is the valuation and optimal exercise of derivatives with American-style exercise features. There have been an increasing number of important security pricing models where analytical solutions are not available. These types of derivatives are traded in all major financial markets (equity, commodity, foreign exchange, insurance, energy, mortgage, swap, municipal, and real estate).

Up to now, the primary methods for pricing American-style options are trees and finite-difference methods to solve partial differential equations (PDEs) with associated boundary values. Broadie and Detemple [7] provide a recent comparison of various existing methods for pricing standard American call and put options written on a single underlying dividend paying asset. In general, the computational speed of these methods is significantly better than that of simulation methods for simple models and contracts. However, due to its flexibility, Monte Carlo simulation has been a strong alternative to price more-complex options.

Trees and finite-difference methods can be used to generate numerical solutions to pricing problems with one or two sources of uncertainty [40]. However, even though finite-difference methods are used exclusively in practice, the major drawback of these methods is that they can often handle only one or two sources of uncertainty. People try to reduce the dimensionality of uncertainty to fit into a lattice model and sometimes the finite-difference approach works well. However, higher dimension or stochastic parameter problems must use Monte Carlo simulation ([9], [66]).

Given the importance of valuing early-exercise features in problems with multiple state variables, the dearth of studies that address these problems must be explained by a need for effective valuation procedures. This paper presents a general method for the valuation of assets with early-exercise features. Our algorithm employs random sampling rather than the enumeration implicit in lattice techniques such as Binomial methods; therefore, it can be applied to models with multiple

state variables and possible path dependencies. The major difficulty in valuing early-exercise features is the need to estimate optimal exercise policies. Standard simulation procedures are forward algorithms, i.e., state variable paths that are simulated forward in time. Given a state trajectory and a pre-specified exercise policy, a path price is determined. An average over independent samples of path prices gives an unbiased estimate of the security price. By contrast, pricing procedures for assets with early-exercise features are generally backward algorithms. That is, the optimal exercise strategy at maturity stems from the difficulty of applying an inherently forward-based procedure to a problem that requires a backward procedure to solve.

With those difficulties, Monte Carlo simulation was first introduced to finance by Boyle [6]. In the past decade, a number of Monte Carlo simulation-based approaches have been proposed to address the problem of pricing American-style options with a finite number of exercise opportunities. In general, these approaches try to approximate the value function or early exercise frontiers with combinations of dynamic programming and Monte Carlo simulation. Reducing the dimensionality of the value function methods was suggested by Tilley [64], Barraquand and Martineau [4], Carr and Yang [13], and others. Approximation of the value function was proposed by Carriere [14], Longstaff and Schwartz [45], Haugh and Kogan [38], and others. The algorithms proposed by Broadie and Glasserman [8, 10] are based on simulated paths and lead to biased high and low estimators that converge to the true values in the appropriate limit. Recently, Longstaff and Schwartz [45] and Tsitsiklis and Van Roy [65] applied least squares methods (LSM) to the pricing of American options by approximating the holding value function at each time step using a linear combination of basis functions fitted to the simulated data. In particular, Longstaff and Schwartz demonstrate the efficiency of their least squares approach through several numerical examples, and Tsitsiklis and Van Roy rigorously establish the general convergence properties of the method. Methods based on the parametrization of the early exercise frontier have been proposed by Grant, Vora, and Weeks [35], Andersen [2], Garcia [28], and Ibáñez and Zapatero [41].

Grant et al. [35] address specifically the pricing of American-Asian options. Their procedure mimics the backward induction solution method of stochastic dynamic programming. At every exercisable date, the optimal threshold parameters are estimated by testing all possible values from a preselected finite parameter grid. Wu and Fu [66] set the threshold of the optimal early exercise

boundary to maximize the expected payoff. First, they set the threshold and estimate each parameter using perturbation analysis to maximize its value. They derived the structural properties for the optimal exercise policy. More thorough overviews of the simulation approach can be found in Broadie and Glasserman [9] or Fu et al. [24].

Duality-based approaches were recently developed by Rogers [53], Andersen and Broadie [3], and Haugh and Kogan [38]. They introduced a dual method to price American options based on simulating the path of the option payoff and a judiciously chosen Lagrangian martingale. The main practical contribution of their paper is a general algorithm for constructing upper and lower bounds on the true price of the option using any approximation to the option price. The computation of the lower bound is straightforward and relies on simulating the suboptimal exercise strategy implied by the approximate option price. A method for finding a tight upper bound was introduced in each paper; however, the problem of obtaining the right one seems to be more art than science. This “art” was the drawback of their algorithm.

In this thesis, we propose a new algorithm using *cross entropy* to set up the early exercise frontiers based on Monte Carlo simulation to estimate security prices. This algorithm can be applied to models with multiple state variables and with possible path dependencies in the state variables and nonstandard dynamics such as jumps.

The adaptive simulation algorithm studied in this paper is a variant of the cross-entropy algorithm developed for combinatorial and multi-extremal optimization and rare event simulation ([55], [68]). The essence of the algorithm lies in adaptive policy learning which leads to convergence to the optimal policy. The advantage of the proposed simulation approach is that it is a very generic algorithm which requires little knowledge on the structure of the optimal policy. It can be shown to converge to an optimal policy in the finite-horizon option pricing problem. Also, this algorithm does not require the approximation of the conditional value of the continuation function as in [45] and [65]. Therefore, we do not have to deal with these sources of errors. In addition, we do not need to worry about the choice of basis functions.

The rest of Part I is organized as follows. Chapter 2 contains well-known numerical methods for option pricing, which are extensively used in practice. In Chapter 3, we propose new algorithms to price path-dependent derivative securities and we show how to apply variance reduction techniques.

Statistical arbitrage modeling is introduced in Part II. Part II is organized as follows. Chapter 4 explains the reasoning of our approach. In Chapter 5, we explain how to set up the model and provide numerical examples and various variance estimators to set up and establish the trading strategy thresholds.

**PART I**  
**PRICING PATH DEPENDENT DERIVATIVE**  
**SECURITIES**

## CHAPTER II

### NUMERICAL METHODS

In the mathematical treatment of financial derivatives and especially that of options, the defining stochastic differential equation coupled with the arbitrage-free pricing condition leads to a deterministic partial differential equation. The solution of this equation under appropriate boundary conditions is interpreted as the price of the asset.

There are three main alternative approaches to finding the function that describes the price of the contingent claim. First, we have classical methods involving the solution of partial differential equations. These methods are widely used in physics, but they have yet to take root in the economic arena. Their importance, however, cannot be overlooked and we note in passing that even in the simplest cases, the solution of a partial differential equation is everything but trivial. A second approach can be found in the methods of mathematical statistics, where the stochastic differential equation describing the asset price is solved via the equivalence between the financial no-arbitrage condition and that of a martingale. Both of these approaches often lead to closed-form solutions that are easy to use and convenient for the valuation of assets in real time. Finally, when a solution in closed form is not available or possible, numerical methods may provide an alternative solution. Nevertheless, the application of these methods in finance still presents great practical difficulties, due mainly to the time involved in obtaining a solution.

This chapter deals with the first of the approaches mentioned above. We first describe the Black-Scholes equation for the valuation of European options. Later, we provide a fairly detailed and simple description of a few useful, well-known numerical methods to solve this type of equation.

#### ***2.1 Black-Scholes Equation***

Standard Brownian motion with drift is described by the stochastic differential equation [52]

$$dx = \mu dt + \sigma dW$$

$$x(t_0) = x_0$$

or

$$x(t) = x_0 + \mu \int_{t_0}^t ds + \sigma \int_{t_0}^t dW(s),$$

where  $W(s)$  is Brownian motion on a given probability space  $(\Omega, \mathcal{F}, P)$ ,  $x_0$  is the initial condition,  $\mu$  is a drift, and  $\sigma$  is a volatility.

Since

$$\int_{t_0}^t dW(s) = \lim_{\Delta s \rightarrow 0} \sum_{j=0}^{N-1} [W(s_{j+1}) - W(s_j)] = W(t) - W(t_0)$$

for time increment  $\Delta s = (t - t_0)/N$ , where  $N$  is the size of discretization, we see that the Brownian motion with drift process is given explicitly by

$$x(t) = x_0 + \mu(t - t_0) + \sigma[W(t) - W(t_0)].$$

The model for an equity asset is not the simple Brownian motion with drift but

$$dS = \mu S dt + \sigma S dW.$$

This is equivalent to

$$du = \mu dt + \sigma dW$$

for  $u = \ln S$ , so that the problem is reduced to simple Brownian motion. However,  $S$  is a stochastic variable and not differentiable so that the chain rule can not be applied to conclude that

$$du(t) = \frac{dS(t)}{S(t)}.$$

Instead we need a new tool, called Itô's Lemma, to determine how a function of a stochastic variable varies with changes of the independent variable.

We need the following approximations from the theory of Brownian motion for the increment  $dW$  over the infinitesimal time interval  $dt$ :

$$E(dt dW) = 0$$

$$\text{Var}(dt dW) = o(dt)$$

$$E((dW)^2) = dt$$

$$\text{Var}((dW)^2) = o(dt),$$

where we say that  $f(t) = o(g(t))$  as  $t \rightarrow 0$  if

$$\lim_{t \rightarrow 0} \frac{f(t)}{g(t)} = 0,$$

i.e.,  $f(t)$  goes to zero faster than  $g(t)$  as  $t \rightarrow 0$ . Note that these approximations say that  $(dW)^2 \rightarrow dt$  and  $dt dW \rightarrow 0$  as  $dt \rightarrow 0$  so that in the limit these quantities are no longer stochastic.

Now, Itô's Lemma can be stated.

**Lemma 1** *Let  $X(t)$  be an Itô process satisfying*

$$\begin{aligned} dX(t) &= a(X, t) dt + b(X, t) dW \\ X(0) &= x_0. \end{aligned}$$

*Assume that  $u(t, x)$  is a smooth function of the independent variables  $t$  and  $x$  (i.e.,  $u$  is twice continuously differentiable on  $[0, \infty) \times \mathbf{R}$ ). Then*

$$du(X, t) = \left[ \frac{\partial u}{\partial t} + a(X, t) \frac{\partial u}{\partial x} + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} b^2(X, t) \right] dt + b(X, t) \frac{\partial u}{\partial x} dW.$$

**Proof** The proof of this lemma is based on a Taylor's series expansion.

$$\begin{aligned} du &= \frac{\partial u}{\partial x} dx + \frac{\partial u}{\partial t} dt + \frac{1}{2} \left( \frac{\partial^2 u}{\partial x^2} (dx)^2 + \frac{\partial^2 u}{\partial t^2} (dt)^2 + \frac{\partial^2 u}{\partial t \partial x} dx dt \right) + \dots \\ &= \frac{\partial u}{\partial x} (a dt + b dW) + \frac{\partial u}{\partial t} dt \\ &\quad + \frac{1}{2} \left( \frac{\partial^2 u}{\partial x^2} (a dt + b dW)^2 + \frac{\partial^2 u}{\partial t^2} (dt)^2 + \frac{\partial^2 u}{\partial t \partial x} dt (a dt + b dW) \right) + \dots \\ &= \left[ \frac{\partial u}{\partial t} + a(X, t) \frac{\partial u}{\partial x} + \frac{1}{2} b^2(X, t) \frac{\partial^2 u}{\partial x^2} \right] dt + b(X, t) \frac{\partial u}{\partial x} dW + O(dt)^{3/2}. \end{aligned}$$

This completes the proof. ■

If we apply Itô's Lemma to  $u = \ln S$ , where

$$dS = \mu S dt + \sigma S dW, \tag{1}$$

then (with  $X = S$ ,  $a(S, t) = \mu S$ ,  $b(S, t) = \sigma S$ ) we find

$$du(t) = (\mu - \sigma^2/2) dt + \sigma dW$$

so that

$$S(t) = S(t_0) \exp \left\{ \left( \mu - \frac{\sigma^2}{2} \right) (t - t_0) + \sigma (W(t) - W(t_0)) \right\}. \quad (2)$$

Such a process is called *geometric Brownian Motion*.

While we shall need this representation of  $S(t)$  later on in connection with the binomial method, we are interested in deriving the Black–Scholes equation for the value of an equity option in which the underlying asset satisfies the above equations.

Let  $V$  be the value of a put or call written on a underlying asset with value  $S(t)$  at time  $t$ . We assume that  $V$  depends differentially on the two independent variables  $S(t)$  and  $t$ . Then by Itô's Lemma,  $V$  changes over infinitesimal time interval  $dt$  according to

$$dV = \left( \frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} \right) dt + \sigma S \frac{\partial V}{\partial S} dW.$$

Consider a portfolio consisting of one option of value  $V$  and  $\Delta$  shares of the underlying, where  $\Delta$  is yet undetermined, with  $\Delta > 0$  for shares held long and  $\Delta < 0$  for shares held short. The value of the portfolio at any time  $t$  is

$$\pi = V(S, t) + \Delta S. \quad (3)$$

Over the time interval  $dt$ , the gain in the value of the portfolio is

$$d\pi = dV(S, t) + \Delta dS,$$

i.e.,

$$d\pi = \left( \frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} \right) dt + \sigma S \frac{\partial V}{\partial S} dW + \Delta (\mu S dt + \sigma S dW). \quad (4)$$

We now observe that if  $\Delta = -\frac{\partial V}{\partial S}$ , then the stochastic terms (i.e., the Brownian motion part) cancel so that the gain is deterministic. Of course, if the gain in the value of  $\pi$  is deterministic, then it cannot be more than or less than the gain in the value of the portfolio were it to have been invested at the risk-free interest rate  $r$  — else an arbitrage opportunity would occur. Thus, it also follows from Equation (3) and  $\Delta = -\frac{\partial V}{\partial S}$  that

$$d\pi = r\pi dt = r(V + \Delta S) dt = r \left( V - \frac{\partial V}{\partial S} S \right) dt. \quad (5)$$

Equating the two expressions from Equations (4) and (5) for  $d\pi$  with  $\Delta = -\frac{\partial V}{\partial S}$ , we have that

$$\frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV + \frac{\partial V}{\partial t} = 0.$$

This is the *Black–Scholes equation* for the value of an option.

There are a few assumptions necessary to derive the Black-Scholes equation:

1. The value of the asset can be described by the equation for geometric Brownian motion.
2. Options and shares can be bought and sold at any time since  $\Delta$  changes smoothly with time.
3.  $\partial V/\partial S$  is a continuous smooth function of  $S$ . Therefore, the number of shares in  $\pi$  is allowed to vary continuously with  $S$ , which means that fractional shares can be traded.
4. The change in value of the portfolio is due solely to the changes of  $V$  and  $S$  and does not include transaction costs or the spread between selling and buying prices for options and assets.
5. There are no transaction costs when options and assets are bought and sold.

## ***2.2 Lattice methods***

There are many products (options) which do not have closed-form solutions. Therefore, we need to implement numerical approximation procedures to price them. We introduce two well-known methods in this section.

First, the binomial model has proved over time to be the most flexible, intuitive and popular approach to option pricing. It is based on the simplification that over a single period (of possibly very short duration), the underlying asset can only move from its current price to two possible levels. Binomial trees can also be used to determine the sensitivity of option values to the underlying asset price (delta and gamma), to the time-to-expiration (theta), to volatility (vega), to the riskless return (rho), and to the payout return (lambda). Of these, gamma is particularly important because it measures the times in the life of the option when replication is likely to prove difficult in practice.

The trinomial method is an extension of the binomial method containing not only all binomial properties but also more flexibility and faster convergence properties ([40]). We first define the binomial method and then go on to the trinomial method later.

### 2.2.1 Binomial Tree

As we mentioned earlier, the underlying process is the geometric Brownian motion described by Equation (1). The analytical differential expression for  $S(t)$  allows us to find the probability density function (PDF) for  $S(t)$ . To do so, we observe from Equation (2) that

$$\begin{aligned} P(\ln(S(t)/S(0)) \leq y) &= P((\mu - \sigma^2/2)t + \sigma W(t) \leq y) \\ &= \Phi\left(\frac{y - (\mu - \sigma^2/2)t}{\sigma\sqrt{t}}\right) \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{y - (\mu - \sigma^2/2)t}{\sigma\sqrt{t}}} e^{-z^2/2} dz, \end{aligned}$$

where  $\Phi$  is the standard normal cumulative distribution function.

Hence, the PDF of  $x$  can be written

$$\frac{dP(S(t) < x)}{dx} = \frac{1}{\sigma\sqrt{2\pi}x\sqrt{t}} \exp\left(-\frac{1}{2}\left\{\frac{\ln x - \ln S(0) - (\mu - \sigma^2/2)t}{\sigma\sqrt{t}}\right\}^2\right)$$

for  $x \in (0, \infty)$ . With this PDF, we can compute the expected value of stock price given the current stock price:

$$E[S(t)|S(0)] = S(0)e^{\mu t}.$$

Thus, the drift  $\mu$  reflects our view on how the mean of the asset will evolve with time. Of course,  $\mu$  is not observable and will differ from one asset to the next. An analogous manipulation applied to

$$E[S^2(t)|S(0)] = S^2(0)e^{2\mu t + \sigma^2 t}$$

yields

$$\text{Var}(S(t)|S(0)) = S^2(0)e^{2\mu t}(e^{\sigma^2 t} - 1).$$

Let us now simulate this random motion of  $S(t)$  with a discrete-time motion where the asset value  $S^n$  at time  $t = t_n$  can rise at time  $t_n + \Delta t$  to  $uS^n$  with probability  $p$  or fall to  $dS^n$  with probability  $(1 - p)$ . We impose on the rise and fall of  $S$  the requirement that over the time interval  $\Delta t$ , the expected values of  $uS$  and  $dS$  and their variances are identical to those of the continuous motion starting with  $S^n$  at  $t = t_n$ . This will be the case if

$$up + d(1 - p) = e^{\mu\Delta t} \tag{6}$$

$$u^2p + d^2(1 - p) = e^{(2\mu + \sigma^2)\Delta t}. \tag{7}$$

There are two equations, (6) and (7), in the three parameters  $u$ ,  $d$ , and  $p$ . They become uniquely defined with the additional assumption that a downward move will cancel a preceding upward move, i.e.,

$$ud = 1.$$

A little algebra will yield values for  $u$  and  $d$ . Starting with Equations (6) and (7), we have

$$p = \frac{e^{\mu\Delta t} - d}{u - d} \quad \text{and} \quad p = \frac{(e^{2\mu+\sigma^2})\Delta t - d^2}{(u + d)(u - d)}.$$

This gives

$$u = A + \sqrt{A^2 - 1}, \quad d = A - \sqrt{A^2 - 1}$$

with

$$A \equiv \frac{e^{-\mu\Delta t} + e^{(\mu+\sigma^2)\Delta t}}{2}.$$

We note that in this model  $p$  is supposed to represent the probability of an upward jump. Hence, we require  $0 < p < 1$ . This will be the case for all  $\Delta t$  because

$$p(\Delta t) \equiv \frac{e^{\mu\Delta t} - A + \sqrt{A^2 - 1}}{2\sqrt{A^2 - 1}} = \frac{1}{2} - \frac{e^{\mu\Delta t} - A}{2\sqrt{A^2 - 1}}$$

and l'Hôpital's rule implies that

$$\lim_{\Delta t \rightarrow 0} p(\Delta t) = \frac{1}{2}.$$

Moreover,  $p(\Delta t) = 0$  for some positive  $\Delta t$  would imply that

$$e^{\mu\Delta t} = d = A - \sqrt{A^2 - 1}$$

or

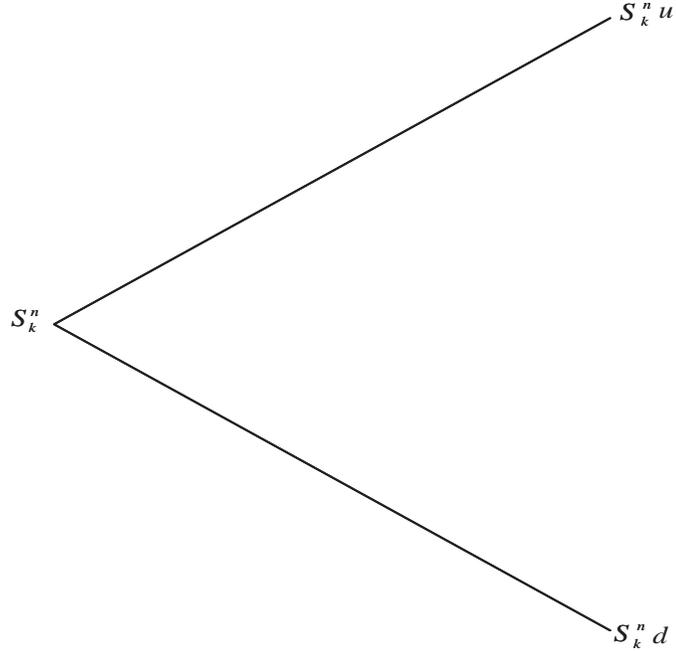
$$e^{\mu\Delta t} \left( A + \sqrt{A^2 - 1} \right) = 1,$$

which is impossible since  $A > 1$  for  $\Delta t > 0$ . Similarly, if  $p(\Delta t) = 1$ , then

$$e^{\mu\Delta t} = u = \tilde{A} + \sqrt{\tilde{A}^2 - 1}.$$

This equation uniquely determines  $\tilde{A}$  as

$$\tilde{A} = \frac{e^{-\mu\Delta t} + e^{\mu+\Delta t}}{2}.$$



**Figure 1:** One-step binomial tree.

Since  $\tilde{A} < A(\Delta t)$  for any  $\Delta t > 0$ , it follows that the equation  $p(\Delta t) = 1$  has no solution for any  $\Delta t > 0$ . Hence,  $p$  may be interpreted as a probability for all  $\Delta t$ .

With  $u$  and  $d$  determined we can now generate a binomial tree. Let  $0 = t_0 < t_1 < \dots < t_N = T$  with  $t_{n+1} - t_n = \Delta t$  be a partition of the interval  $[0, T]$ . Then starting with  $S_0^0$  we can generate all the nodes  $S_k^n, k = 0, 1, \dots, n$  which can be reached by our discrete random walk by setting

$$S_{k+1}^{n+1} = S_k^n u, \quad S_k^{n+1} = S_k^n d,$$

which is shown in Figure 1. The only nodes at the time of expiry  $T$  are

$$S_k^N, k = 0, \dots, N.$$

The value of the option  $V_k^N$  at expiry is known at every node and provides the initial condition for recursively pricing the option at the nodes of the preceding time interval. An arbitrage argument as in the Black-Scholes model is used to find  $V_k^n$  from the option value at time  $t_{k+1}$ . Thus, suppose we have a portfolio at time  $t_n$  of the form

$$\pi = V - \tilde{\Delta} S,$$

where  $\Delta$  is the number of shares of the underlying with value  $S$ . Let  $V^+$  and  $V^-$  be the option values at time  $t_{n+1}$  corresponding to  $Su$  and  $Sd$ . Then the value of the portfolio will be the same at  $t_{n+1}$  if

$$V^+ - \tilde{\Delta}Su = V^- - \tilde{\Delta}Sd,$$

in which case

$$\tilde{\Delta} = \frac{V^+ - V^-}{(u - d)S}.$$

For a European option, this value of the portfolio must be the same as the value of the riskless investment, i.e.,

$$V^+ - \tilde{\Delta}Su = (V - S)e^{r\Delta t}.$$

A little algebra allows us to find  $V$  in terms of  $V^+$  and  $V^-$  as

$$V = e^{-r\Delta t}[\hat{p}V^+ + (1 - \hat{p})V^-],$$

where

$$\hat{p} = \frac{e^{r\Delta t} - d}{u - d}.$$

Note that  $\hat{p}$  depends on  $r$  and, through  $u$  and  $d$ , on the drift parameter  $\mu$ . If  $\hat{p} \in (0, 1)$ , then  $\hat{p}$  may be thought of as a probability and the value of the option  $V_k^n$  at  $S_k^n$  is the discounted value of the expected value of the option at the nearest nodes at the next time level. The price is given by

$$V_k^n = e^{-r\Delta t}[\hat{p}V_{k+1}^{n+1} + (1 - \hat{p})V_k^{n+1}]$$

Hence, we first generate all possible binary random walks by building the binomial tree and then price the option at the nodes of the walk by moving backwards through the tree from  $t = T$  to  $t = 0$ .

We note that the binomial tree reflects the stochastic movement of the underlying asset and is independent of the option. The pricing, however, depends on the option under consideration. The above formula holds for any European option. For an American option, the pricing must be augmented by the requirement that an American-style option's value cannot fall below its intrinsic value,  $(K - S)^+$  for a put or  $(S - K)^+$  for a call. Therefore, the pricing formula is modified as in the explicit finite-difference method — which will be discussed in Subsection 2.3.1 — by requiring that

$$P_k^n = \max \{K - S, e^{-r\Delta t}[\hat{p}P_{k+1}^{n+1} + (1 - \hat{p})P_k^{n+1}]\}$$

for a put, and

$$C_k^n = \max \{K - S, e^{-r\Delta t} [\hat{p}C_{k+1}^{n+1} + (1 - \hat{p})C_k^{n+1}]\}$$

for a call.

The pricing formula is a difference equation for the function  $V(S, t)$  at the three points  $(S, t - \Delta t)$ ,  $(uS, t)$ , and  $(dS, t)$ . It is a common practice in numerical analysis to determine whether such a difference formula represents a consistent approximation to an underlying differential equation. The procedure for finding such a differential equation is standard. Let  $\phi(S, t)$  be an arbitrary smooth function of  $S$  and  $t$ . In particular, we shall assume that

$$\max \left\{ \left| \frac{\partial^3 \phi}{\partial S^3} \right|, \left| \frac{\partial^2 \phi}{\partial t^2} \right| \right\} \leq K$$

for some constant  $K$  and all  $(S, t)$  in the domain where the differential formula is assumed to hold.

By the Taylor expansion about the point  $(S, t)$ , we have

$$\begin{aligned} & [\hat{p}\phi(Su, t) + (1 - \hat{p})\phi(Sd, t)] - e^{r\Delta t}\phi(S, t - \Delta t) \\ &= \hat{p} \left[ \phi + \frac{\partial \phi}{\partial S}(Su - S) + \frac{1}{2} \frac{\partial^2 \phi}{\partial S^2}(Su - S)^2 + \frac{1}{3!} \frac{\partial^3 \phi(\tilde{S}, t)}{\partial S^3}(uS - S)^3 \right] \\ &+ (1 - \hat{p}) \left[ \phi + \frac{\partial \phi}{\partial S}(Sd - S) + \frac{1}{2} \frac{\partial^2 \phi}{\partial S^2}(Sd - S)^2 + \frac{1}{3!} \frac{\partial^3 \phi(\check{S}, t)}{\partial S^3}(uS - S)^3 \right] \\ &- e^{r\Delta t} \left[ \phi - \frac{\partial \phi}{\partial t}\Delta t + \frac{1}{2} \frac{\partial^2 \phi(S, \hat{t})}{\partial t^2}(\Delta t)^2 \right], \end{aligned}$$

where  $\tilde{S}$ ,  $\check{S}$  and  $\hat{t}$  are various intermediate points. Collecting coefficients, we can rewrite the above equation as

$$\begin{aligned} & [1 - e^{r\Delta t}] \phi + [\hat{p}(u - 1)S + (1 - \hat{p})(d - 1)S] \frac{\partial \phi}{\partial S} \\ &+ \frac{S^2}{2} [\hat{p}(u - 1)^2 + (1 - \hat{p})(d - 1)^2] \frac{\partial^2 \phi}{\partial S^2} \\ &= O((\Delta t)^2 + (u - 1)^3 + (d - 1)^3), \end{aligned}$$

where the right-hand side simply indicates that the remainder terms depend on the given rates. Now we observe that

$$\begin{aligned} e^{r\Delta t} &= 1 + r\Delta t + O((\Delta t)^2), \\ \hat{p}(u - 1) + (1 - \hat{p})(d - 1) &= e^{r\Delta t} - 1 = r\Delta t + O((\Delta t)^2), \end{aligned}$$

and

$$\begin{aligned}
\hat{p}(u-1)^2 + (1-\hat{p})(d-1)^2 &= \hat{p}(u-d)(u+d) - 2\hat{p}(u-d) + d^2 - 2d + 1 \\
&= (e^{r\Delta t} - d)(u+d-2) + d^2 - 2d + 1 \\
&= e^{r\Delta t}(u+d-2) - ud + 1.
\end{aligned}$$

At this stage, we bring in the properties of  $u$  and  $d$  which up to now have not been used. We see from the Taylor expansion that

$$u + d - 2 = e^{-\mu\Delta t} + e^{(\mu+\sigma^2)\Delta t} - 2 = \sigma^2\Delta t + O(\Delta t^2)$$

so that

$$e^{r\Delta t}(u+d-2) = \sigma^2\Delta t + O(\Delta t^2)$$

with  $ud = 1$ . It follows that

$$\hat{p}(u-1)^2 + (1-\hat{p})(d-1)^2 = \sigma^2\Delta t + O(\Delta t^2).$$

Dividing through by  $\Delta t$ , we find that

$$\begin{aligned}
\frac{1}{\Delta t} ([\hat{p}\phi(uS, t) + (1-\hat{p})\phi(dS, t)] - e^{r\Delta t}\phi(S, t - \Delta t)) - \left( \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \phi}{\partial S^2} + rS \frac{\partial \phi}{\partial S} - r\phi + \frac{\partial \phi}{\partial t} \right) \\
= O(\Delta t + (u-1)^3/\Delta t + (d-1)^3/\Delta t).
\end{aligned}$$

Finally, we observe that

$$\begin{aligned}
u-1 &= \sqrt{A-1}(\sqrt{A-1} + \sqrt{A+1}) = O(\sqrt{\Delta t}) \\
d-1 &= \sqrt{A-1}(\sqrt{A-1} - \sqrt{A+1}) = O(\sqrt{\Delta t}),
\end{aligned}$$

so that the remainder term on the right goes to zero like  $\sqrt{\Delta t}$ . Therefore, the binomial formula is a *consistent approximation* to the Black-Scholes equation. We see that the drift parameter  $\mu$  does not appear in the Black-Scholes equation so that its solution only depends on the riskless interest rate  $r$ . Since in the limit as  $\Delta t \rightarrow 0$ , the value of  $\mu$  drops out, it is reasonable to generate the binomial tree with  $\mu = r$ , which is an observable quantity. In this case,  $\hat{p} = p \in (0, 1)$  is known as the risk neutral probability. We give the details in Chapter 3.

**Definition 1** *The initial-value problem*

$$\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

is said to be a well-posed problem if:

- A unique solution,  $y(t)$ , to the problem exists;
- For any  $\epsilon > 0$ , there exists a positive constant  $k(\epsilon)$  with the property that, whenever  $|\epsilon_0| < \epsilon$  and  $\delta(t)$  is continuous with  $|\delta(t)| < \epsilon$  on  $[a, b]$ , a unique solution,  $z(t)$ , to the problem

$$\frac{dz}{dt} = f(t, z) + \delta(t), \quad a \leq t \leq b, \quad z(a) = \alpha + \epsilon_0,$$

exists with

$$|z(t) - y(t)| < k(\epsilon)\epsilon, \quad \text{for all } a \leq t \leq b.$$

**Definition 2** *Stability: Given an algorithm  $f(x)$  with  $x$  the input data and  $\epsilon$  the error in the input data, we say the algorithm is numerically stable for the relative error if*

$$x - (x + \epsilon) \simeq f(x) - f(x + \epsilon)$$

and numerically stable for the absolute error if

$$\frac{x - (x + \epsilon)}{x} \simeq \frac{f(x) - f(x + \epsilon)}{f(x)}.$$

Consistency of the approximation is no guarantee that the solution computed with the binomial method will converge to the solution of the Black-Scholes equation as  $\Delta t \rightarrow 0$ . However, convergence is simple to show for a European option by observing that for  $p \in (0, 1)$  the binomial algorithm is necessarily stable so that the lax equivalence theorem below applies. A direct argument can be made as follows.

**Theorem 1** *Lax Equivalence Theorem: Given a well-posed (linear) initial value problem and a consistent finite-difference approximation to it, stability is necessary and sufficient for convergence.*

**Proof** See Myer [51]. ■

A direct argument can be made as follows. Let  $V(S, t)$  be an analytic solution of the Black-Scholes equation with bounded higher derivatives, and let  $V_k^n$  be the value obtained from the binomial tree at the point  $S_k^n$ , where  $S_k^n = dS_k^{n-1}$  and  $S_{k+1}^n = uS_k^{n-1}$ . Then the above analysis shows that the magnitude of the error

$$\text{err}_k^n = |V(S_k^n, t_n) - V_k^n|$$

satisfies

$$\text{err}_k^{n-1} \leq e^{-\Delta t} \{ (p)\text{err}_{k+1}^n + (1-p)\text{err}_k^n \} + C(\Delta t)^{3/2},$$

where  $C$  depends on  $\partial^2 V / \partial t^2$  and  $\partial^3 V / \partial S^3$  but not on  $\Delta t$ . Since  $p \in (0, 1)$ , we see that

$$\max_k \text{err}_k^{n-1} \leq \max_k \text{err}_k^n + C(\Delta t)^{3/2}.$$

From  $\text{err}_k^N = 0$ , it follows now that

$$\max_k \text{err}_k^n \leq (N - n)C(\Delta t)^{3/2} < CT\sqrt{\Delta t}.$$

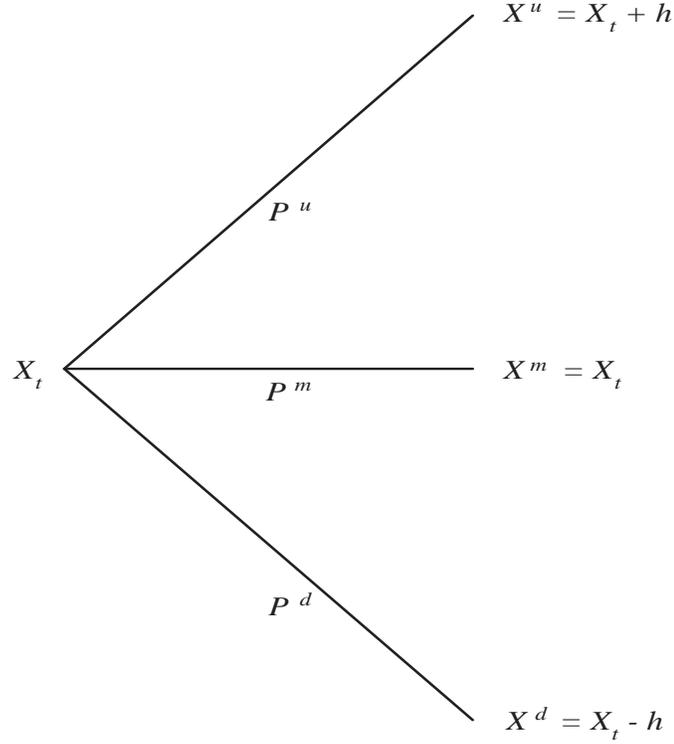
Therefore, the error  $\text{err}_k^n$  goes to zero like  $\sqrt{\Delta t}$ , i.e., the values obtained with the binomial method converge to the values of the Black-Scholes equation, albeit slowly like  $\sqrt{\Delta t}$ .

## 2.2.2 Trinomial Method

The reasoning behind a trinomial tree method is very similar to that of the binomial method. The obvious difference is that a binomial method has only two possible movements from the current state, while the trinomial method has three possible movements from the current state.

Let us simulate this random motion of  $S(t)$  with a discrete-time motion where the asset value  $S^n$  at time  $t = t_n$  can rise at time  $t_n + \Delta t$  to  $uS^n$  with probability  $p_u$ , stay at  $S^n$  with probability  $p_m$ , or fall to  $dS^n$  with probability  $p_d$ . At this time, we need to consider the first, second, and fourth moments which are identical to those of the continuous motion starting with  $S^n$  at time  $t = t_n$ . This will be the case if

$$\begin{aligned} up_u + p_m + dp_d &= e^{\mu\Delta t} \\ u^2p_u + p_m + d^2p_d &= e^{(2\mu+\sigma^2)\Delta t} \\ u^4p_u + p_m + d^4p_d &= e^{4\mu\Delta t+3\sigma^4\Delta t^2} \\ p_u + p_m + p_d &= 1. \end{aligned}$$



**Figure 2:** Trinomial tree with one time step from  $t$  to  $t + \Delta t$ .

We will use a transformation of the underlying process to simplify the computation. Let  $X \equiv \ln S$ . Then

$$dX_t = \alpha dt + \sigma dW_t$$

and for simplicity, suppose  $\alpha = 0$ , so that

$$\begin{aligned} \mathbb{E}[X(t + \Delta t) - X(t)] &= 0 \\ \mathbb{E}[(X(t + \Delta t) - X(t))^2] &= \sigma^2 \Delta t \\ \mathbb{E}[(X(t + \Delta t) - X(t))^4] &= 3\sigma^4 \Delta t. \end{aligned}$$

With this information, let  $X_t^* = \ln(S(t))$ ,  $X_t = X_t^* - \mu t$ ,  $h$  be the step size in space  $X$ , and  $\Delta t$  be the step size in time. For an  $N$ -step trinomial tree,  $\Delta t = T/N$ , where  $T$  is the time horizon. Since we know the distribution of  $(X_{t+\Delta t} - X_t)$ , the criterion for constructing a trinomial tree  $\tilde{X}_t$  is to choose  $(p_u, p_m, p_d)$  and  $h$  so that the distribution of  $(\tilde{X}_{t+\Delta t} - \tilde{X}_t | \tilde{X}_t)$  converges to the distribution of  $(X_{t+\Delta t} - X_t | X_t)$  at all times  $t$  as  $\Delta t \rightarrow 0$ . We match the first, second, and fourth moments of

$(\tilde{X}_{t+\Delta t} - \tilde{X}_t)$  with those of  $(X_{t+\Delta t} - X_t)$  by

$$\begin{aligned} hp_u - hp_d &= 0 \\ h^2 p_u + h^2 p_d &= \sigma^2 \Delta t \\ h^4 p_u + h^4 p_d &= 3\sigma^4 (\Delta t)^2 \\ p_u + p_m + p_d &= 1. \end{aligned}$$

By solving the above equations, we have

$$\left[ p^u, p^m, p^d, h \right] = \left[ \frac{1}{6}, \frac{2}{3}, \frac{1}{6}, \sigma\sqrt{3\Delta t} \right].$$

To price the option on the trinomial tree, one follows the risk-neutral valuation principle. Let  $C(X_t, t)$  denote the option price at node  $X_t$  and time  $t$ ,

$$C(X_t, t) = e^{-r\Delta t} \left( p^u C(X_t + h, t + \Delta t) + p^m C(X_t, t + \Delta t) + p^d C(X_t - h, t + \Delta t) \right),$$

where the terminal condition is

$$C(X_T, T) = g(S_T, T) = g(e^{X_T + \mu T}),$$

and where  $g$  is the payoff function, i.e.,  $g(e^{X_T + \mu T}) = \max(e^{X_T + \mu T} - K, 0)$  for a call option. The details are given in Subsection 2.3.1. Also, there are numerous variants derived from binomial and trinomial tree methods ([19], [63]).

### 2.3 Finite-Difference Method

The Black-Scholes equation for European and some related options can be solved by explicit formulas but for many other options, notably American puts and calls, the boundary conditions cannot be satisfied by the Black-Scholes formula. Hence, an analytic solution is not available and one has to resort to numerical methods. While such methods will always be applied to the Black-Scholes equation in its original form, except perhaps for scaling out the strike price  $K$ , their analysis is usually carried out for simple model problems. If the methods fail to solve the model problem then they usually will fail in a more-complex setting.

The model problem associated with the Black-Scholes equation is the pure initial value problem

$$Lu \equiv \rho u - u_t = 0, \tag{8}$$

$$u(x, 0) = e^{ikx}, \quad -\infty < x < \infty,$$

where  $u_t = \partial u / \partial t$ ,  $\rho > 0$ , and where  $k$  is an arbitrary but fixed integer. By inspection, we see that

$$u(x, t) = e^{ikx - k^2 \rho t}$$

solves this problem. It follows that for all  $k$ , the solution  $u(x, t)$  is bounded for all  $x$  and  $t$ .

**Definition 3** *Let  $\psi(y, t)$  be an arbitrary smooth function. Then  $L_n \psi$  is a consistent approximation to  $L\psi$  if*

$$\lim_{h \rightarrow 0} L_n \psi - L\psi = 0,$$

where  $h$  is mesh size of the grid on which the differential equation is approximated.

A numerical method for this problem typically involves a finite-dimensional algebraic approximation to the problem. The approximation in general depends on a discretization parameter which we shall call  $h$ . In this exposition, we should consider only finite-difference methods for (8), where  $h$  will be identified with the mesh size of the grid on which the differential equation is approximated. There are many different finite-difference methods for this and related problems but all must satisfy the same requirements ([51]).

1. The numerical approximation must be consistent so that one correctly approximates the given problem as  $h \rightarrow 0$ .
2. The numerical method must be convergent as  $h \rightarrow 0$  so that the value of the numerical solution at a given point approaches that of the analytic solution as  $h \rightarrow 0$ .
3. The numerical method must be stable. This means that the value at an arbitrary fixed point in the domain of computation must remain bounded as  $h \rightarrow 0$ .

We shall make these notions more precise when we talk about specific numerical methods for (8). There are three different methods to solve this problem, and we will describe them one by one.

### 2.3.1 Explicit Method

Convergence and stability are very closely related. This is indeed the case for the so-called “well-posed” problem.

**Definition 4** (Hadamard [36]) *A problem is well posed if it has a unique solution which depends continuously on the data of the problem.*

The connections among consistency, convergence, and stability are given by the famous *lax equivalence theorem* (Theorem 1). The practical importance of the theorem is due to the fact that consistency and stability are often easy to establish while convergence of a method may require more work.

It can be shown that the initial value problem for (8) is indeed well posed so that we need only be concerned with consistency and stability of the numerical method.

For the numerical integration of (8) over a time interval  $[0, T]$ , we shall use a three-point finite-difference method. Let

$$t_n = n\Delta t, \quad n = 0, 1, \dots, N$$

$$x_j = -X + j\Delta x, \quad j = 0, 1, \dots, M,$$

where  $\Delta t = T/N$ ,  $\Delta x = 2X/M$  and  $\{X, M, N\}$  are chosen so that the mesh points  $\{x_j, t_n\}$  cover the region over which we need a numerical solution.

We now approximate (8) with the explicit finite-difference formula.

$$L_h u_j^n \equiv \rho \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} - \frac{u_j^{n+1} - u_j^n}{\Delta t} = 0. \quad (9)$$

This formula indeed yields an explicit method since given values at  $j - 1$ ,  $j$ , and  $j + 1$  at time level  $n$ , one can solve explicitly for

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{(\Delta x)^2} [u_{j+1}^n + u_{j-1}^n - 2u_j^n]. \quad (10)$$

Note that for a pure initial value problem, the solution is computed on a narrowing mesh since at each new time level, the right and left endpoints of the  $x$ -mesh have to be moved inward one  $\Delta x$  step from the endpoints at the preceding time level. In particular, if the solution at  $t = T$  is desired at only one point  $x_j$ , one can restrict the computation to a triangular domain with its apex at  $x_j$ . This is like the trinomial method at node  $x_j$ .

In our finite-difference approximation, there are two mesh parameters  $\Delta t$  and  $\Delta x$ . We shall make the assumption that

$$\Delta x = g(\Delta t),$$

where  $g(r) \rightarrow 0$  as  $r \rightarrow 0$ . Thus, the mesh parameter  $h$  can be identified with  $\Delta t$ . The finite difference approximation (9) is a consistent approximation to (8) if

$$\lim_{h \rightarrow 0} [L\phi(x_j, t_n) - L_h\phi(x_j, t_n)] = 0$$

for an arbitrary infinitely differentiable function  $\phi(x, t)$ . This property is easy to verify since Taylor's theorem yields

$$\frac{\phi(x_j, t_{n+1}) - \phi(x_j, t_n)}{\Delta t} = \phi_t(x_j, t_n) + O(\Delta t)$$

and

$$\frac{\phi(x_{j+1}, t_n) + \phi(x_{j-1}, t_n) - 2\phi(x_j, t_n)}{(\Delta x)^2} = \phi_{xx}(x, t) + O(\Delta x),$$

where  $\phi_{xx} = \partial^2 \phi / \partial x^2$  and  $\phi_t = \partial \phi / \partial t$ . We see that  $L_h$  correctly approximates  $L$  provided only that  $\Delta t \rightarrow 0$  and  $\Delta x \rightarrow 0$ . Let us now turn to stability and the behavior of  $u_j^n$  at a given fixed point  $(x_j, t_n)$  where  $n$  is the number of time steps it took to reach the fixed value  $t_n$  as  $\Delta t = T/N \rightarrow 0$ . Note that  $n \rightarrow \infty$  as  $N \rightarrow \infty$  but  $n\Delta t \leq T$ .

We take as an initial condition  $u_j^0 = e^{-ikx_j}$ ,  $j = 0, 1, \dots, M$ , where  $i = \sqrt{-1}$ . Substitution into (10) shows that for  $n = 1$ , we have

$$u_j^1 = 1 + \rho \frac{\Delta t}{(\Delta x)^2} \left( e^{ik\Delta x} + e^{-ik\Delta x} - 2 \right) e^{ikx_j},$$

so that

$$u_j^1 = A(k, \Delta t) e^{ikx_j}, \quad \text{for all } j,$$

where

$$A(k, \Delta t) = 1 + \rho \frac{\Delta t}{(\Delta x)^2} \left( e^{ik\Delta x} + e^{-ik\Delta x} - 2 \right).$$

If we proceed from time level to time level, we find that

$$u_j^n = A^n(k, \Delta t) e^{ikx_j}.$$

It follows that  $u_j^n$  will remain bounded if

$$|A(k, \Delta t)| \leq 1.$$

Since  $A(k, \Delta t) = 1 + \rho \frac{2\Delta t}{(\Delta x)^2} (\cos(k\Delta x) - 1)$ , we see that

$$1 - \rho \frac{4\Delta t}{(\Delta x)^2} \leq A(k, \Delta t) \leq 1,$$

so that  $|A(k, \Delta t)| \leq 1$  whenever  $\rho \frac{4\Delta t}{(\Delta x)^2} \leq 2$  or  $\rho \frac{\Delta t}{(\Delta x)^2} \leq 0.5$ . This inequality is sufficient for stability. One might argue that for certain values of  $k$ , the condition can be relaxed because our estimates are based on the worst-case scenario of

$$\cos(k\Delta x) - 1 = -2.$$

However, in applications the initial condition will be a general function  $u_0(x)$ . Many such functions can be approximated by the complex Fourier series

$$u_0(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx}.$$

The numerical solution now will be a superposition of the solutions for each  $k$ . Hence, one must expect that for some  $k$  and  $\Delta x$ , one can attain  $\cos(k\Delta x) - 1 = -2$ . Thus we cannot allow

$$\rho \frac{\Delta t}{(\Delta x)^2} \geq \frac{1}{2} + \epsilon$$

for a fixed  $\epsilon > 0$  and for all  $\Delta t$  as  $\Delta t \rightarrow 0$ .

Let us examine the implication of this stability restriction for the numerical solution of the Black-Scholes equation. As we have seen, the equation is formally equivalent to

$$\frac{1}{2}\sigma^2 u_{yy} - u_t = 0.$$

The numerical integration of this equation with the explicit Euler method on a uniform grid is subject to the stability condition ([51])

$$\sigma^2 \frac{\Delta t}{(\Delta y)^2} \leq 1.$$

Note that a uniform grid on the  $y$ -axis corresponds to a non-uniform grid on the  $x$ -axis for the original scaled Black-Scholes equation, with

$$\Delta x_i = x_i(e^{\Delta y} - 1) \approx x_i \Delta y.$$

The stability restriction in terms of  $x$  and  $t$  then is

$$x_i^2 \sigma^2 \frac{\Delta t}{(\Delta x_i)^2} \leq 1.$$

While in principle the Black-Scholes equation can be transformed into a constant coefficient equation, it is in general safer to avoid such transformations and discretize the equation in its original form. The explicit Euler method for the Black-Scholes equation on a uniform  $(x, t)$  grid is ([51])

$$\frac{1}{2}\sigma^2 x_j^2 \frac{u_{j+1}^n + u_{j-1}^n - 2u_j^n}{(\Delta x)^2} + r x_j \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - u_j^n + \frac{u_j^{n+1} - u_j^n}{\Delta t} = 0.$$

For an equation with variable coefficients, it is in general not easy to give a stability analysis. However, theory and experience suggest that

1. The first derivative term  $r x u_x$  and the linear term  $r u$  may be ignored in a stability analysis.
2. The stability condition known for the constant coefficient equation should hold locally at all  $x$ .

Thus, the stability condition imposed on the explicit Euler method for the Black-Scholes equation is

$$\sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} \leq 1,$$

which is consistent with the stability condition derived from the transformed Black-Scholes equation. Failure to heed this constraint on  $\Delta x$  and  $\Delta t$  will lead to non-sensical numerical results.

### 2.3.2 Implicit Method

The explicit Euler method is of interest because of its relationship to the binomial method for option pricing as discussed in Subsection 2.2.1. However, much more effective and more generally applicable is the implicit Euler method on a fixed uniform grid placed on the bounded set  $x_0 \leq x \leq X$ ,  $0 \leq t \leq T$ . As before, we write the finite-difference approximation to the equation in the form

$$L_h u_j^n \equiv \frac{1}{2}\sigma^2 x_j^2 \frac{u_{j+1}^n + u_{j-1}^n - 2u_j^n}{(\Delta x)^2} + r x_j \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - u_j^n + \frac{u_j^{n+1} - u_j^n}{\Delta t} = 0,$$

for  $j = 1, \dots, M - 1$ . Here  $u_0^n$  and  $u_M^n$  are assumed known. For example, for a European put

$$u_0^n = e^{-r(T-t_n)}, \quad \text{for } x_0 = 0$$

$$u_M^n = 0, \quad \text{for } x_M \text{ sufficiently large.}$$

Boundary conditions for a call can be read off the put-call parity relationship.

**Definition 5** *Put-Call parity:*  $c + Ke^{-rT} = p + S(0)$ , where  $c$  is a European call price,  $p$  is a European put price, and  $K$  is a strike price.

The essential difference with the explicit method is that at each time level  $t_n$ , a linear system of equations needs to be solved. To be specific, let  $U^n$  be the vector

$$U^n = (u_1^n, \dots, u_{M-1}^n).$$

Then the system to be solved is

$$AU^n = b^n,$$

where  $A$  is the  $(M-1) \times (M-1)$  tridiagonal matrix with entries ([51])

$$\begin{aligned} A_{j,j-1} &= \frac{1}{2}\sigma^2 x_j^2 \frac{1}{(\Delta x)^2} - rx_j \frac{1}{2\Delta x} \\ A_{j,j} &= -\sigma^2 x_j^2 \frac{1}{(\Delta x)^2} - r - \frac{1}{\Delta t} \\ A_{j,j+1} &= \frac{1}{2}\sigma^2 x_j^2 \frac{1}{(\Delta x)^2} + rx_j \frac{1}{2\Delta x} \end{aligned}$$

for  $j = 1, \dots, M-1$ , and where  $b^n$  has the components

$$b_j^n = \begin{cases} -\frac{1}{\Delta t}u_1^{n+1} - A_{10}u_0^n, & j = 1 \\ -\frac{1}{\Delta t}u_1^{n+1}, & 1 < j < M-1 \\ -\frac{1}{\Delta t}u_{M-1}^{n+1} - A_{M-1,M}u_M^n, & j = M-1. \end{cases}$$

It is straightforward to verify that  $L_h\phi(x_j, t_n)$  is a consistent approximation to  $L\phi(x_n, t_n)$  for any smooth function  $\phi$ . It remains to establish stability of the method. We make the assumption that

$$A_{j,j-1} \geq 0, \quad \text{for all } j \geq 1. \quad (11)$$

Then it follows by inspection that

$$|A_{jj}| > A_{j,j-1} + A_{j,j+1}, \quad \text{for all } j.$$

This inequality implies that  $A$  is strictly diagonally dominant, which can be shown to imply that  $A$  is invertible. Hence if (11) holds then the value  $\{u_j^n\}$  can be computed. Suppose that at some interior mesh point  $(x_k, t_p)$  with  $0 < k < M$ ,  $0 \leq p < N$ , the value  $u_k^p$  is larger than all the other

values, i.e.,  $u_k^p \geq u_j^n$  for all  $j$  and all  $n$ . Since the initial and boundary data are non-negative, we have necessarily that  $u_k^p \geq 0$ . Then it follows from  $A_{k,k-1}, A_{k,k+1} \geq 0$  that

$$\begin{aligned} A_{kk}u_k^p + A_{k,k-1}u_{k-1}^p + A_{k,k+1}u_{k+1}^p &\geq A_{kk}u_k^p + A_{k,k-1}u_k^p + A_{k,k+1}u_k^p \\ &= -ru_k^p - \frac{1}{\Delta t}u_k^p, \end{aligned} \quad (12)$$

and Equation (12) is less than  $-\frac{1}{\Delta t}u_k^{p+1}$ , which is inconsistent with the  $k$ th equation of

$$AU^p = b^p.$$

Hence, there cannot be a value of  $u_k^n$  which exceeds the maximum value of  $u$  at  $x_0$ , at  $x_M = X$  or  $t_N = T$ . Since  $u$  at  $x = 0$ ,  $x = X$  and  $t = T$  is bounded and independent of  $\Delta x$  and  $\Delta t$ , it follows that  $u_k^p$  is uniformly bounded above all  $\Delta x$  and  $\Delta t$ . An analogous argument shows that the minimum of  $\{u_j^n\}$  is bounded below by the minimum of the data functions. Therefore, if condition (11) holds, then the numbers  $\{|u_j^n|\}$  are uniformly bounded so that the implicit Euler method is stable for all  $\Delta x$  and  $\Delta t$ .

The unconditional stability of the implicit Euler method is bought at the expense of having to solve the linear system

$$Au^n = b^n$$

at each time level. Fortunately, the tridiagonal structure of  $A$  makes this solution extremely simple and rapid if a special Gaussian elimination procedure known as the Thomas algorithm ([51]) is employed. A tridiagonal matrix is factored into lower and upper triangular matrices  $\tilde{L}$  and  $\tilde{U}$  so that

$$A = \tilde{L}\tilde{U}$$

This can be achieved by setting  $\tilde{L}_{ij} = \tilde{U}_{ij} = 0$  for  $|i - j| > 2$ ,  $\tilde{U}_{i,i-1} = \tilde{L}_{i,i+1} = 0$ ,  $\tilde{U}_{ii} = 1$ , and  $\tilde{L}_{i,i-1} = A_{i,i-1}$ . The remaining elements  $\tilde{L}_{ii}$  and  $\tilde{U}_{i,i+1}$  are found for  $i = 1$  from  $\tilde{L}_{11} = A_{11}$  and  $\tilde{L}_{11}\tilde{U}_{12} = A_{12}$  and for  $i = 2, \dots, M - 1$  from

$$\tilde{L}_{i,i-1}\tilde{U}_{i-1,i} + \tilde{L}_{ii} = A_{ii}, \quad \tilde{L}_{ii}\tilde{U}_{i,i+1} = A_{i,i+1}.$$

It is straightforward to verify from

$$(\tilde{L}\tilde{U})_{ij} = \sum_{k=1}^{M-1} \tilde{L}_{ik}\tilde{U}_{kj} = \tilde{L}_{i,i-1}\tilde{U}_{i-1,j} + \tilde{L}_{ii}\tilde{U}_{ij}$$

that  $A = \tilde{L}\tilde{U}$ , i.e., that  $A$  has been factored into lower and upper triangular matrices. The linear system

$$Au^n = b^n$$

can now be solve by simple substitution. First we find the solution  $y^n$  of

$$\tilde{L}y^n = b^n$$

and then obtain  $u^n$  from

$$\tilde{U}u^n = y^n.$$

It is clear from  $Au^n = \tilde{L}\tilde{U}u^n = \tilde{L}y^n = b^n$  that  $u^n$  is the desired solution.

## CHAPTER III

### MONTE CARLO SIMULATION

In this chapter, we describe a method of pricing options using the Monte Carlo simulation method. Given the increasing complexity of options that contain early exercise characteristics, a number of methods based on Monte Carlo simulation have been studied and implemented recently. Monte Carlo simulation is a flexible method whose applicability does not depend on the dimension of the problem and does not suffer from the curse of dimensionality. However, the computational cost incurred by Monte Carlo simulation might get very expensive in some cases. Therefore, various variance reduction techniques are often implemented in many applications.

We first start with the following elementary definition.

**Definition 6** An *option* gives its holder the right to trade an underlying asset at a specified price (the exercise or strike price) on (and sometimes before) a specified date (the exercise or maturity date).

There are two basic types of options. A *call option* gives its holder the right to buy an underlying asset. A *put option* gives its holder the right to sell an underlying asset.

An *European option* can only be exercised at maturity. Therefore, its value at expiration is known in many cases. In particular, an investor would choose to exercise a put only if  $S < K$ , where  $S$  is the stock price at maturity and  $K$  is the strike price. Otherwise, the option expires worthless. Similarly, a holder of a call will exercise his option only if  $S > K$  at expiration. The payoffs from European options for put and call options are

$$p_0(S_T, T) \equiv (K - S_T)^+ \quad \text{and} \quad c_0(S_T, T) \equiv (S_T - K)^+,$$

respectively, where  $S_T$  is the stock price at maturity. The owner of an American option has the right to exercise early. An American option, therefore, is worth at least the price of its equivalent European option. The details of American-style option values will be discussed as we progress. The following theorem gives a general idea of what the American-style option price could be.

**Theorem 2** *It is never optimal to exercise an American call before the maturity if the stock does not pay any dividend. It is sometimes optimal to exercise an American put early on a stock paying no dividends.*

**Proof** See Merton [48]. ■

This theorem says that the call price of an American option with no dividend payment is the same as the European call option price. However, American put prices are no less than that of a European put.

With this basic knowledge on option pricing, we propose new algorithms for pricing options. We start by discussing the standard call option on a single underlying asset. The price process in the Black–Scholes model can be expressed as an Itô process. Let  $\{W(t), 0 \leq t \leq T\}$  be a one-dimensional Brownian motion process relative to filtration  $\mathcal{F}_t$ . The stock price then has the form

$$S_t = S_0 e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma W(t)},$$

where  $S_0 > 0$ ,  $\mu$  and  $\sigma > 0$  are constants. The above equation can be written as

$$\frac{dS}{S} = \mu dt + \sigma dW. \quad (13)$$

Equation (13) can be interpreted heuristically as expressing the relative or percentage increment  $dS/S$  in  $S$  during an instant of time  $dt$ . Assuming that the stock does not pay any dividends, all the return to an investment in the stock comes in the form of stock price appreciation (or depreciation); and so the relative increment  $dS/S$  can be interpreted as the instantaneous rate of return on the stock. The expected instantaneous rate of return is  $\mu$ , and the standard deviation of the instantaneous rate of return is  $\sigma$ , called the volatility of  $S$ .

In the calculation above, the  $-\frac{1}{2}\sigma^2$  term in the Black–Scholes stock price processes gets “eaten up” by the second-order term from Itô’s Lemma.

By a similar method, we can calculate characteristics of a money market account having a price process of the form

$$M_t = M_0 e^{rt}$$

or

$$\frac{dM}{M} = r dt. \quad (14)$$

By using Equations (13) and (14), we can calculate the discounted stock price. The stock price, measured in units of the money market account or discounted at the rolled-over instantaneous interest rate  $r$ , is  $S/M$ , which has the differential given by ([43])

$$d\left(\frac{S}{M}\right) = \left(\frac{S}{M}\right) [(\mu - r) dt + \sigma dW]. \quad (15)$$

We can rewrite Equation (15) as

$$S_t^* = S_0 e^{(\mu - r - \frac{1}{2}\sigma^2)t + \sigma W(t)}. \quad (16)$$

The difference,  $\mu - r$ , of the stock's appreciation rate and the money market rate from Equation (15) is called the *risk premium* and measures the additional rate per unit time present for investing in the risky stock rather than in the riskless money market. The quantity  $\theta$  defined by

$$\theta \equiv \frac{\mu - r}{\sigma} \quad (17)$$

is called the *market price of risk*. The market price of risk  $\theta$  relates the risk premium,  $\mu - r$ , and the volatility,  $\sigma$ . The probability measure  $\mathcal{Q}(\cdot)$  is called the *risk-neutral probability measure* or the *equivalent martingale measure* associated with the market price of risk  $\theta$  of Equation (17).

**Theorem 3 (Girsanov)** *Let  $W(t), 0 \leq t \leq T$ , be a Brownian motion on a probability space  $(\Omega, \mathcal{F}, P)$ . Let  $\mathcal{F}_t, 0 \leq t \leq T$ , be the accompanying filtration, and let  $\theta(t), 0 \leq t \leq T$ , be a process adapted to this filtration. For  $0 \leq t \leq T$ , define*

$$\begin{aligned} \mathcal{W}(t) &\equiv \int_0^t \theta(u) du + W(t), \\ Z(t) &\equiv \exp \left\{ - \int_0^t \theta(u) dW(u) - \frac{1}{2} \int_0^t \theta^2(u) du \right\}, \end{aligned}$$

and define a new probability measure by

$$\mathcal{Q}(A) = \int_A Z(T) dP, \quad \forall A \in \mathcal{F}.$$

Under  $\mathcal{Q}$ , the process  $\mathcal{W}(t), 0 \leq t \leq T$ , is a Brownian motion.

Now we define a new process  $\{\mathcal{W}(t), 0 \leq t \leq T\}$  as

$$\mathcal{W}(t) = W(t) + \theta t,$$

where  $\mathcal{W}$  is a standard Brownian motion under the risk-neutral measure  $\mathcal{Q}$  by Girsanov's Theorem.

Under the risk-neutral probability measure,  $\mathcal{Q}$ , we can write (16) as the following,

$$S_t = S_0 e^{(r - \frac{1}{2}\sigma^2)t + \sigma \mathcal{W}(t)}, \quad (18)$$

where  $S_0 > 0$ ,  $r$  and  $\sigma > 0$  are constants.

With this background, this chapter is composed of three parts. First, we propose a new heuristic algorithm for pricing path-dependent derivative securities by applying an early exercise boundary obtained by regression and then optimizing its coefficients. Next, we develop an adaptive policy search algorithm for searching for the optimal policy in exercising an American-style option. The option price is obtained by first estimating the optimal option exercising policy and then evaluating the option with the estimated policy via simulation. Both high-biased and low-biased estimators of the option price are computed, and we show that this algorithm leads to the convergence to the true optimal policy with probability one. We end this chapter by showing how the variance of the relevant estimators is reduced.

### ***3.1 Boundary with Regression***

This section describes a heuristic algorithm for pricing American-style path-dependent derivative securities. The typical Monte Carlo simulation method used in European call option pricing is to use simulation to estimate the expectation. That is,

$$c_0(S) = \mathbb{E}_{\mathcal{Q}}[e^{-rT}(S_T - K)^+].$$

Therefore, pricing a European-style option is very simple and straightforward. The American option pricing problem for a *call* is to find

$$C_0(S) = \sup_{\tau \in [0, T]} \mathbb{E}_{\mathcal{Q}}[e^{-r\tau}(S_{\tau} - K)^+]$$

over all stopping times. Throughout this section, we use a discrete-time approximation to the problem, where we restrict the exercise opportunities to lie in the finite set of times  $0 = t_0 < t_1 <$

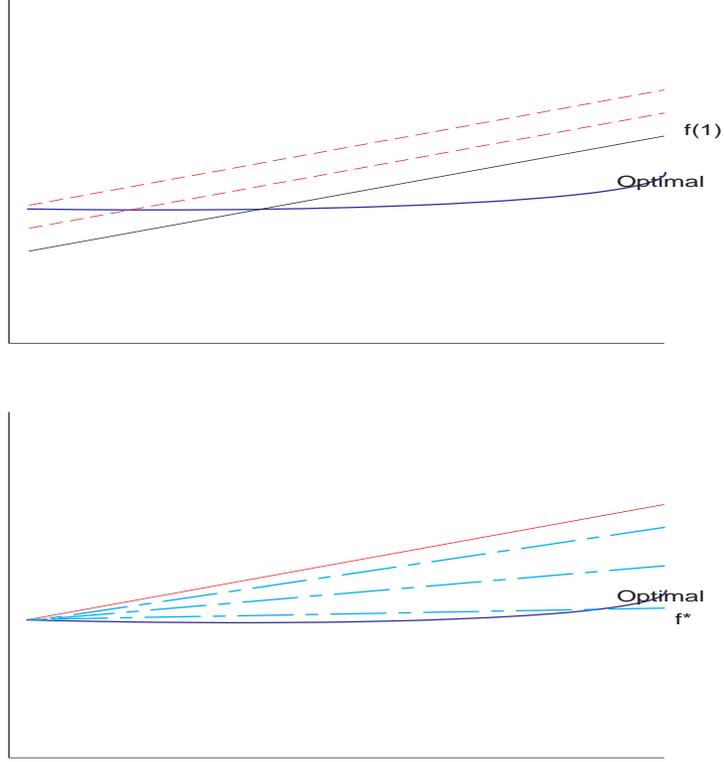
$\dots < t_d = T$  and let  $\Delta t \equiv T/d$ . To estimate the value of an American option, we would simulate a path of asset prices,  $S_0, S_1, S_2, \dots, S_d$ , at corresponding times  $0 = t_0 < t_1 < \dots < t_d = T$ , then calculate a discounted option value corresponding to this path, and average the results over many simulated paths.

We first obtain the exercise boundary (policy), and then we use this boundary to exercise the right since it enables us to use simulation directly. To have an initial boundary, we first obtain the highest price in the single path through the time horizon. We repeat this process a few times to obtain enough data to regress against time. We initially incorporate into the regression function linear (L), quadratic (Q) and cubic (C) terms and set them as an exercise boundary which we optimize later. Findings indicate that a higher adjusted  $R^2$  in regression functions tends to yield better option estimation in general, where adjusted  $R^2$  can be defined as follows:

$$R_a^2 = 1 - \frac{\frac{SSE}{n-p}}{\frac{SSTO}{n-1}} = 1 - \left( \frac{n-1}{n-p} \right) \frac{SSE}{SSTO},$$

and where  $n$  is the number of observations and  $p$  is the number of parameters.

This suggests that if we optimize the coefficients of the highest adjusted  $R^2$  regression, then we will have a better estimator for options. It is now necessary to optimize the boundary. A key assumption here is that there exists an optimal policy for the American-style option under consideration which maximizes the option value. Suppose that  $f^*(x) = \alpha + \beta x$  is the optimal exercise boundary and let  $f^{(1)}(x)$  be our first approximation of the exercise boundary using the least squares method. First, we find better coefficients for  $f^{(1)}(x)$  to improve the boundary. Let  $f^{(1\cdot)}(x) = a^{(1)} + bx$ , let  $b$  remain the same, and let its corresponding option price be  $P^{(1)}$ . Now, set  $a^{(2)} = a^{(1)} + \epsilon$ , where  $\epsilon > 0$ , and let  $f^{(2\cdot)}(x) = a^{(2)} + bx$ , with resulting option price  $P^{(2)}$ . Then compare the estimates using  $f^{(1\cdot)}(x)$  and  $f^{(2\cdot)}(x)$ . If the latter is larger ( $P^{(1)} < P^{(2)}$ ), we keep moving in the same direction. Otherwise, we move in the reverse direction until the estimate stops increasing. After that, we change coefficient  $b$ . With the last coefficient  $a^{(i)}$ , we set  $b^{(2)} = b^{(1)} + \delta$ , where  $\delta > 0$  and  $f^{(i2)}(x) = a^{(i)} + b^{(2)}x$ . Denote price estimates with  $f^{(i1)}(x)$  and  $f^{(i2)}(x)$  by  $\psi^{(1)}$  and  $\psi^{(2)}$ , respectively. Then compare the price estimates,  $\psi^{(1)}$  and  $\psi^{(2)}$ . If  $\psi^{(1)} < \psi^{(2)}$ , then we keep moving in the same direction. Otherwise, we change the direction; that is,  $\delta < 0$ . By the same token, we can find the  $b^{(j)}$  maximizing the estimate. With this algorithm, we find a  $f^{(ij)} = \hat{f}(x) = a^{(i)} + b^{(j)}x$



**Figure 3:** Searching for optimal boundary.

such that  $\|f^* - \hat{f}\|_2 < \tilde{\epsilon}$ , where  $\tilde{\epsilon} > 0$ . Figure 3 demonstrates this algorithm.

### 3.1.1 Implementation and Numerical Results

We provide some numerical results to illustrate our algorithm. We first start with the standard call option on a single asset which pays a continuous dividend and for which the price process is a geometric Brownian motion process described by

$$S_n = S_{n-1} e^{(r-\delta-\sigma^2/2)\Delta t + \sigma\sqrt{\Delta t}Z_n} \quad (19)$$

for some random variable  $Z_n$  independent of the parameters and the initial stock price  $S_0$ , riskless rate  $r$ , volatility  $\sigma$ , dividend rate  $\delta$ , and time increment  $\Delta t$ .

For discrete dividend cases,

$$S_n = S_{n-1} e^{(r-\sigma^2/2)\Delta t + \sigma\sqrt{\Delta t}Z_n} - D_i,$$

where  $D_i$  is the  $i$ th dividend. With this in mind, we first price American call options.

**Table 1:** Price estimate with IEB and OEB. The optimized coefficients are only for a linear function. Option parameters:  $K = 100$ ,  $r = 0.05$ ,  $\delta = 0.10$ ,  $T = 1.0$ , and  $\sigma = 0.2$  with  $\Delta t = 0.25$ yr and 1M replications. Error unit is %.

$S$	L-O	L	Q	C	Rel err(L-O)	Rel err(L)
70	0.122	0.116	0.125	0.111	0.826	4.132
80	0.668	0.613	0.613	0.616	0.299	8.507
90	2.258	2.054	1.993	2.003	1.954	10.812
100	5.700	4.841	4.484	4.343	0.541	15.530
110	11.418	11.417	11.242	11.158	0.679	0.670
120	19.975	19.001	19.296	19.303	0.125	4.995
130	29.987	27.801	29.986	29.421	0.043	7.330

### 3.1.1.1 Example 1: An American call with a single asset with dividend payments

For this example, we apply our algorithm to price a standard American call option with a single underlying with continuous dividend payments following Equation (19). The value of an American put option is equal to the value of an American call with the following change of parameters:  $S_t \rightarrow K$ ,  $K \rightarrow S_t$ ,  $r \rightarrow \delta$ , and  $\delta \rightarrow r$ .

We obtained two different price estimates with Initial early Exercise Boundary (IEB) and Optimized early Exercise Boundary (OEB) and compared them to Glasserman and Broadie’s outputs. For the first stage of sampling for data collection, we used 320 as the number of time steps and 3200 replications ( $320 \times 10$ ). For this case, we used the linear, quadratic, and cubic regressions for IEB and we optimized the coefficients of the linear regression function.

Table 1 shows that OEB performs better than all other IEBs except for the case  $S_0 = 70$ . This finding suggests that by optimizing the quadratic or cubic function, a better estimate will be achieved for the option prices. Relative error is computed by

$$\frac{|\text{Estimate} - \text{True}|}{\text{True}} \times 100.$$

Also, Table 2 compares our estimates to Broadie and Glasserman’s estimates as well as the true values ([8]). It also shows that our estimates are better for all cases, including out-of-the-money, at-the-money, or in-the-money.

The option price is not affected much by  $\Delta t$ , i.e., the number of exercise opportunities for the optimized coefficients for linear and quadratic boundaries. As can be seen in Table 3, option

**Table 2:** Price estimate comparison with OEB and the stochastic tree method of Glasserman and Broadie. The optimized coefficients are only for a linear function. Option parameters:  $K = 100$ ,  $r = 0.05$ ,  $\delta = 0.10$ ,  $T = 1.0\text{yr}$ , and  $\sigma = 0.2$ . TRUE(E) represents the true value of European options.

$S$	OEB	TRUE	TRUE(E)	High	Low
70	0.122	0.121	0.120	0.117	0.115
80	0.668	0.670	0.654	0.662	0.649
90	2.258	2.303	2.197	2.316	2.251
100	5.700	5.731	5.302	5.824	5.628
110	11.418	11.341	10.154	11.603	10.988
120	19.975	20.000	16.154	20.329	19.743
130	29.987	30.000	24.060	30.154	29.763

**Table 3:** Price estimates with IEB and OEB. The optimized coefficients are only for a linear function (L-O) and quadratic function (Q-O). Option parameters:  $S_0 = 70$ ,  $K = 100$ ,  $r = 0.05$ ,  $\delta = 0.10$ ,  $T = 1.0\text{yr}$ , and  $\sigma = 0.2$ .

Time step	Reps	L-O	Q	C	L	Q-O	C
4	2M	0.121	0.120	0.120	0.120	0.121	0.120
8	2M	0.122	0.120	0.120	0.121	0.122	0.118
16	1M	0.122	0.120	0.120	0.118	0.121	0.118
32	1M	0.121	0.117	0.117	0.117	0.122	0.117
64	1M	0.123	0.116	0.116	0.116	0.124	0.117
128	1M	0.121	0.114	0.114	0.113	0.122	0.114
256	1M	0.122	0.114	0.114	0.113	0.124	0.113

prices do not vary depending on the step size for optimized boundaries. However, the option prices decrease if we make  $\Delta t$  smaller for non-optimized boundaries. That means that the boundary is not the optimal boundary.

It may seem logical that the more chances to exercise, i.e., smaller  $\Delta t$ , the higher the option price. For example, if Option A has 4 exercise opportunities and Option B has 8, Option B should be more expensive than Option A. This is true as long as we know the true early exercise boundary. However, we approximate the true boundary; and as long as our approximation is close enough to the true boundary, it does not really matter how many exercise opportunities one has since we can make a best decision every time. This finding provides us the opportunity to check if our OEB is really close to the true boundary. This argument is supported by our simulation results in Table 3's

3rd and 7th columns.

Also, we determined that if the options are deep out-of-the-money, we may not have enough data to regress. Therefore, we must run more replications to obtain data. On the other hand, if the options are deep in-the-money, we have a good quantity of data. In such cases, we do not need to run many replications.

### 3.1.1.2 Example 2: American min put options with two assets

This example is an American put on two independent assets following geometric Brownian motion, so that we can compare our results with Rogers (2001). The underlying processes can be described by the following,

$$S_t^i = S_0^i \exp \left( (r - \sigma_i^2/2) t + \sigma_i W_i(t) \right),$$

with payoff function

$$f(t) = \max_{i=1,2} e^{-rt} (K - S_i(t))^+.$$

The payoff function can be rewritten as

$$f(t) = e^{-rt} \left( K - \min_{i=1,2} S_i(t) \right)^+.$$

We again optimize the early exercise boundary from the initial boundary. First, a one-piece linear boundary is optimized. With the optimized boundary, we compute option values and a slightly lower estimate is obtained. We next separate the time maturity as multiple pieces of boundaries. Since multiple pieces give more flexibility for fitting the optimal boundary, the estimate based on multiple pieces is recommended in this example.

For simplicity, two piecewise linear boundary functions are used. Let  $T_1 = [0, T/2)$  and  $T_2 = [T/2, T]$ , where  $T$  is the time to maturity. From these two time frames, we find initial early exercise boundaries, say  $f_1(t_1)$  and  $f_2(t_2)$ , where  $0 \leq t_1 < T/2$  and  $T/2 \leq t_2 \leq T$ . From these boundaries, we again optimize the coefficients of each function depending on time regions. First, we optimize the boundary function  $f(t)$ ,  $0 \leq t < T/2$ . After the coefficients are optimized, we optimize the second boundary using the first optimized boundary function. Let us describe the boundary functions based on this searching process:

1. Let  $f(t)$ ,  $0 \leq t \leq T$ , be the initial boundary function and  $f_1(t)$ ,  $0 \leq t < T/2$ , and  $f_2(t)$ ,  $T/2 \leq t \leq T$ , be the initial boundaries for each time region. Therefore,  $f(t)$  can be described as the union of two boundary functions,  $f(t) = f_1(t) \cup f_2(t)$ .
2. Optimize the first boundary function and let the optimized boundary for this region be  $\hat{f}_1(t)$ . The overall boundary function after optimization is then  $f(t) = \hat{f}_1(t) \cup f_2(t)$ .
3. Optimize the second half boundary function  $f_2(t)$ ,  $T/2 \leq t \leq T$ , by using  $\hat{f}_1(t)$ ,  $0 \leq t < T/2$ . The overall boundary function after optimization is  $f(t) = \hat{f}_1(t) \cup \hat{f}_2(t)$ .

Once we obtain the optimized boundary functions, we can apply the forward algorithm to price the options. In this example, we use  $K = 100$ ,  $T = 0.5$ ,  $r = 0.06$ , and  $\sigma_1 = \sigma_2 = 0.6$ .

We used 20000 replications to perform the optimization and this process takes less than 10 seconds with a 1.8Mhz P4 processor. Then we ran 200000 replications for each case, and this takes less than 20 seconds. In each case, we set  $\Delta t = T/50$ .

**Table 4:** Price estimation with optimized two-piecewise linear early exercise boundary. The optimized coefficients are only for a linear function. Parameters are  $K = 100$ ,  $T = 0.5$ ,  $r = 0.06$ , and  $\sigma_1 = \sigma_2 = 0.6$

$S_1$	$S_2$	OEB	FD	Rogers	European
80	80	37.14	37.30	37.63	36.859
80	100	31.96	32.08	32.30	31.639
80	120	29.06	29.14	29.38	28.652
100	100	24.94	25.06	25.17	24.728
100	120	20.87	20.91	21.10	20.610
120	120	15.91	15.92	16.02	15.704

In Table 4, we compare our optimized early exercise method (OEB) to the finite-difference (FD) method, Rogers' upper bound estimate, and the European price computed by numerical integration. Our estimates are always lower than the other methods. The difference, however, is very small (less than 0.5%).

We can assume that if we have an  $n$ -piecewise linear boundary function, we can provide a better estimate than with our two-piecewise linear boundary functions. However, we then need to optimize the coefficients of  $n$  functions and the computational cost grows tremendously. At this stage, we do

not attempt to have more than two-piecewise functions. Since the two-piecewise functions give us a better estimate, they drive us to develop the adaptive algorithm discussed in Section 3.2.

### 3.2 An Adaptive Simulation Algorithm

In this section, we focus on the presentation of an adaptive simulation algorithm for pricing an American option. Let the price processes of  $n$  assets  $\{\vec{S}_t = (S_t^{(1)}, \dots, S_t^{(n)}), 0 \leq t \leq T\}$  be general  $n$ -dimensional stochastic processes adapted to a filtration  $\mathbf{F} \equiv \{\mathcal{F}_t : 0 \leq t \leq T\}$ . For quite general stochastic processes, the American-style option's initial value is given by the solution to an optimal stopping problem ([12]) with a risk-neutral measure  $\mathcal{Q}$ ,

$$P_0 = \sup_{\tau_s \in [0, T]} E_{\mathcal{Q}}[\alpha \mathcal{J}(\vec{S}_{\tau_s})], \quad (20)$$

where  $\alpha$  is a discount factor,  $\mathcal{J}(\cdot)$  is a function of the underlying  $n$ -dimensional processes, and  $\tau_s$  is a stopping time.

For the discretized stochastic processes version of the continuous-time problem, let  $\{\vec{S}_{t_n}, 0 \leq t_n \leq T\}$  be the stock price at time step  $t_n$ , where  $t_n = n\Delta t$ ,  $\Delta t = T/N$ , and  $N$  is the total number of exercisable times. These  $n$ -dimensional stochastic processes are right-continuous and adapted to the filtration  $\{\mathcal{F}_t : 0 \leq t \leq T\}$ . Hence, it is progressively measurable with respect to  $\{\mathcal{F}_t\}$  ([26]). The optimal stopping time is the earlier of the maturity and the first passage time to the exercise boundary. Consequently, the American-style put options could be valued as

$$P_0 = \sup_{\{B(t_n): t_n \in [0, T]\}} E_{\mathcal{Q}}[\alpha \mathcal{J}(\vec{S}_{\min(\tau_B, T)})], \quad (21)$$

where  $\tau_B$  is the first passage time from  $\vec{S}_0$  to an exercise boundary  $B(t)$ ,  $t \in [0, T]$ , and  $f(\cdot)$  is a payoff function.

Under regularity considerations, the solution to the optimal stopping problem exists and it is unique. Namely, there exists an optimal option exercising policy of the American-style option under consideration which maximizes the option value. In this section, we show that finding an optimal exercise boundary provides an optimal exercise rule, and it enables us to estimate the price of securities using Monte Carlo simulation. Now, we consider how to approximate the early exercise boundary,  $\{B(t) : t \in [0, T]\}$ . Searching for an optimal boundary in our algorithm is equivalent to

finding an optimal boundary of the parameter vector based on the Kullback-Leibler *cross entropy* (CE) measure, which defines a distance between the two distributions  $f(y)$  and  $h(y)$  (see [25]). That can be written as

$$\mathcal{D}(f, h) = \int f(y) \ln \frac{f(y)}{h(y)} dy. \quad (22)$$

CE has been widely used in rare-event simulations and especially for finding importance sampling density functions (see [39]).

Our objective is to design an algorithm that starts from some initial distribution  $f$  and iteratively converges to the degenerate probability density function (PDF) with unit mass on the true optimal policy. The idea is then to compute the parameter of PDF  $f$  that minimizes the cross entropy between  $f$  and  $h$ , where  $f$  is a probability distribution from which an early exercise policy is sampled and  $h$  is the optimal exercise policy distribution.

**Remark 1** *Variance minimization (VM) with importance sampling and the CE method are used to approximate the optimal importance sampling density in rare-event simulation. In our algorithm, we identify the optimal sampling density for obtaining an option exercising policy which best approximates the optimal option exercising policy.*

The basic idea of our algorithm is to design an adaptive procedure for identifying the optimal option exercising policy over a policy space  $\Pi$ , which contains the optimal policy for exercising an American-style option. We define the performance measure for a policy which measures the quality of the policy. This measure is usually the value of the option corresponding to the to-be-evaluated policy. These policies can be sampled from a family of probability distributions over  $\Pi$ . With both the performance measure and a probability distribution for sampling the option exercising policy, we design an iterative algorithm to obtain an optimal probability distribution such that the estimated optimal distribution samples the optimal exercising policy with the highest likelihood.

It is possible to show that we can redefine Equation (22) as

$$\max_{\nu \in V} \{D(u, \nu) := E_{\nu_1} [1_{\{\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) \geq \gamma\}} W(\mathbf{\Lambda}, u, \nu_1) \ln f(\mathbf{\Lambda}, \nu)]\}, \quad (23)$$

where  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$  is an option payoff value with policy vector  $\mathbf{\Lambda}$  and  $W(\cdot)$  is a weight. More details will be given later. Next, take  $\nu_1 = u$  in the above formula; we keep solving (23) to find a  $\nu^*$

---

**Algorithm 1** General Adaptive Simulation Algorithm

---

**Initialization:**

- Specify
  - $N_E$ : number of exercise opportunities,
  - $N_S$ : number of sample stock price paths,
  - $N_G$ : number of exercise policies,
  - $\gamma_G$ :  $\rho$  quantile of performance measure, where  $\rho \in (0, 1)$ .
- Specify an initial policy sampling distribution  $f(\cdot, \nu_0)$ .
- Sample a set of policies  $\mathbf{\Lambda}^{\{0\}} \equiv \{G_1^{(0)}, \dots, G_p^{(0)}\}$  from the population policy space  $\Pi(\mathbb{G}_1, \dots, \mathbb{G}_p)$ , where  $\mathbf{\Lambda}^{\{0\}} \sim f(\cdot, \nu_0)$ .

**Repeat until a specified stopping rule is applied:**

Define a performance measure  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$  which measures the quality of a policy  $\mathbf{\Lambda}$  over a given sample path set  $\mathcal{G}$  for updating the policy sampling distribution.

- Simulate a set of sample paths for  $n$  stock prices,  $\mathcal{G}(S^{(1)}, \dots, S^{(n)}) \equiv \{(S_t^{(1)}(\omega), \dots, S_t^{(n)}(\omega)) : 0 \leq t \leq T, \omega \in \Omega\}$ .
- For each  $\mathbf{\Lambda}_i^{\{k\}}$  ( $i = 1, 2, \dots, N_G$ ), evaluate  $\mathcal{H}(\mathbf{\Lambda}_i^{\{k\}}, \mathcal{G})$ .
- Obtain  $\hat{\gamma}_k = \mathcal{H}(\mathbf{\Lambda}^{\{k-1\}}(\lceil(1 - \rho)N_G\rceil), \mathcal{G})$  and set  $\hat{\gamma}_k$  to be  $\gamma_G$ .
- Compute

$$\max_{\nu \in V} \left\{ D(u, \nu) := E_{\nu_k} [1_{\{\mathcal{H}(\mathbf{\Lambda}^{\{k\}}, \mathcal{G}) \geq \gamma_G\}} W(\mathbf{\Lambda}^{\{k\}}, u, \nu_k) \ln f(\mathbf{\Lambda}^{\{k\}}, \nu)] \right\}$$

and set its solution to be  $\nu_{k+1}$ .

- Sample the next set of policies as  $\mathbf{\Lambda}^{\{k+1\}} = \{G_1^{(k+1)}, \dots, G_p^{(k+1)}\}$  with  $f(\cdot, \nu_{k+1})$ .
  - $k \leftarrow k + 1$
-

so that CE of  $f(\cdot, \nu^*)$  and  $h(\cdot, \nu)$  is minimized. The CE algorithm works well if the distribution of the basic random variables in the model has finite support or if the random variables belong to the so-called natural exponential family (NEF) (see [39] for a definition). Analytical solutions for  $\nu^*$  exist in the NEF. However, if there is no analytical solution, we need to implement numerical optimization procedures to solve Equation (23).

First, a high-level description of our adaptive simulation algorithm is summarized in Algorithm 1. We now provide a detailed discussion as follows.

**INITIALIZATION:**

1. Pick positive simulation parameters:  $\{\vec{K}, M, N_E\}$ ,  $N_S$ ,  $N_G$ , and  $\rho \in (0, 1)$ , where  $\vec{K} = \{K_1, K_2, \dots, K_p\} \in \mathcal{R}^p$ .
2. Generate  $N_S$  stock price sample paths of the process  $\{\vec{S}_t = (S_t^{(1)}, \dots, S_t^{(n)}) \in \mathcal{R}^n\}$  over time horizon  $[0, T]$ .
3. Policy classes are defined over a bounded subset of  $\Theta \times [0, T]$ , where  $\Theta \triangleq \{(\theta_1, \theta_2, \dots, \theta_p) : \|\theta_i\| \leq \beta(\sum_{i=1}^p K_i^2 + T^2)^{1/2}, i = 1, \dots, p\}$ .
  - Make a coarse grid on a region  $[0, T] \times [\underline{\theta}_1, \bar{\theta}_1] \times \dots \times [\underline{\theta}_p, \bar{\theta}_p]$ , where  $\underline{\theta}_i$  and  $\bar{\theta}_i$  are lower and upper values of the  $i$ th policy space, respectively.
4. Let  $k = 0$ . Initialize  $\vec{P}^{\{0\}}$  to be a joint probability mass function of  $N_E$  independent discrete uniform random variables with masses on  $m\Delta s$  ( $m = 0, 1, \dots, M$ ).
5. An exercise boundary of the American option is given by an  $N_E$ -tuple  $(\vec{y}_1, \vec{y}_2, \dots, \vec{y}_{N_E}) \in \mathcal{R}^p$  on the grid generated in step 3.

**ITERATIVE POLICY IMPROVEMENT:**

6. Use  $\vec{P}^{\{k\}}$  to generate  $N_G$  exercise boundaries.
7. Apply each exercise boundary to the set of  $N_S$  sample paths and obtain an approximated value  $V_{n_b}$  ( $n_b = 1, 2, \dots, N_G$ ) of the option value. Find a “good” set of exercise boundaries

based on  $V_{\lceil \rho N_G \rceil}^*$ , where  $V_{\lceil \rho N_G \rceil}$  denotes the upper  $\rho$  quantile of all the approximated option values. Let  $\vec{B}_i^{(k)}$  denote the  $i$ th highest option value achieving the exercise policy where  $k$  is the policy search iteration time.

8. Let  $k \leftarrow k + 1$ . Compute a new joint probability mass function (PMF)  $\vec{P}^{\{k\}}$  depending on each time step from good sets,  $\vec{B}_i^{(k)}(t)$ , where  $t = 1, \dots, N_E$ ,  $i = 1, \dots, \lceil \rho N_G \rceil$ , and  $\rho$  is a quantile of the good set. For a fixed time  $t$ ,

$$P_j^{*\{k\}} = \frac{E_{P^{\{1\}}}[\mathbf{1}_{\{\mathcal{H}(\Lambda, \mathcal{G}) \geq x\}} W(\Lambda^{\{k\}}, P_j^{\{k\}}, P_j^{\{1\}}) | X_t = x_{tj}] P(X_t = x_{tj})}{E_{P^{\{1\}}}[\mathbf{1}_{\{\mathcal{H}(\Lambda, \mathcal{G}) \geq x\}} W(\Lambda^{\{k\}}, P, P^{\{1\}})]},$$

where  $j = 1, \dots, M$ .

9. Exit the loop if the option price converges; else, go to step 6.
10. Obtain the optimal exercise boundary based on  $\vec{P}^{\{k\}}$ .
11. Generate a set of  $N_1$  ( $N_1 > N_S$ ) stock price sample paths and evaluate the American option over these sample paths based on the optimal boundary obtained at step 10.

**Remark 2** *At step 8, we can directly find an optimal parameter  $\nu$  by equating the gradient  $\nabla_\nu D(u, \nu)$  to zero if there exists an optimal solution of Equation (23). Moreover, if the expectation and differentiation operators can be interchanged, the stationary point of Equation (23) can be obtained from the solution of the following nonlinear equations:*

$$\nabla_\nu D(u, \nu) = E_{\nu_1} [\mathbf{1}_{\{\mathcal{H}(\Lambda, \mathcal{G}) \geq \gamma\}} W(\Lambda, u, \nu_1) \nabla \ln f(\Lambda, \nu)] = 0$$

and for sample estimation

$$\nabla_\nu \hat{D}(u, \nu) = N_G^{-1} \sum_{i=1}^{N_G} [\mathbf{1}_{\{\mathcal{H}(\Lambda_i, \mathcal{G}) \geq \gamma\}} W(\Lambda_i, u, \nu_1) \nabla \ln f(\Lambda_i, \nu)] = 0,$$

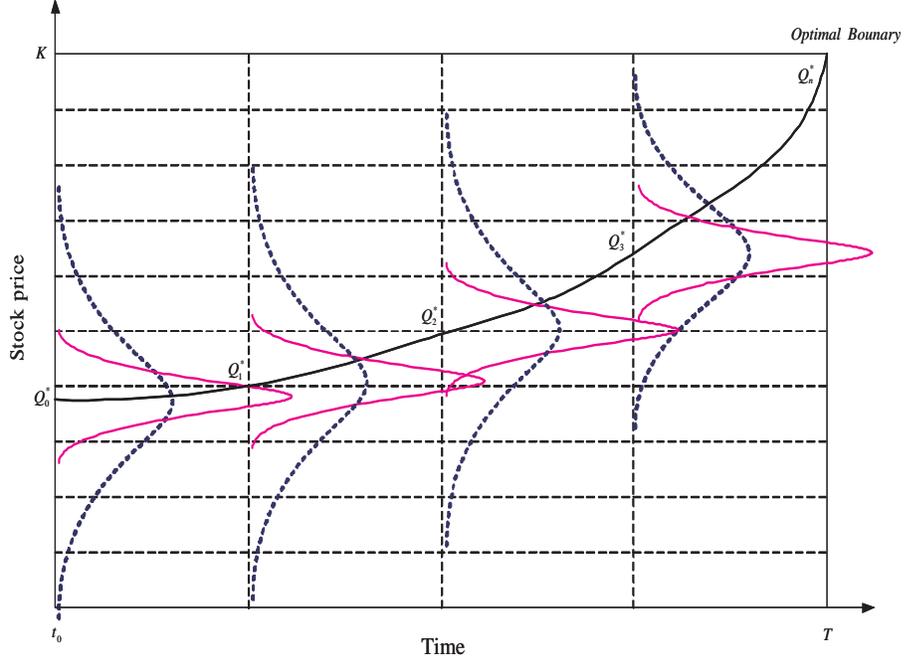
where  $\Lambda_1, \dots, \Lambda_{N_G}$  are random samples from the PDF  $f(\cdot, \nu_1)$ .

*If the components of random vector  $Y$  are independent and each has a distribution in the natural exponential family, then there exists an analytical solution. The explicit solution at the  $j$ th element is*

$$\nu_j^* = \frac{E_u[Y_j \mathbf{1}_{\{\mathcal{H}(Y) \geq \gamma\}}]}{E_u[\mathbf{1}_{\{\mathcal{H}(Y) \geq \gamma\}}]} = \frac{E_{\nu_1}[\Lambda_j \mathbf{1}_{\{\mathcal{H}(\Lambda) \geq \gamma\}} \mathcal{H}(\Lambda) W(\Lambda, u, \nu_1)]}{E_{\nu_1}[\mathbf{1}_{\{\mathcal{H}(\Lambda) \geq \gamma\}} \mathcal{H}(\Lambda) W(\Lambda, u, \nu_1)]},$$

---

\*As usual,  $\lceil \cdot \rceil$  is a ceiling function.



**Figure 4:** Distribution of the boundary for each time step. As we have more iterations, the shape of the distribution becomes narrower, and it converges to the optimal boundary.

where  $\mathbf{\Lambda} = (\Lambda_1, \dots, \Lambda_{N_E})$  and  $\Lambda_1, \dots, \Lambda_{N_E}$  are independent random variables following an NEF.

For sample estimation,

$$\nu_j^* \approx \frac{\sum_{i=1}^{N_G} \Lambda_{ji} \mathbf{1}_{\{\mathcal{H}(\mathbf{\Lambda}_i) \geq \gamma\}} W(\mathbf{\Lambda}_i, u, \nu_1)}{\sum_{i=1}^{N_G} \mathbf{1}_{\{\mathcal{H}(\mathbf{\Lambda}_i) \geq \gamma\}} W(\mathbf{\Lambda}_i, u, \nu_1)},$$

where  $\mathbf{\Lambda} = (\Lambda_1, \dots, \Lambda_{N_E})$  and  $\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_{N_G}$  are independent random variables such that  $\mathbf{\Lambda}_j$  has an NEF distribution. Note that  $\mathbf{\Lambda}_j$  is an  $N_E$ -dimensional vector of  $j$ th sample.

**Remark 3** At step 8,  $\gamma$  can be replaced with  $\hat{\gamma}$ . The details for obtaining this approximation are discussed in Section 3.2.2.

Figure 4 shows the general shape of the distributions for the boundary of each time step. As the algorithm iterates more, it puts more weight around the optimal points, and it will eventually converge to the optimal boundaries.

Now we provide the computational complexity of the adaptive simulation algorithm. We first discuss the complexity of finding an optimal policy and then that of the low and high estimates. We ignore the generation of the lattice points, stock paths, and policy. Computing the optimal boundary

can be split into two parts. One is computing the value of policies and the other is updating the new PMF from the good sets. We compute a option value based on each policy with sample paths. Therefore, the maximum number of computations is

$$N_G \times N_S \times N_E,$$

where  $N_G$  is number of policies,  $N_S$  is the number of stock paths, and  $N_E$  is the number of exercise opportunities. Once the policies are evaluated, updating the new PMF is performed only with good sets. That is, the number of computations is

$$[\rho N_G] \times N_E.$$

Therefore, the total running time for finding an optimal boundary takes  $O(N_G N_S N_E)$ .

Finally, after we have computed the optimal boundary, we price the options with  $N_1$  sample paths, where  $N_1 \gg N_S$ . That is,  $N_E \times N_1$  operations take place.

### 3.2.1 High and Low Biased Estimators

We obtained two estimators, high biased and low biased estimators. Let  $\Omega$  be the stock price space and assume that there exists a true price and a corresponding early exercise boundary. Let  $S_{(i)}$  be a subset of  $\Omega$  and

$$\lim_{k \rightarrow \infty} \bigcup_{i=1}^k S_{(i)} = \Omega.$$

For the fixed  $S_{(n)}$ , we generate  $m$  policies, denoted  $B_n^{(j)}(t)$ , where  $t \in [0, T]$  and  $j = 1 \dots, m$ . Using an adaptive algorithm, we can find an early exercise boundary under the stock price set  $\{S_{(n)}\}$ . Let the optimal policy under  $S_{(n)}$  be  $B_n^*(t)$ . The exercise strategy is that if the sample stock price path hits the optimal boundary for the first time, then exercise. If it never hits the boundary, then we exercise the option at maturity if it is in-the-money.

#### Low biased estimator, $\phi$

Let the low biased estimator be  $\phi$ . This can be obtained in the following way. For fixed time  $t$ , a sample stock price  $S_t$  hits the  $B^*(t)$  obtained by the adaptive algorithm and its payoff value

$f_t(S_t) = (K - S_t)^+$  for a put option on a single asset. Therefore, the low biased estimator is a discounted price of intrinsic value when the underlying hits the boundary for the first time,  $\phi = e^{-rt} f_t(S_t)$ . We replicate it many times and its expected value is our low biased estimate, denoted by  $E[\phi_0(N_1)]$ , where  $N_1$  is the number of replications.

### High biased estimator, $\Phi$

For the high biased estimator  $\Phi$ , we adopt an anticipative policy when the sample path hits the boundary for the first time at time  $t$ . The option is exercised and the payoff is given by the higher of  $S_t$  and  $S_{t+\Delta t}$ . Namely, the payoff value at  $t$  is  $F_t(S_t) = \max(f_t(S_t), e^{-r\Delta t} f_{t+\Delta t}(S_{t+\Delta t}))$ . Then we discount back to current value,  $\Phi = e^{-rt} F_t(S_t)$ . By the same method, we have the high biased estimate, denoted by  $E[\Phi_0(N_1)]$ .

We compare two estimators. The high biased estimator is always higher than the low biased estimator, and the difference between the two estimators converges to zero if  $\Delta t$  approaches zero.

**Proposition 1** *The high biased estimate  $E[\Phi_0(N_1)]$  is no less than the low biased estimate  $E[\phi_0(N_1)]$ .*

**Proof** The proof is straightforward to show. At time  $t_s$ , where a sample stock price path hits the exercise boundary for the first time, we have

$$F_{t_s}(S_{t_s}) = \max(f_{t_s}(S_{t_s}), e^{-r\Delta t} f_{t_s+\Delta t}(S_{t_s+\Delta t})) \geq f_{t_s}(S_{t_s}).$$

Since  $F_b \geq f_b$  for  $0 \leq b \leq t_s$ , where  $t_s$  is the stopping time, we can conclude that  $E[\Phi_0] \geq E[\phi_0]$ . ■

**Proposition 2** *The difference between the two estimates converges to zero if the adapted  $\sigma$ -field filtration is right-continuous.*

### Proof

We only need to show that the price difference between the stopping time ( $t_s$ ) and one-step ahead ( $t_s + \Delta t$ ) converges to zero as the one-step interval length converges to zero. Since the stochastic process  $S$  is right continuous and adapted to the filtration  $\{\mathcal{F}_t\}$  as well as measurable

with respect to  $\{\mathcal{F}_t\}$ , it is straightforward to see that

$$\lim_{\Delta t \rightarrow 0} \| S_{t_s} - S_{t_s + \Delta t} \| = 0.$$

That shows that the differences between the two estimates converges to zero. ■

We also know that if we have a big enough policy set under the given policy space, we can obtain a good approximation to the true price with a large enough sample path set. The following proposition states this:

**Proposition 3** *If the sizes of the sample stock price set and the policy set approach infinity and the policy space contains the optimal policy, the policy obtained from the policy iteration converges to the true early exercise boundary, and the corresponding price converges to the true option value under the given policy space  $\Pi(\mathbb{G})$ .*

**Proof**

Let  $|Q_n| = n$  be the size of the policy set and let  $|S_k| = k$  be the size of the sample path set. First, fix  $|S|$  and let  $S = \Omega$ . Also let  $q_i$  be the probability of finding a true boundary out of  $i$  boundaries (policies) with  $S$ . Then there exist  $n$  and  $m$ , where  $m < n$ , such that

$$P(q_n > q_m) \rightarrow 1.$$

Since we have a finite number of sample paths, the above statement might not be accomplished with a small number for the sample path set. However, since the size of the sample path set converges to infinity, the above statement is always true. By Equation (21), its corresponding price is the true price. ■

**Corollary 1** *For an American put on a single asset, the estimate by a policy obtained with the adaptive algorithm converges to the true option value if the numbers of sample price paths and policies approach infinity.*

At this point, we can enumerate the advantages of our algorithm. First, the adaptive algorithm requires little knowledge on the structure of the option policies. By iterating, we can find the structure

of the optimal policy automatically. Second, it does not require an approximation of the conditional value functions as in LSM ([45],[65]). Hence, we do not have to deal with this source of errors. In addition, we need not be concerned with choosing basis functions in our algorithm. Our CPU time is significantly improved compared to Rogers [53]. We could not directly make comparisons to other competitors since our computation is run in Matlab for all cases. However, we compared our CPU time to LSM CPU time for a jump-diffusion process case.

### 3.2.2 Convergence of the Optimal Exercising Policy

In this section, we discuss some issues related to the convergence of our adaptive simulation presented in Section 3.2. All proofs are provided in the appendix. Let us introduce additional notation.

- $\nu_{n-1}$  :  $(n - 1)$ th accumulated data set,
- $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$  : the highest value obtained by choosing the best option exercising policy among a given set of policies  $\mathbf{\Lambda} = \{\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_{N_G}\}$  with a fixed stock price set  $\mathcal{G}$ . That is,  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) \equiv \sup_{\Lambda(t_k): t_k \in [0, T]} E_{\mathcal{Q}}[\alpha(K - S_{\min(\tau_{\Lambda}, T)}(\mathcal{G}))^+]$ .

First, we show how to compute an optimal vector for one iteration. For the  $(n - 1)$ th iteration, adaptive updating of the exercising policy at the  $n$ th iteration can be obtained. First, for a fixed  $\nu_{n-1}$ , let  $\gamma_n$  be the  $(1 - \rho)$  quantile of  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$  under  $\nu_{n-1}$ . That is,

$$P_{\nu_{n-1}} \{ \mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) \geq \gamma_n \} \geq \rho,$$

where  $\mathbf{\Lambda} \sim f(\cdot, \nu_{n-1})$ . Therefore, a simple estimator  $\hat{\gamma}_n$  of  $\gamma_n$  can be obtained by drawing a random sample  $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \dots, \mathbf{\Lambda}_{N_G}$  from  $f(\cdot, \nu_{n-1})$ , calculating  $\mathcal{H}(\mathbf{\Lambda}_i, \mathcal{G})$  for all  $i$ , ordering them from smallest to biggest,  $\mathcal{H}_{(1)} \leq \mathcal{H}_{(2)} \leq \dots \leq \mathcal{H}_{(N_G)}$ , and finally evaluating the  $(1 - \rho)$  quantile as

$$\hat{\gamma}_n = \mathcal{H}_{(n, [(1-\rho)N_G])}.$$

Then the adaptive estimation of  $\nu_n$  is carried out as follows. For a fixed  $\gamma_{n-1}$ , derive  $\nu_n$  from the solution of the following program:

$$\max_{\nu \in V} \{ D(u, \nu) := E_{\nu_{n-1}} [1_{\{\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) \geq \gamma_{n-1}\}} W(\mathbf{\Lambda}, u, \nu_{n-1}) \ln f(\mathbf{\Lambda}, \nu)] \}. \quad (24)$$

The stochastic counterpart of Equation (24) is obtained by replacing  $\gamma_{n-1}$  with  $\hat{\gamma}_{n-1}$  and the expected value with the corresponding sample average, where the sample is the same as that used to compute  $\gamma_{n-1}$ .

Now, we consider the general case. Suppose we are interested in

$$\max_{\nu \in V} \{D(u, \nu) := E_{\nu_1}[1_{\{\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) \geq \gamma\}} W(\mathbf{\Lambda}, u, \nu_1) \ln f(\mathbf{\Lambda}, \nu)]\}. \quad (25)$$

Since there is typically no analytical solution for Equation (25), numerical optimization procedures must be implemented in such a case. Given a sample  $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \dots, \mathbf{\Lambda}_{N_G}$  from  $f(\cdot, \nu)$ , we can estimate the optimal solution  $\nu^*$  of Equation (25) using our proposed algorithm from Section 3.2.

We discuss convergence of the adaptive algorithm presented in Section 3.2. The parameter  $\rho$  plays a crucial role when we expect the adaptive algorithm to converge to the correct value. The priori determination of which  $\rho$  is acceptable can be a difficult task<sup>†</sup>. To overcome this problem, we suggest that  $\rho$  can be changed adaptively. The modified new step will be introduced later. We now state a series of assumptions that will be necessary in the sequel.

**Assumption A:**

$$P_{\nu}(\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) \geq x) > 0 \quad \text{for all } \nu \in V,$$

where  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$  is the sample performance and the set of random policies  $\mathbf{\Lambda} \sim f(\cdot, \nu)$ .

Assumption A simply ensures that the probability being estimated does not vanish. This assumption is satisfied when the distribution of  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$  has an infinite tail. For zero-tail distributions such as truncated exponential, the assumption holds as long as either  $x$  is less than the maximum value of the function  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$ , or if there is a positive probability that  $x$  is attained.

**Lemma 2** *Suppose that Assumption A holds and let  $\nu \in V$ . Then there exists  $\rho_x^* > 0$  such that  $\gamma(\nu, \rho) \geq x$  for any  $\rho \in (0, \rho_x^*)$ .*

Lemma 2 shows that by decreasing  $\rho$  sufficiently, we can force the quantile  $\gamma$  to increase. In particular, we can force  $\gamma$  to increase at least by some pre-specified amount  $\delta > 0$ . Thus, we can

---

<sup>†</sup>Based on our experiments,  $\rho \in (0, 0.10)$  would be a reasonable choice depending on the structure of the products under consideration.

modify our algorithm by replacing  $\rho$  with  $\rho_{t-1}$  and adding the following step after Step 6 in Section 3.2 as follows:

*Let  $\rho_n$  be such that  $\rho(\nu_n, \rho_n) \geq \min\{x, \gamma(\nu_{n-1}, \rho_{n-1})\} + \delta$ , where  $\delta$  is a positive constant.*

With this modification, it is clear that  $\gamma(\nu_{n-1}, \rho_{n-1}) \geq x$  for some  $n$ . Also, we can rewrite Equation (24) as

$$\max_{\nu \in V} \{E_{\nu_{n-1}} [1_{\{\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) \geq \min\{x, \gamma(\nu_{n-1}, \rho_{n-1})\}\}} W(\mathbf{\Lambda}, u, \nu_{n-1}) \ln f(\mathbf{\Lambda}, \nu)]\}. \quad (26)$$

**Proposition 4** *Suppose that Assumption A holds. Then the modified version of the adaptive algorithm converges to an optimal solution after a finite number of iterations.*

A few comments are in order about the modified version of our algorithm. First, the algorithm aims to reach  $\nu^*$  by a sequence of calculations controlled by the parameters  $\rho_n$ , which should be kept as large as possible. In fact, in many cases the  $\rho_n$  are initialized with a “moderate” value such as 0.2 and are never reduced. This follows from the fact that by construction of the algorithm, the distribution of  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$  when  $\mathbf{\Lambda} \sim f(\cdot, \nu_{n+1})$  tends to have larger tails than when  $\mathbf{\Lambda} \sim f(\cdot, \nu_n)$ , and thus  $\gamma(\nu_{n+1}, \rho) > \gamma(\nu_n, \rho)$  for all  $\rho$  in these cases.

In that sense, the modified algorithm makes sure that our algorithm works even though the latter property fails. As discussed above, there always exist  $\rho_{n+1}$  satisfying the property in the step we added. In practice, we can put  $\delta = 0$  in the algorithm.

Now, we turn to the issue of replacing the expected values with their respective sample means in the modified version of our algorithm. That is,

$$\hat{\nu}_n \in \operatorname{argmax}_{\nu \in V} \left\{ \frac{1}{N_G} \sum_{i=1}^{N_G} 1_{\{\mathcal{H}(\mathbf{\Lambda}_i, \mathcal{G}) \geq \hat{\gamma}(\mathbf{\Lambda}, \rho_{n-1})\}} W(\mathbf{\Lambda}_i, u, \nu) \ln f(\mathbf{\Lambda}_i, \nu) \right\},$$

where  $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \dots, \mathbf{\Lambda}_{N_G}$  is a sample from  $f(\cdot, \hat{\nu}_{n-1})$  (for a given realization of  $\hat{\nu}_{n-1}$ ),  $W(\mathbf{\Lambda}_i, u, \hat{\nu}_{n-1})$  is a weight function, and  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho_{n-1})$  is defined as the sample  $(1 - \rho_{n-1})$  quantile of  $\mathcal{H}(\mathbf{\Lambda}_1, \mathcal{G}), \dots, \mathcal{H}(\mathbf{\Lambda}_{N_G}, \mathcal{G})$ . The fact that  $\gamma(\nu, \rho_{n-1})$  is replaced by  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho_{n-1})$  means that the new step added in our algorithm may not be carried out. For example, it could happen that the samples  $\mathcal{H}(\mathbf{\Lambda}_i, \mathcal{G})$  produced at two consecutive iterations are identical. The proposition below shows that when  $N_G$  is large enough, the new step can be executed.

**Proposition 5** *Suppose Assumption A holds and let  $a \in (0, x]$ . Let  $\nu \in V$  and  $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \dots$  be i.i.d. with density  $f(\lambda, \nu)$ . Then there exists  $\rho_a > 0$  and a random  $N_a > 0$  (size of the sampled exercising policies), such that with probability one,  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho) \geq a$  for all  $\rho \in (0, \rho_a)$  and all  $N_G \geq N_a$ . Moreover, the probability that  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho) \geq a$  for a given  $N_G$  goes to one exponentially fast with  $N_G$ .*

The above proposition shows not only that  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho)$  — which corresponds to the  $(1 - \rho)$ -quantile of the highest option value obtained in the samples — reaches any threshold  $a$  for sufficiently small  $\rho$  and sufficiently large  $N_G$  (which ensures that the algorithm terminates), but also that one expects  $N_G$  not to be too large due to the exponential convergence, at least for moderate values of  $\rho$ . In any case, to ensure that the sample size grows as needed, we can check after the added step in our algorithm if such a  $\rho_{n+1}$  can be found. If not, the sample size should be increased and the process is repeated.

Now we can compare the approximated solution  $\tilde{\nu}$  and the true  $\nu^*$  using the asymptotic analysis for optimal solutions of the stochastic optimization problem discussed in [56]. Let us consider a real-valued function  $h(\lambda, \nu)$  of two vector variables  $\lambda$  and  $\nu$ , and let  $\mathbf{\Lambda}$  be a random vector policy with probability measure  $F(\lambda, \nu)$ . Suppose that

$$l(\nu) = E[h(\mathbf{\Lambda}, \nu)] = \int h(\lambda, \nu) dF(\lambda, \nu)$$

exists for all  $\nu$  in a region  $V \subset \mathcal{R}^n$  and consider the problem

$$\max \{ l(\nu), \nu \in V \}.$$

Let  $\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_{N_G}$  be an i.i.d. random sample of exercising policy random vectors with common distribution  $F(\lambda, \nu)$ . Then the stochastic counterpart of the program is

$$\max \bar{l}_{N_G}(\nu) = N_G^{-1} \sum_{i=1}^{N_G} h(\mathbf{\Lambda}_i, \nu), \quad \nu \in V,$$

where

$$h(\mathbf{\Lambda}_i, \nu) = \mathbf{1}_{\{\mathcal{H}(\mathbf{\Lambda}_i, \mathcal{G}) \geq \gamma\}} W(\mathbf{\Lambda}_i, u, \nu) \ln f(\mathbf{\Lambda}_i, \nu).$$

Denote by  $\tilde{P}$  be the value of an American put on a single asset,

$$\tilde{P} = \sup_{\nu \in V} \left\{ \sup_{\mathbf{\Lambda}} \left( N_1^{-1} \sum_{i=1}^{N_1} \alpha(K - S_{\min(\tau_{\mathbf{\Lambda}}, T)}^{(i)})^+ \right) \right\},$$

with exercising policy  $\mathbf{\Lambda} \sim f(\cdot, \nu)$ , and let  $\tilde{\nu}_{N_G}$  be the optimal solution of the program. We consider  $\tilde{P}$  and  $\tilde{\nu}_{N_G}$  to be statistical estimators of the optimal value  $P^*$  and an optimal solution  $\nu^*$  of the program.

For any fixed  $\nu \in V$ , we have by the Strong Law of Large Numbers that  $l_{N_G}(\nu)$  converges to  $l(\nu)$  with probability 1. It is possible to show that the above convergence is uniform if the set  $V$  is compact and the following two conditions hold ([56]).

**Assumption B:** For almost every  $\lambda$  with respect to the probability measure  $F(\lambda, \cdot)$ , the function  $h(\lambda, \cdot)$  is continuous on  $V$ .

**Assumption C:** The family  $\{|h(\lambda, \nu)|, \nu \in V\}$  is dominated by an integrable function with respect to  $h(\lambda, \cdot)$ .

**Lemma 3** *Suppose that Assumptions B and C hold. Then the expected value function  $l(\nu)$  is continuous on  $V$ . If, in addition, the set  $V$  is compact, then w.p.1  $\bar{l}_{N_G}(\nu)$  converges to  $l(\nu)$  uniformly on  $V$ .*

**Theorem 4** *Suppose Assumptions B and C hold and the set  $V$  is compact. If  $\nu^*$  is a unique maximizer of  $l(\nu)$  over  $V$ , then  $\tilde{\nu}_{N_G}$  converges to  $\nu^*$  w.p.1. Moreover,  $\tilde{P}$  — which is the estimated option value with the sampled early exercise policy by  $f(\cdot, \tilde{\nu}_{N_G})$  — converges to  $P^*$  (the true price) w.p.1.*

From this, we obtain a consistency result: under appropriate assumptions as  $N_G \rightarrow \infty$ , the distance between  $\tilde{\nu}_{N_G}$  and the solution set of Equation (25) goes to zero w.p.1.

Since our algorithm has a discrete  $M$ -point distribution, we have to verify that the above arguments can be applied into our case. We shall classify distributions into two categories: ones with unbounded support like exponential, Poisson, and normal distributions, and ones with bounded support such as uniform(a,b), truncated exponential, and discrete  $n$ -point distributions. More formally, we say that  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$  has bounded support if  $P(|\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})| > a) = 0$  for some  $a$  large enough. Note that bounded support distributions can be viewed as zero-tail distributions as compared to their counterparts with infinite tails.

Finite-support distributions possess the following important general property ([55] and [56]).

**Proposition 6** *Let  $x^*$  be the maximum value of  $\mathcal{H}(\cdot)$  over the discrete set*

$$\mathcal{Y} = \{y_{11}, \dots, y_{1M}\} \times \dots \times \{y_{NE1}, \dots, y_{NEM}\},$$

*and suppose that the maximizer of  $\mathcal{H}(\cdot)$  over  $\mathcal{Y}$  (call it  $y^*$ ) is unique. Suppose that the random vector  $Y$  has independent components with discrete distribution  $\mathcal{Y}$ . Then the solution obtained is the atomic measure with mass at  $y^*$ .*

Figure 5 shows a standard put option price convergence depending on different initial stock prices. In most cases, prices have converged after about 20 iterations. Interestingly, if the option is deep out-of-the-money, then the exercising policy distribution converges more quickly than that of deep in-the-money.

### 3.2.3 Implementation and Numerical Examples

In this section, we implement our algorithm to price several options with various initial stock prices. We compare our estimates to other algorithms proposed in other papers.

To enhance the computation, we sort the early exercise boundaries for each iteration when we use the policy iteration. Also, we use a coarse mesh for the time horizon for obtaining the early exercise boundary. However, we use a cubic spline method for computation of the early exercise boundary so that we can apply a small time interval for the low and high biased estimates.

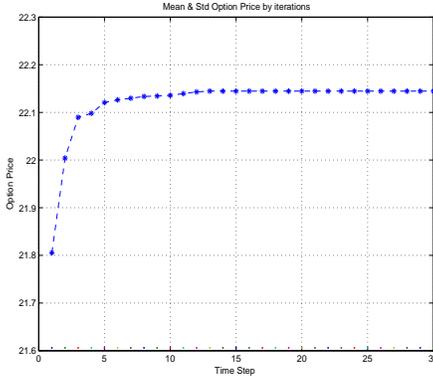
**Remark 4** *If the asset price model is multiplicative, i.e., a geometric Brownian motion model, then the early exercise boundary is convex (see [66]).*

#### 3.2.3.1 Example 1: American put options on a single asset

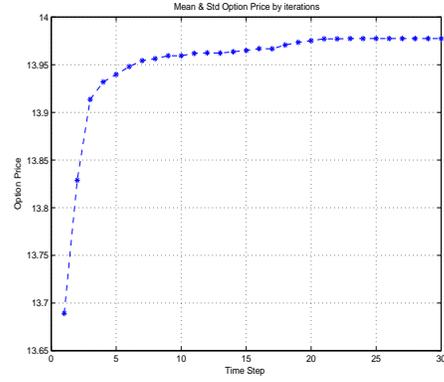
Our first example is an American put on a single asset following the geometric Brownian motion (GBM) model, whose process is given by

$$S_t = S_0 \exp \left\{ \left( r - \frac{\sigma^2}{2} \right) t + \sigma \mathcal{W}_t \right\},$$

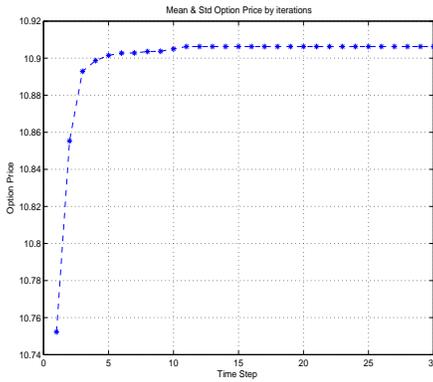
where  $r$  is the riskless rate of interest assumed to be constant,  $\sigma$  is the constant volatility, and  $\mathcal{W}_t$  is a standard Brownian motion process. No closed-form solution for the price is known. However, there are a few numerical methods giving good approximations.



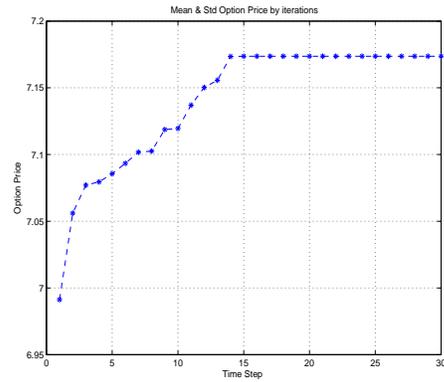
$$S_0 = 80$$



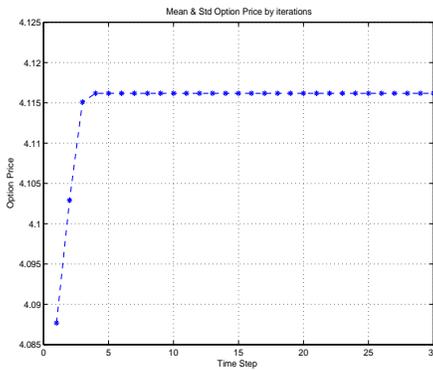
$$S_0 = 90$$



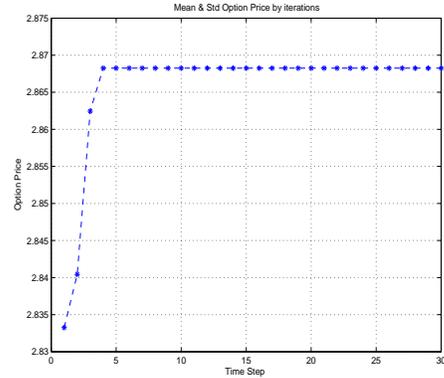
$$S_0 = 100$$



$$S_0 = 110$$



$$S_0 = 120$$



$$S_0 = 130$$

**Figure 5:** Standard put price convergence by updating the free boundary with different initial stock prices. The  $X$ -axis represents the iteration number and the  $Y$ -axis represents option prices. Parameters are:  $K = 100$ ,  $r = 0.05$ ,  $T = 0.5$ , and  $\sigma = 0.4$

To find an optimal boundary, we generate a uniform  $(50, K)$  random variable for the initial boundary and update the distribution 10 times with a set of sample paths having  $\Delta t = T/20$ . The set of sample stock prices is of size 2000 and the number of policies is 100. From this, we obtain one

boundary. After that, we repeat the procedure 10 times, pick the 3 best policies, and then average them out for early exercise boundaries.

With the early excise boundary, we used 500000 sample paths with 200 exercise opportunities for the low and high biased estimators. The calculations were performed throughout in Matlab and can be expected to improve further by coding with a compiled language. CPU time is recorded only for obtaining the early exercise boundary at this time.

The results of our simulation are presented in Table 5. Parameter values are  $r = 0.06$ ,  $\sigma = 0.4$ ,  $K = 100$ , and  $T = 0.5$ , with varying initial stock prices as shown in the table. The column of true American prices is quoted from Ait-Sahalia and Carr [1]. Then we give the low biased estimate and high biased estimate from our method. Point estimates are obtained by averaging the low and high estimates. Error was calculated with point estimates and true values; a minus sign in the Error(%) column denotes the undervaluation of prices compared to the true price. CPU time is the average of 10 runs. Based on our algorithm, the point estimates are very close to the true prices for all cases. All errors are less than 0.25%.

**Table 5:** Simulation prices of high biased and low biased standard American puts. Option parameters:  $K = 100$ ,  $r = 0.06$ ,  $T = 0.5$ , and  $\sigma = 0.4$ .

$S_0$	Lower	Upper	Point	TRUE	Error(%)	CPU (s)
80	21.56	21.76	21.66	21.61	0.24	3.97
90	14.81	15.00	14.91	14.92	-0.09	4.55
100	9.87	10.02	9.94	9.95	-0.03	4.66
110	6.38	6.48	6.43	6.44	-0.11	4.32
120	4.02	4.10	4.06	4.06	-0.02	4.57

### 3.2.3.2 Example 2: American min put options on 2 assets

This example concerns American put options on two independent assets following GBM, so that we can compare our results with a similar example from Rogers (2001) given by

$$S_i(t) = S_i(0) \exp \left\{ \left( r - \frac{\sigma_i^2}{2} \right) t + \sigma_i W_i(t) \right\},$$

with payoff function

$$f(t) = \max_{i=1,2} e^{-rt} (K - S_i(t))^+.$$

The payoff function can be rewritten as

$$f(t) = e^{-rt} \left( K - \min_{i=1,2} S_i(t) \right)^+,$$

and we can find the early exercise boundary for the two assets by treating the problem as a single asset pricing problem,

$$S(t) = \min \{S_1(t), S_2(t)\}. \quad (27)$$

With this change, we calculate the high biased and low biased estimates even though the stopping rule should ideally depend on the values of each of the underlying stocks. Hartley [37] shows that numerical results obtained by using Equation (27) are within 1% of the finite-difference values in the case of two assets. If we compute 3-dimensional (3D) boundaries  $(S_1, S_2, t)$ , then we can have better estimates. However, it takes more effort to find boundaries. Hence, we simply attempt to transfer from the 3D to 2D setting as shown in Equation (27); numerical results are given in Tables 6 and 7.

**Table 6:** Simulation prices on min-puts on two assets with the same volatility by changing from 3D to 2D. Option parameters:  $K = 100$ ,  $r = 0.06$ ,  $T = 0.5$ , and  $\sigma_1 = \sigma_2 = 0.6$ .

$S_1$	$S_2$	Euro	FD	Lower	Upper	Point	Error(%)	CPU(s)
80	80	36.86	37.30	36.95	37.51	37.23	-0.19	6.84
80	100	31.64	32.08	31.75	32.39	32.07	-0.03	6.52
80	120	28.65	29.14	28.87	29.38	29.12	-0.06	6.29
100	100	24.73	25.06	24.82	25.93	25.37	-0.01	6.65
100	120	20.61	20.91	20.67	21.19	20.93	0.09	6.56
120	120	15.70	15.92	15.81	16.09	15.95	0.19	6.49

We test two different settings: symmetric and asymmetric. In Table 6, we have symmetric initial stock prices and compare our algorithm to finite-difference methods quoted from Hartley [37]. We ran our simulation with 100 policies and 2000 sample paths for 15 iterations to find the free boundaries. After obtaining the boundaries, we ran 500000 additional paths with 100 exercise opportunities. The results are very good compared to those of the finite-difference methods. Our error is less than 0.2% for all cases when compared to the finite-difference methods.

Table 7 presents the results obtained for asymmetric settings, with two different volatilities. We again compare our estimates to a those of finite-difference methods. We find interesting trends from

our results. Estimates with lower initial stock prices and a low volatility are always underestimated compared to those from a finite-difference method. On the other hand, estimates with higher initial stock prices and a low volatility are overestimated. Our estimation errors are less than 0.65% for all cases.

Again, the stopping rule should ideally depend on the two underlying processes. However, we also attempt to solve this problem by looking at the minimum of the two underlying prices and the difference of the two underlyings. With an appropriate exercise rule, we have higher estimates compared to the finite-difference method quoted from Hartley. Hartley shows in the case of two assets that his approximate method delivers numerical results within 1% of the finite-difference method; our estimates are always higher than the finite-difference method from Hartley, and not more than 1.1% for all cases. We could not directly compare our results to the finite-difference method for the 3D case since it was not provided by Rogers. We did not attempt to duplicate his algorithm at this time. Tables 18 and 19 in the appendix show the numerical results.

**Table 7:** Simulation prices on min-put options on two assets with different volatilities by reducing from 3D to 2D. Option parameters:  $K = 100$ ,  $r = 0.06$ ,  $T = 0.5$ , and  $\sigma_1 = 0.4$ ,  $\sigma_2 = 0.8$ .

$S_1$	$S_2$	Euro	FD	Lower	Upper	Point	Error(%)	CPU(s)
80	80	37.55	38.01	37.71	38.35	38.03	0.05	6.51
80	100	31.81	32.23	31.92	32.42	32.17	-0.18	6.38
80	120	28.09	28.54	28.16	28.56	28.36	-0.64	6.22
100	80	32.86	33.34	33.13	33.67	33.40	0.19	6.58
100	100	25.47	25.81	25.63	26.17	25.90	0.34	6.45
100	120	20.48	20.75	20.58	20.93	20.76	0.03	6.19
120	80	30.69	31.21	31.13	31.66	31.39	0.58	6.10
120	100	22.44	22.77	22.63	23.42	23.02	0.51	6.59
120	120	16.76	16.98	16.80	17.23	17.02	0.21	6.70

### 3.2.3.3 Example 3: Bermudan max calls with $n$ assets

The Bermudan  $n$ -max-call had been studied in Broadie and Glasserman (BG) [8], Rogers [53], and Haugh and Kogan [38]. The Bermudan option has similar properties as American options but with a limited number of exercise opportunities. In this example, we price the following option whose

payoff function is

$$f(t) = e^{-rt} \left( \max_{i=1, \dots, n} S_i(t) - K \right)^+.$$

As usual, the assets  $S_i(t)$  are GBM's which are independent and identically distributed for this example. There is a continuous dividend payout at rate  $\delta$ . The stock price processes for only two assets under the risk-neutral measure are assumed to follow a correlated geometric Brownian motion process. For example,

$$\begin{aligned} \frac{dS_1(t)}{S_1(t)} &= (r - \delta_1) dt + \sigma_1 dW_t^1 \\ \frac{dS_2(t)}{S_2(t)} &= (r - \delta_2) dt + \rho\sigma_2 dW_t^1 + \sqrt{1 - \rho^2}\sigma_2 dW_t^2. \end{aligned}$$

We compute an early exercise boundary with 100 policies and 2000 sample paths and repeat this process 10 times. We pick the three best boundaries and average them out for the early exercise boundary. CPU time is recorded Table 8. Once we obtain the early exercise boundary, we can compute the low and high biased estimators. To estimate the high biased estimator, we need a little more effort. First, we generate the sample paths with  $N = 3, 6, 9$  and apply the exercise rules for low biased estimators. However, we need to generate forward stock prices for each exercise opportunity. For example, let  $t_1, t_2, t_3$  be possible exercise points and  $\Delta t = T/N = t_2 - t_1$ . Then we have  $S_{t_0}, S_{t_1}, S_{t_2}, S_{t_3}$ . But we generate prices from each possible exercise point. That is,  $S_{t_0+\delta t}, S_{t_1+\delta t}, S_{t_2+\delta t}$ . Therefore, if  $t_2$  were the optimal stopping time, then the low biased value is  $f(S_{t_2})$  and the high biased value would be  $F(S_{t_2+\delta t})$ .

We ran 200000 sample paths to estimate the prices. For the high biased estimator, we used  $\delta t = T/1000$ . Since they do not have a closed-form solution, we just compare our estimates to those of the other algorithms. Our upper estimate is always higher than the BG method and lower than Rogers' method; the point estimates are very close to those of the BG method.

### 3.2.3.4 Example 4: American-Bermudan-Asian option

This example concerning an American-Bermudan-Asian option has been studied by Longstaff and Schwartz [45] and Rogers [53]. The model is a single GBM and uses the cumulative average

$$A_t = \frac{A_0\delta + \int_0^t S_u du}{t + \delta}, \quad t \geq 0. \quad (28)$$

**Table 8:** Simulation prices of a max call option on 5 assets:  $K = 100$ ,  $r = 0.05$ ,  $\delta = 0.1$  months,  $T = 3$  yrs, and  $\sigma_i = 0.2$ . The option can be exercised at any of times  $t = iT/d$ ,  $i = 0, 1, \dots, d$ , where  $d = 3, 6, 9$ . CPU-I is for computing the early exercise boundary, CPU-II is for the low and high biased estimator, and CPU-III is quoted from Rogers. CPU-I and CPU-II are computed by Matlab with a P4-2.4Ghz machine and CPU-III is by Scilab with a 600Mhz PC. For Rogers' computation, 1000 sample paths were used for the optimization step and 8000 paths to refine the estimate. All CPU units are seconds. Rogers' CPU times are included reference purposes only.

$d$	$S_0$	BG	Rogers	Lower	Upper	Point	CPU-I	CPU-II	CPU-III
3	90	16.01	16.24	15.97	16.09	16.03	4.94	4.25	337.73
	100	25.28	25.70	25.24	25.47	25.35	4.80	4.25	227.47
	110	35.70	36.19	35.58	35.87	35.72	4.83	4.14	208.07
6	90	16.47	16.91	16.45	16.68	16.56	6.43	6.59	299.63
	100	25.92	26.40	25.77	26.07	25.92	6.42	6.62	329.52
	110	36.50	37.18	36.35	36.74	36.55	6.41	6.69	345.98
9	90	16.66	16.98	16.51	16.75	16.63	8.13	9.04	710.71
	100	26.16	26.75	25.98	26.34	26.16	8.06	8.97	419.56
	110	36.78	37.61	36.63	37.05	36.84	7.98	8.99	431.86

The positive value of  $\delta$  prevents fluctuations near  $t = 0$ . Also, there is a lockout period  $t^*$  during which the option cannot be exercised. But the option can be exercised at any time between  $t^*$  and  $T$  and the payoff is  $(A_t - K)^+$ . The price of this option is

$$C_0 = \sup_{t \in [t^*, T]} E_q[\alpha(A_t - K)^+]. \quad (29)$$

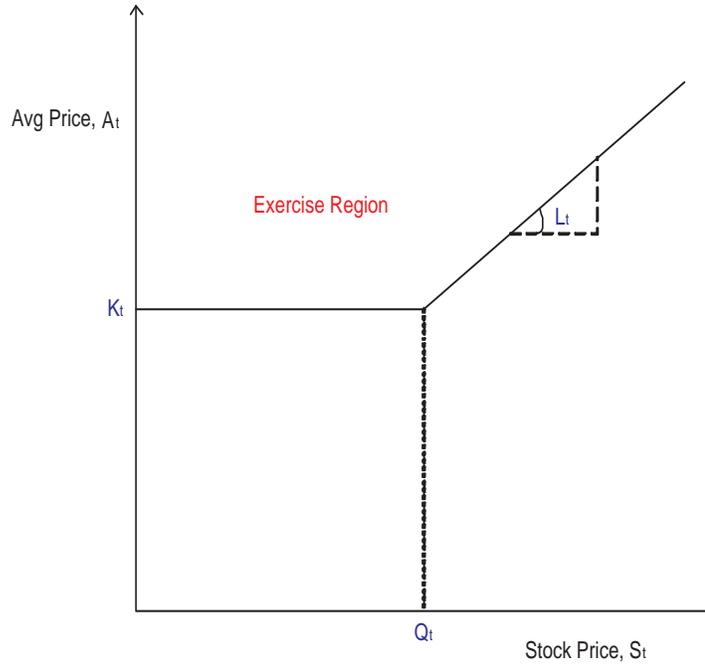
Rogers points out that Longstaff and Schwartz use 100 discretization points per year to approximate the continuous exercise feature of the option. Therefore, it may be more accurate to say that Longstaff and Schwartz are pricing a Bermudan-Asian option.

We shall estimate the values for the American-Bermudan-Asian option where there is no restricted exercise between  $t^*$  and  $T$ . We compare our estimates to the Longstaff-Schwartz figures quoted from [53] for the finite-difference value of the options (based on a discretization using 10000 time steps per year, and 200 space steps in each of two dimensions) and Rogers' upper boundary of estimation.

For this option, we need to pay attention to two things. First, there would never be an exercise at a time when  $A_t \leq K$ ; and second, there would never be an exercise at a time when

$$G_t \equiv e^{-rt} \left[ \frac{S_t - A_t}{t + \delta} - r(A_t - K) \right] > 0,$$

since  $G$  is the derivative of the payoff at a fixed time point with respect to  $t$ ; and it is clear that if the exercise value is increasing, then the optimal exercise requires the holder to wait to exercise since the value will assuredly rise in the next small instant of time.



**Figure 6:** Approximated early exercise boundary for American-Bermudan-Asian option at time  $t$ .

We cannot directly apply the policy iteration at this frame since at each time we need to consider the stock price,  $S_t$ , and average price of the stock,  $A_t$ . Figure 6 shows the shape of the boundary at fixed time  $t$ , where  $0 < t < T$ . In this figure, there are three parameters,  $K_t$ ,  $Q_t$ , and  $L_t$  which we need to find to maximize the value of the option. Therefore, it is a much more-complicated problem as compared to our previous problems. However, we can simplify this matter using the boundary property of an American-Asian option.

First, we know that the early exercise boundary at maturity is at the strike price,  $K_T = K$ . Since  $K_t$  is a monotonic decreasing function to  $K$ , we can use a linear function to approximate  $K_t$ . That is,

$$K_t = K_0 - \left( \frac{K_0 - K}{T} \right) t, \quad t \in (0, T).$$

Here, we try to find  $K_0$ .

In addition, the “tilt point” at the maturity time,  $Q_T$ , is infinite or the slope is zero. However,

we set the approximate initial and end tilt points,  $Q_0$  and  $Q_T$ , and approximate  $Q_t$  for each time  $t$  since it is a monotonic increasing function,

$$Q_t = Q_0 + \left( \frac{Q_T - Q_0}{T} \right) t, \quad t \in (0, T).$$

For this, we need to find two values,  $Q_0$  and  $Q_T$ , using policy iteration. First we attempt to fix  $Q_T$  to find the optimal  $Q_0$ , and then later we relax this assumption.

Lastly, the slope after the tilt point is zero at maturity. Again, we assume that the slope is a monotonic decreasing function to zero. That is,

$$L_t = L_0 \left( 1 - \frac{t}{T} \right), \quad t \in (0, T),$$

and the early exercise boundary at time  $t$  can be described as a piecewise linear function,

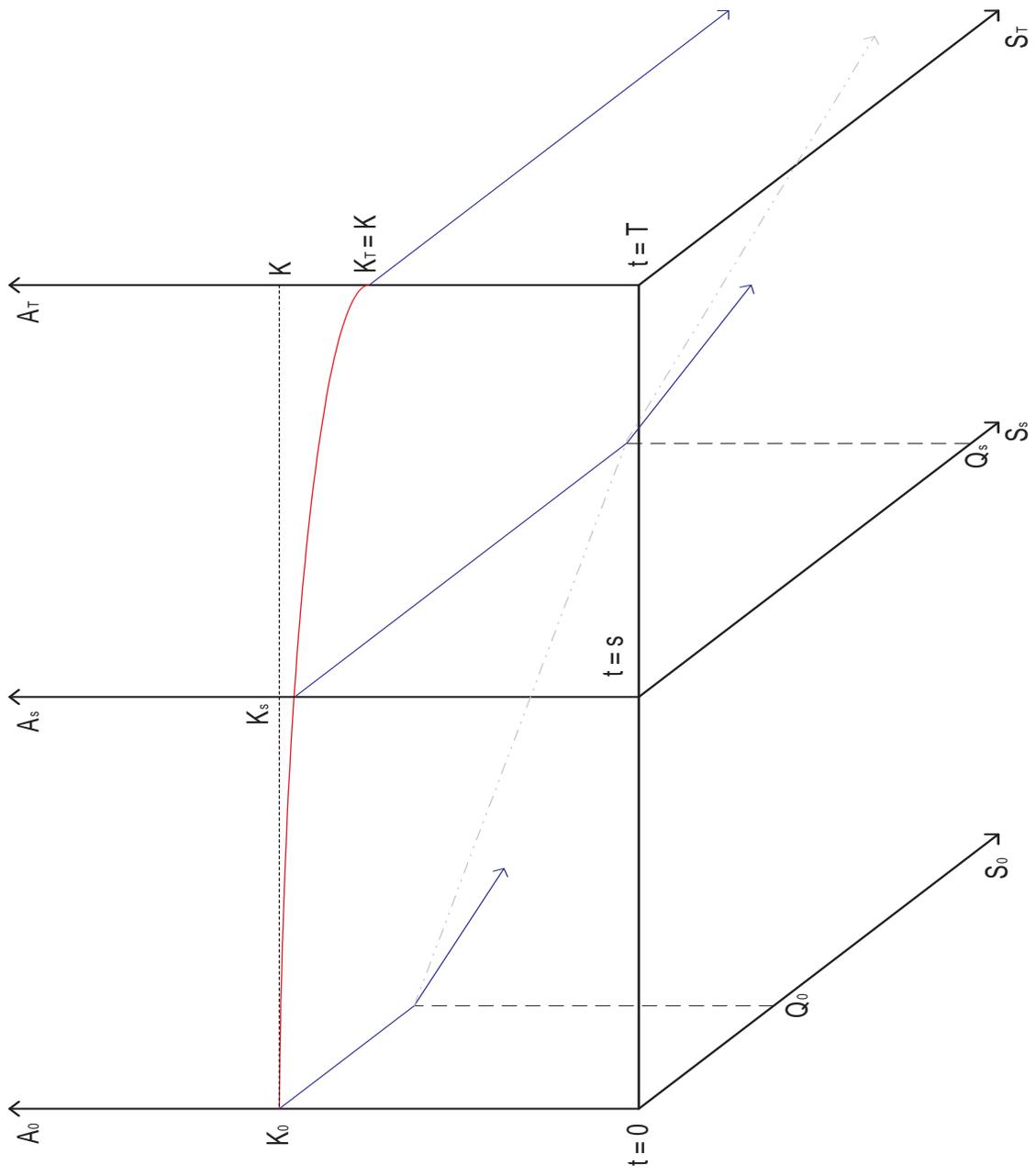
$$B(S_t) = \begin{cases} K_t & \text{for } S_t < Q_t \\ K_t + L_t(S_t - Q_t) & \text{for } S_t > Q_t. \end{cases} \quad (30)$$

Hence, we need to find three values  $K_0$ ,  $Q_0$ , and  $L_0$  using our algorithm. This simple structure enhances our search algorithm very efficiently. Figure 7 shows the overall shape of the boundary.

We modify slightly the algorithm given in Section 3.2 to adjust to the structure of this product. That is,

1. Generate  $M$  initial triples,  $K_0, Q_0, L_0$  from a uniform distribution. Let these values be  $Q_i^{(1)}(0) = (K_0, Q_0, L_0)_i$ , where  $i = 1, \dots, M$ .
2. From the initial triple, calculate  $K_t, Q_t, L_t$  from the above equations.
3. Generate one set of stock prices,  $S_n(t)$  following GBM, and calculate  $A_n(t)$  from  $S_n(t)$ .
4. Apply Steps 7 through 11 from Section 3.2.

Table 9 compares estimates to the finite-difference method (FD), Rogers' duality approach, and the least squares method (LSM). Both finite-difference and least squares methods use 200 time steps for stock prices and average prices, and Rogers uses 40 time steps. We use 200 time steps to compare our algorithm to FD and LSM.



**Figure 7:** Approximated boundary for American-Bermudan-Asian option.

**Table 9:** Simulation prices of American-Bermudan-Asian option:  $K = 100$ ,  $r = 0.06$ ,  $t^* = \delta = 3$  months,  $T = 2$  years, and  $\sigma = 0.2$ . There are 200 exercise opportunities.

$A_0$	$S_0$	Lower	FD	Rogers	LSM	Error(%)
90	80	0.955	0.949	0.952	0.961	0.632
90	90	3.311	3.267	3.297	3.309	0.060
90	100	7.898	7.889	7.892	7.886	0.114
90	110	14.471	14.538	14.575	14.518	-0.461
90	120	22.273	22.423	22.513	22.378	-0.669
100	80	1.107	1.108	1.094	1.101	-0.090
100	90	3.710	3.710	3.697	3.700	0.000
100	100	8.638	8.658	8.752	8.669	-0.231
100	110	15.581	15.717	15.913	15.703	-0.865
100	120	23.735	23.811	23.924	23.775	-0.319
110	80	1.295	1.288	1.265	1.265	0.543
110	90	4.163	4.136	4.409	4.186	0.652
110	100	9.751	9.821	10.359	9.830	-0.713
110	110	17.231	17.399	17.684	17.362	-0.966
110	120	25.256	25.453	25.661	25.406	-0.774

To find the optimal boundary, we use 100 sample policies and 1000 stock prices. After obtaining the boundary, we run another 200000 simulations. At this time, we simply compute the low biased estimate since  $G_t$  already contains the future price information, and it does not provide further information as long as the time increment for the high biased estimator is larger than the time increment for price generation ( $\delta t > \Delta t$ ). Hence, we did not attempt to compute the high estimator.

Our estimation errors are calculated with a finite-difference method. In most cases, our errors are less than 1% compared to the finite-difference method and LSM. In addition, our estimates compared to LSM are higher in some cases and lower in some cases.

### 3.2.3.5 Example 5: American put options on a single asset with jump-diffusion processes

In this section, we describe how our algorithm can be applied to pricing American put options with a jump-diffusion underlying process. We compare our estimates to the LSM method to check the robustness of our algorithm. First, we assume the underlying security consists of two parts: a continuous part and a discontinuous part. The continuous part is a geometric Brownian motion with constant instantaneous drift  $\mu$  and volatility  $\sigma$ . The discontinuous part represents the change in the security value upon arrival of some rare event. Rare events include a major disaster or political

changes, or the release of unexpected firm or economic news. The process is

$$\frac{dS_t}{S_{t^-}} = \mu dt + \sigma dW_t + V_t dq_t, \quad (31)$$

where first two terms on the right-hand side represent the continuous part, the third term represents the discontinuous part, and  $q = (q_t)_{t \geq 0}$  is a homogeneous Poisson process with intensity parameter  $\lambda \geq 0$  per year.

Let  $T_1, T_2, \dots$ , denote the arrival times of the jumps. Moreover, let  $V_{T_i} = (S_{T_i^+}/S_{T_i^-}) - 1$ , where  $V_{T_1}, V_{T_2}, \dots$  are i.i.d. random variables representing the successive percentage changes in the security value at the jump events. We assume that  $W, q$ , and  $V_{T_i}$  are jointly independent.

Let  $A_i$  be the logarithm of the ratio of the security value after and before the jump. We assume it is normally distributed and state independent. That is,

$$A_i = \ln S_{T_i^+} - \ln S_{T_i^-} = \ln(V_{T_i} + 1),$$

where  $f(A_i) \sim N(\mu_A, \sigma_A^2)$ .

Let  $L = \sum_{i=1}^m A_i$  be the sum of the log jump-sizes in the interval  $[0, T]$ . Under this assumption,  $L$  is also normally distributed,

$$L \sim N(\mu_A \lambda T, (\mu_A^2 + \sigma_A^2) \lambda T)$$

by Wald's equation ([54]).

Applying the Doléans-Dade stochastic exponential formula for semimartingales, we get a unique solution to the stochastic differential equation,

$$S_T = S_0 \exp \left\{ \left( r - \frac{\sigma^2}{2} \right) T + \sigma W_T^{\mathcal{Q}} + L^{\mathcal{Q}} - \lambda^{\mathcal{Q}} E_{\mathcal{Q}}(V_{T_i}) T \right\}, \quad (32)$$

where  $\mathcal{Q}$  is the equivalent martingale measure that makes the security process a local martingale.

To simplify the illustration, we use the jump-to-ruin model presented in Merton [48]. In this model, the stock price follows a geometric Brownian motion until a Poisson event occurs at which point the stock price becomes zero. The dynamics for this jump-diffusion process are given by

$$dS = (r + \lambda)S dt + \sigma S dW_t - S dq_t.$$

When a Poisson event occurs, the value of  $q$  jumps from zero to one and the stock price jumps downward from  $S$  to zero.

By solving the stochastic differential equation (SDE), we have in the discrete time setting,

$$S_{t+\Delta t} = \begin{cases} (1 - \zeta)S_t \exp \left\{ \left( r + \lambda - \frac{\sigma^2}{2} \right) \Delta t + \sigma \sqrt{\Delta t} Z \right\} & \text{if jumps occur} \\ S_t \exp \left\{ \left( r + \lambda - \frac{\sigma^2}{2} \right) \Delta t + \sigma \sqrt{\Delta t} Z \right\} & \text{otherwise,} \end{cases} \quad (33)$$

where  $\zeta$  is the magnitude of jumps and  $\zeta = 1$  for the jump-to-ruin model.

To make a comparison to LSM, we simply followed the setting of the Longstaff and Schwartz paper. That is, we compare the price for the American put option for the cases where there is (i) no possibility of a jump  $\lambda = 0.0$  and (ii) when a jump can occur with intensity  $\lambda = 0.05$ . If  $\lambda > 0$ , then the distribution of stock price is no longer conditionally lognormal. Furthermore, the conditional variance of the stock price increases as  $\lambda$  increases.

To make an even more meaningful comparison, we match the first and second moments for two different underlying processes. For example, to equalize the variance between the with-jump and without-jump cases, we adjust the parameters  $\lambda$  and  $\sigma$  such that  $\lambda + \sigma^2$  is the same for all different cases, since the variance of stock prices is

$$S^2 \exp(2r) \left\{ \exp \left( (\lambda + \sigma^2)T \right) - 1 \right\},$$

where  $S$  is the initial stock price.

To find an optimal boundary, we generate a uniform  $(0, K)$  random variate for the initial boundary and iterate 15 times with a set of sample paths having  $\Delta t = T/20$ . We generate a set of 2000 sample stock prices and 100 policies. From this, we obtain one boundary.

With an early exercise boundary, we used the 20000 sample paths with 50 exercise opportunities for low and high biased estimators to compare our algorithm to LSM. CPU-I is only for obtaining the early exercise boundary, CPU-II is for calculating the low and high biased estimators, and CPU-III is for the LSM algorithm. The computational cost is more expensive than without jump processes to obtain the early exercise boundary; however, the difference in computing time for estimating the low biased price is insignificant.

Table 10 shows the prices of an American put with jumps and without jumps. We compare directly our algorithm to LSM for the at-the-money option case since LSM estimation is only directly available for that case. But we compute other cases such as in-the-money and out-of-the-money options even though LSM estimates are not available in [45]. Therefore, we ran the simulation using

**Table 10:** Simulation prices of high biased and low biased standard American puts with jump-diffusion process. All computations are done on a P4 2.4Ghz PC with Matlab. CPU time units are in seconds. Option parameters:  $K = 100$ ,  $r = 0.06$ , and  $T = 1$ .

$S_1$	$\lambda, \sigma$	LSM	Lower	Upper	Point	CPU-I	CPU-II	CPU-III
44	0, 0.3	2.53	2.45	2.55	2.50	6.34	1.89	26.64
	0.05, 0.2	2.60	2.60	2.63	2.61	6.41	1.78	18.13
42	0, 0.3	3.11	3.02	3.16	3.09	6.25	1.86	35.14
	0.05, 0.2	2.91	2.88	2.92	2.90	6.32	1.85	24.89
40	0, 0.3	3.84	3.78	3.94	3.86	6.20	1.84	53.72
	0.05, 0.2	3.40	3.39	3.46	3.43	6.34	1.81	38.95
38	0, 0.3	4.73	4.64	4.82	4.73	6.20	1.92	91.91
	0.05, 0.2	4.09	4.04	4.13	4.08	6.27	1.84	85.44
36	0, 0.3	5.69	5.64	5.84	5.74	5.96	1.84	143.97
	0.05, 0.2	4.90	4.96	5.10	5.03	6.01	1.83	165.61

the LSM algorithm to compare the least squares method to our estimates. We use 50 exercise points per year in the LSM algorithm with 20000 replications.

The values of the options are lower when there is a possibility of a jump, holding fixed the variance across the example but in the deep out-of-the-money case ( $S = 44$ ). This makes intuitive sense because for the diffusion coefficient in the  $\lambda = 0.05$  case, there is a very low possibility of jumps. That is why jumps do not affect the gain of the option holders for at-the-money and in-the-money options. However, if the option is deep out-of-the-money, gains for the option holder are higher even though  $\lambda$  is small because only one jump can dramatically change its payoff.

### 3.2.4 Conclusion

This chapter presents a generic algorithm for pricing American-style financial options by an adaptive simulation. This method first estimates the early exercise frontiers and then evaluates the options with the estimated policies through Monte Carlo simulation. High biased and low biased estimators are computed and a point estimator is obtained yielding the two biased estimates. Also, the adaptive algorithm leads to the convergence to the true early exercise frontier with probability one. Therefore, we provide an efficient simulation-based algorithm for pricing American-style options.

We presented several examples: a put option, min puts with two assets, Bermudan max calls with five assets, American-Bermudan-Asian options, and American put options with jump diffusion

processes. Compared with other simulation estimators, our algorithm is very competitive in terms of computing time as well as performance properties of its estimates.

We have already shown that the difference between the estimated optimal value and the true value goes to zero (w.p.1) under the appropriate assumptions. Hence, we can further investigate confidence intervals based on the estimators. The error bounds for the high biased estimator need to be investigated more since they used future information in the paths and such information determines the distance between the high biased estimator and the true value.

### ***3.3 Variance Reduction***

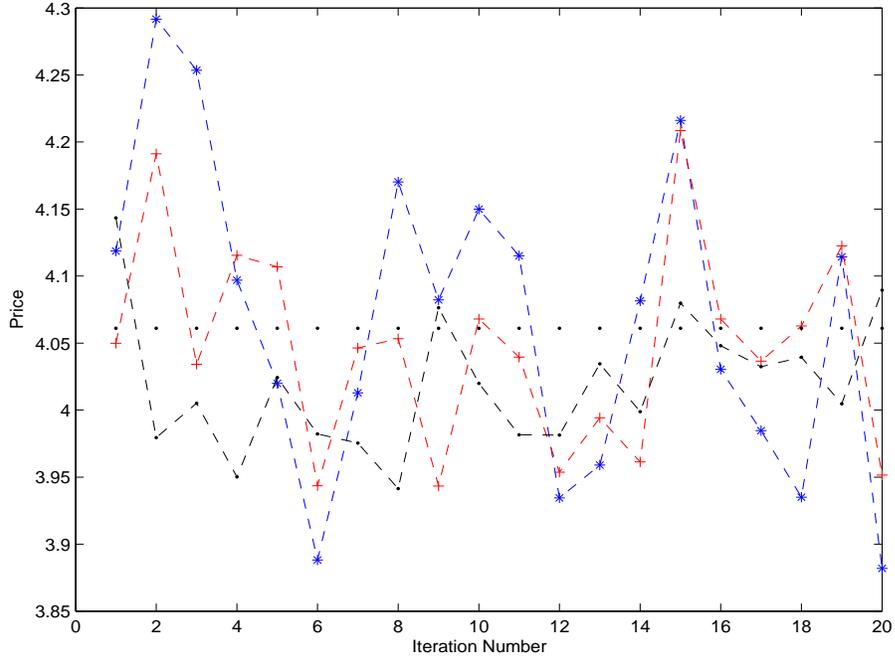
The computational burden can be reduced significantly using variance reduction techniques (VRTs). This section develops methods for increasing the efficiency of Monte Carlo simulation by reducing the variance of simulation estimates. The greatest gains in efficiency from variance reduction techniques result from exploiting specific features of a problem rather than from a generic application of a generic method. Therefore, we need to investigate each case and choose a specific VRT for the purpose at hand. In this section, we go over a few well-known techniques such as antithetic variates, control variates, importance sampling, and stratified sampling and its applications.

Figure 8 shows a sample run with three different estimates, without the use of a variance reduction technique, with Antithetic Variates (AV), and with Control Variates (CV). The detailed descriptions of these techniques are given in the next section. This figure shows the variance reduction visually.

As we expected, the variances of our estimators are reduced significantly when we use AV and CV. This shows that these two different techniques can be well implemented in our adaptive algorithm and as a consequence, we can reduce the computational cost. Note that AV has better performance than CV in our example here.

#### **3.3.1 Antithetic Variates (AV)**

This method is very simple and easy to implement compared to other variance reduction techniques. The central idea is to make pairs of runs of the model such that a small observation on one of the runs in a pair tends to be offset by a large observation on the other one; i.e., the two observations



**Figure 8:** Price estimation with three different methods. The dotted line is the true price obtained by the finite-difference method, the dashed asterisk is without any variance reduction technique, the dashed plus sign is with AV, and the dashed point is with CV.

are negatively correlated.

In its simplest form, AV tries to induce negative correlation by using complementary random numbers to derive the two runs in a pair. That is, if  $U_k$  is a particular uniform (0,1) random number used for a particular purpose in the first run, we use  $1 - U_k$  for this same purpose in the second run. It is perfectly valid to use  $1 - U_k$  instead of simply an independent direct draw from the random number generator since  $U \sim U(0, 1)$  implies that  $1 - U \sim U(0, 1)$  as well.

An important point is that the use of  $U_k$  in one replication and its complement  $1 - U_k$  in the paired replication must be synchronized, i.e., used for the same purpose; the benefit of Antithetic Variates could be otherwise lost or even perhaps could even backfire in the form of a variance increase.

There is a mathematical basis for AV. Suppose that we make  $n$  pairs of runs of the simulation, resulting in observations  $(S_1^{(1)}, S_1^{(2)}), (S_2^{(1)}, S_2^{(2)}), \dots, (S_n^{(1)}, S_n^{(2)})$ , where  $S_j^{(1)}$  is from the first run of the  $j$ th pair and  $S_j^{(2)}$  is from the antithetic run of the  $j$ th pair. Both  $S_j^{(1)}$  and  $S_j^{(2)}$  are legitimate stock prices of the simulation model so that  $E[S_j^{(1)}] = E[S_j^{(2)}] = \mu = \text{true mean}$ . Also, each pair

is independent of every other pair; i.e., for  $j_1 \neq j_2$ ,  $S_{j_1}^{(l_1)}$  and  $S_{j_2}^{(l_2)}$  are independent regardless of whether  $l_1$  and  $l_2$  are equal. For  $j = 1, \dots, n$ , let  $S_j = (S_j^{(1)} + S_j^{(2)})/2$ , and let the average of the  $S_j$ 's,  $\bar{S}(n)$ , be the unbiased point estimator of  $\mu$ . Then since the  $S_j$ 's are i.i.d., we have

$$\text{Var}(\bar{S}(n)) = \frac{\text{Var}(S_j)}{n} = \frac{\text{Var}(S_j^{(1)}) + \text{Var}(S_j^{(2)}) + 2\text{Cov}(S_j^{(1)}, S_j^{(2)})}{4n} = \frac{\text{Var}(S_j^{(1)}) + \text{Cov}(S_j^{(1)}, S_j^{(2)})}{2n}$$

If the two runs within a pair were made independently, then  $\text{Cov}(S_j^{(1)}, S_j^{(2)}) = 0$ . On the other hand, if we could indeed induce negative correlation between  $S_j^{(1)}$  and  $S_j^{(2)}$ , then  $\text{Cov}(S_j^{(1)}, S_j^{(2)}) < 0$ , which reduces  $\text{Var}(\bar{S}(n))$ .

In many examples, AV has been shown analytically to lead to variance reductions although the magnitude of the reduction is not necessarily known. Also note that the total number of replications is  $2n$  and thus, we need to run  $2n$  replications without AV to make a fair comparison of the magnitude of variance reduction.

Tables 11 and 12 compare the variance of each estimation technique. In Table 11, none of the variance reduction techniques are used when searching for the early exercising policy. However, we used AV in Table 12 to find the early exercising policy. The parameter values are  $r = 0.06$ ,  $\sigma = 0.4$ ,  $K = 100$ , and  $T = 0.5$  with varying initial prices as shown in the table. Standard deviation is only computed from low biased estimators. We ran 5000 sample paths for AV and 10000 for the non-variance reduction technique and repeated the entire exercise 20 times. For all cases, we have a smaller variances as we expected.

### 3.3.2 Control Variates (CV)

This method is among the most effective and broadly applicable techniques for improving the efficiency of Monte Carlo simulation. In principle, at least, there is an appealing intuition to CV. Let  $X$  be an output random variable, such as the average of the 100 sample stock prices for a standard American call option, and assume that we want to estimate  $C = E[X]$ . Suppose that  $Y$  is another random variable involved in the simulation that is thought to be correlated with  $X$  (either positively or negatively), and that we know the value of  $c = E[Y]$ . For instance,  $c$  could be an European call option price. It is reasonable to expect that a bigger than average European call price tends to lead to a bigger than average American call price and vice versa. Thus, if we run a simulation and notice that  $Y > c$ , we might suspect that  $X$  is above its expectation  $C$  as well and accordingly adjust  $X$

**Table 11:** Simulation prices of low biased standard American puts with Antithetic Variates under GBM. Option parameters:  $K = 100$ ,  $r = 0.06$ ,  $\sigma = 0.4$ , and  $T = 0.5$ . None of the variance reduction techniques are used in obtaining the optimal early exercising policy.

$S_0$	Regular			AV			CV			TRUE
	Low	High	Point	Low	High	Point	Low	High	Point	
80	21.56	22.01	21.78 (0.13)	21.55	21.79	21.67 (0.07)	21.57	22.02	21.80 (0.11)	21.61
90	14.80	15.13	14.96 (0.14)	14.82	15.01	14.91 (0.05)	14.80	15.14	14.97 (0.06)	14.92
100	9.95	10.22	10.08 (0.12)	9.95	10.07	10.01 (0.06)	9.94	10.21	10.08 (0.05)	9.95
110	6.45	6.66	6.56 (0.08)	6.42	6.52	6.47 (0.06)	6.44	6.65	6.55 (0.05)	6.44
120	4.05	4.17	4.11 (0.08)	4.05	4.11	4.08 (0.05)	4.06	4.18	4.12 (0.04)	4.06

**Table 12:** Simulation prices of low biased standard American puts with Control Variates under GBM. Option parameters:  $K = 100$ ,  $r = 0.06$ ,  $\sigma = 0.4$ , and  $T = 0.5$ . AV is used to find the optimal early exercising policy.

$S_0$	Regular			AV			CV			TRUE
	Low	High	Point	Low	High	Point	Low	High	Point	
80	21.49	21.91	21.70 (0.12)	21.54	21.80	21.67 (0.08)	21.56	21.99	21.77 (0.12)	21.61
90	14.79	15.13	14.96 (0.14)	14.81	15.04	14.93 (0.05)	14.80	15.14	14.97 (0.06)	14.92
100	9.94	10.18	10.06 (0.11)	9.90	10.00	9.95 (0.06)	9.89	10.13	10.01 (0.05)	9.95
110	6.42	6.57	6.50 (0.08)	6.41	6.49	6.45 (0.06)	6.43	6.59	6.51 (0.04)	6.44
120	4.03	4.14	4.09 (0.09)	4.05	4.10	4.07 (0.08)	4.04	4.15	4.10 (0.04)	4.06

downward by some amount since we are interested in estimating  $C$ . We call  $Y$  a control variate for  $X$  since it is used to adjust  $X$ , or partially control it.

To carry out the above idea, we must quantify the amount of the upward or downward adjustment to  $X$ . It is convenient to express this amount in terms of the deviation  $Y - c$ , of  $Y$  from its expectation. Let  $\alpha$  be a constant that has the same sign as the correlation between  $X$  and  $Y$ . We use  $\alpha$  to scale to the deviation  $Y - c$  to arrive at an adjustment to  $X$  and thus define the controlled estimator

$$X_c = X - \alpha(Y - c).$$

Note that if  $Y$  and  $X$  are positively correlated, so that  $\alpha > 0$ , we adjust  $X$  downward whenever  $Y > c$ , and upward if  $Y < c$ .

Since  $E[X] = C$  and  $E[Y] = c$ , it is clear that for any real number  $\alpha$ ,  $E[X_c] = C$ ; that is,  $X_c$  is an unbiased estimator of  $C$  and it is consistent because, with probability 1,

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n X_c^{(i)} &= \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \left( X^{(i)} - \alpha(Y^{(i)} - c) \right) \\ &= E[X - \alpha(Y - c)] \\ &= E[X]. \end{aligned}$$

Also it might have a lower variance than  $X$ . Specifically,

$$\text{Var}(X_c) = \text{Var}(X) + \alpha^2 \text{Var}(Y) - 2\alpha \text{Cov}(X, Y), \quad (34)$$

so that  $X_c$  is less variable than  $X$  if and only if

$$2\alpha \text{Cov}(X, Y) > \alpha^2 \text{Var}(Y),$$

which may or may not be true, depending on the choice of  $Y$  and  $\alpha$ .

To find a “best” value of  $\alpha$  for a given  $Y$ , we can view the right-hand side of Equation (34) as a function  $f(\alpha)$  of  $\alpha$  and set its derivative to zero; i.e.,

$$\frac{df}{d\alpha} = 2\alpha \text{Var}(Y) - 2 \text{Cov}(X, Y) = 0,$$

and solve for the optimal (variance-minimizing) value

$$\alpha^* = \frac{\text{Cov}(X, Y)}{\text{Var}(Y)}. \quad (35)$$

Note that the second derivative  $d^2f/d\alpha^2 = 2\text{Var}(Y)$  is positive, a sufficient condition for  $\alpha^*$  to be a minimizer of  $f(\alpha)$ .

Plugging  $\alpha^*$  from Equation (35) into the right-hand side of Equation (34), we get that the minimum variance adjusted estimator  $X_c^*$  over all choices of  $\alpha$  has variance

$$\text{Var}(X_c^*) = \text{Var}(X) - \frac{\text{Cov}(X, Y)^2}{\text{Var}(Y)} = (1 - \rho_{XY}^2) \text{Var}(X),$$

where  $\rho_{XY}$  is the correlation between  $X$  and  $Y$ . Thus, using the optimal value  $\alpha^*$  for  $\alpha$ , the optimally controlled estimator  $X_c^*$  can never be more variable than the uncontrolled  $X$  and will in

fact have lower variance if  $Y$  is at all correlated with  $X$ . Moreover, the stronger the correlation between  $X$  and  $Y$ , the greater the variance reduction; in the extreme, as  $\rho_{XY} \rightarrow \pm 1$ , we see in fact that  $\text{Var}(X_c^*) \rightarrow 0$ . Intuitively, this says that if the correlation between  $X$  and  $Y$  were nearly perfect, we can control  $X$  almost exactly to  $C$  every time, thereby eliminating practically all of its variance.

In practice, CV is not easy to implement directly. Depending on the source and nature of the control variate  $Y$ , we may or may not know the value of  $\text{Var}(Y)$ , and we will certainly not know  $\text{Cov}(X, Y)$  exactly, making it impossible to find the exact value of  $\alpha^*$  from simulation runs; therefore, we need to estimate them. One of the simplest methods is to replace  $\text{Cov}(X, Y)$  and  $\text{Var}(Y)$  in Equation (35) by their sample estimators. Suppose that we make  $n$  independent replications to obtain  $n$  i.i.d. observations  $X_1, X_2, \dots, X_n$  on  $X$  and  $n$  i.i.d. observations  $Y_1, Y_2, \dots, Y_n$  on  $Y$ . Let  $\bar{X}(n)$  and  $\bar{Y}(n)$  be the sample means of the  $X_j$ 's and  $Y_j$ 's, respectively, and let  $S_Y^2(n)$  be the unbiased sample variance of the  $Y_j$ 's. The covariance between  $X$  and  $Y$  is estimated by

$$\hat{C}_{XY}(n) = \frac{\sum_{j=1}^n (X_j - \bar{X}(n))(Y_j - \bar{Y}(n))}{n - 1}$$

and the estimator for  $\alpha^*$  is then

$$\hat{\alpha}^*(n) = \frac{\hat{C}_{XY}(n)}{S_Y^2(n)}.$$

Putting everything together, we arrive at the final point estimator for  $C$ ,

$$\bar{X}_C^*(n) = \bar{X}(n) - \hat{\alpha}^*(n) (\bar{Y}(n) - c).$$

We should note that since the constant  $\alpha^*$  has been replaced by the random variable  $\hat{\alpha}^*(n)$ , which is generally dependent on  $\bar{Y}(n)$ , we cannot take expectations across the factors in the second term of  $\bar{X}_C^*(n)$ . Unfortunately, then  $\bar{X}_C^*(n)$  — unlike  $X_c$  and  $X_c^*$  — will in general be biased for  $C$ .

Tables 11 and 12 compare the variance of each estimator with the CV technique. The parameter values are  $r = 0.06$ ,  $\sigma = 0.4$ ,  $K = 100$ , and  $T = 0.5$  with varying initial prices as shown in the tables. We simply compared the lower estimates at this time. We ran 10000 sample paths for the non-variance reduction technique and CV, and we repeated the entire exercise 20 times. For all cases, we have smaller variances than those of without variance reduction techniques.

### 3.3.3 Importance Sampling (IS)

Importance sampling attempts to reduce the variance by changing the probability measure from which paths are generated. In importance sampling, we change the measures to try to give more weight to important outcomes. To make this idea tangible, consider

$$I = E[h(X)] = \int h(x)f(x)dx,$$

where  $X$  is a random variable with probability density  $f$  and  $h$  is a function from  $\mathcal{R}$  to  $\mathcal{R}$ . Then the Monte Carlo estimator is

$$\hat{I} = \hat{I}(n) = \frac{1}{n} \sum_{i=1}^n h(X_i),$$

where  $X_1, \dots, X_n$  are independent draws from  $f$ . Let us define a new probability density function  $g$  on  $\mathcal{R}$  satisfying

$$f(x) > 0 \implies g(x) > 0,$$

for all  $x \in \mathcal{R}$ . Then we can redefine  $I$  as

$$I = \int h(x) \frac{f(x)}{g(x)} g(x) dx.$$

This integral can be interpreted as an expectation with respect to the density  $g$ , which we write as

$$I = E_{\mathcal{G}} \left[ h(X) \frac{f(X)}{g(X)} \right], \tag{36}$$

where  $E_{\mathcal{G}}$  is the expectation taken with  $X$  distributed according to  $g$ . Then the Monte Carlo estimator with  $g$  is

$$\hat{I}_g = \hat{I}_g(n) = \frac{1}{n} \sum_{i=1}^n h(X_i) \frac{f(X_i)}{g(X_i)}.$$

The weight  $f(X_i)/g(X_i)$  is the *likelihood ratio* or *Radon-Nikodym* derivative evaluated at  $X_i$ .

It follows from Equation (36) that  $E_{\mathcal{G}}[\hat{I}_g] = I$  and thus that  $\hat{I}_g$  is an unbiased estimator of  $I$ . To compare the variances with and without IS, it suffices to compare second moments. With IS, we have

$$E_{\mathcal{G}} \left[ \left( h(X) \frac{f(X)}{g(X)} \right)^2 \right] = E \left[ h(X)^2 \frac{f(X)}{g(X)} \right].$$

This could be larger or smaller than the second moment  $E[h(X)^2]$  without IS; indeed, depending on the choice of  $g$  it might even be infinitely larger or smaller. Successful importance sampling lies

in the art of selecting an effective importance sampling density  $g$ . For instance, if

$$g(x) \propto h(x)f(x), \quad (37)$$

then  $h(X_i)f(X_i)/g(X_i)$  equals the constant of proportionality in Equation (37) regardless of the value of  $X_i$ ; thus, the importance sampling estimator in Equation (36) provides a zero variance estimator in this case. Of course, one cannot achieve this great variance reduction every time because one cannot usually use such a nice  $g(x)$ .

### 3.3.4 Stratified Sampling

Stratified sampling refers broadly to any sampling mechanism that constrains the fraction of observations drawn from specific subsets (strata) of the sample space. Suppose our goal is to estimate  $E[X]$  with  $X$  real-valued, and let  $A_1, \dots, A_N$  be disjoint subsets of the real line for which  $P(X \in \bigcup_{i=1}^N A_i) = 1$ . Then

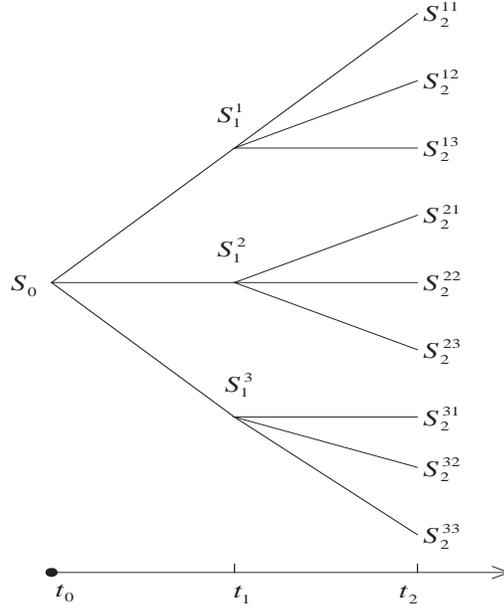
$$E[X] = \sum_{i=1}^N P(X \in A_i)E[X|X \in A_i] = \sum_{i=1}^N p_i E[X|X \in A_i],$$

where  $p_i = P(X \in A_i)$ . We first decide what fraction of samples should be drawn from each stratum  $A_i$ ; and each observation drawn from  $A_i$  is constrained to have the distribution of  $X$  conditional on  $X \in A_i$ .

The simplest case is proportional sampling, in which we ensure that the fraction of observations drawn from stratum  $A_i$  matches the theoretical probability  $p_i = P(X \in A_i)$ . If the total sample size is  $n$ , this entails generating  $n_i = np_i$  samples from  $A_i$ . Let  $X_{ij}, j = 1, \dots, n_i$  for each  $i = 1, \dots, N$ , be independent draws from the conditional distribution of  $X$  given  $X \in A_i$ . An unbiased estimator of  $E[X|X \in A_i]$  is provided by the sample mean of observations from the  $i$ th stratum. The unbiased estimator of  $E[X]$  is

$$\hat{X} = \sum_{i=1}^N p_i \frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij} = \frac{1}{n} \sum_{i=1}^N \sum_{j=1}^{n_i} X_{ij}.$$

This estimator should be contrasted with the usual sample mean  $\bar{X} = \sum_{i=1}^n X_i/n$  of  $X_1, \dots, X_n$ . Compared with  $\bar{X}$ , the stratified estimator  $\hat{X}$  eliminates sampling variability across strata without affecting sampling variability within strata.



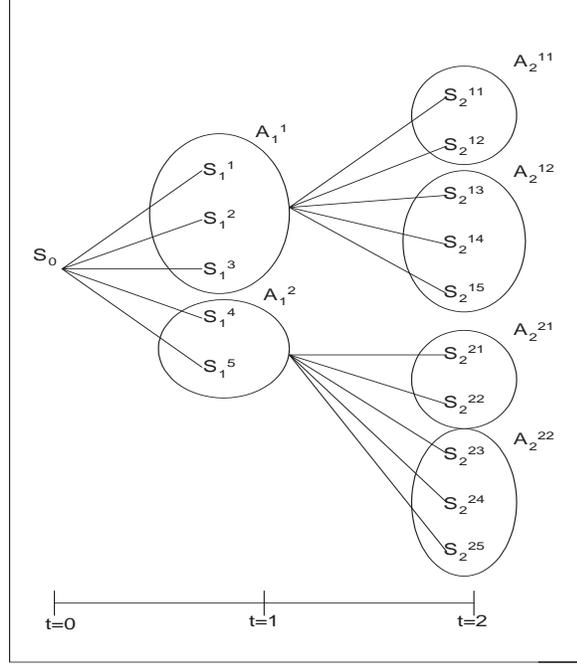
**Figure 9:** Simulated tree for  $b = 3$ .

We apply this technique to the simulated trees of Broadie and Glasserman [8]. The simulated trees are parameterized by  $b$ , the number of branches per node. State variables are simulated at a finite number of possible decision points, i.e., exercise times. Figure 9 with  $b = 3$  shows how to generate the simulated tree. For example,  $S_0$  is the initial stock price, and  $S_1^1, S_1^2,$  and  $S_1^3$  are stock prices generated from  $S_0$ . Again,  $S_2^{11}, S_2^{12},$  and  $S_2^{13}$  are generated from  $S_1^1$ . The prices at each node do not appear according to their node values as we see in the typical binomial lattice method. From the stochastic tree, Broadie and Glasserman obtained their high and low estimates.

We elaborate on the price generation process using the stratification idea. The reason we are using *stratification* is that we can reduce the variance of the option price while maintaining reality of the market process. As we know, the stock price process has the *martingale* property under the risk-neutral measure. That is,

$$E_{\mathcal{Q}}[S_t | \mathcal{F}_t] = S_{t-1}.$$

Under regular price generation, i.e., without strata, we simply generate the  $b$  branches. If  $b$  goes to infinity, then the price processes possess the martingale property. However, if  $b$  is large, then we have a computational problem since the tree grows exponentially. Therefore, we set  $b$  to be between 20 and 50 under regular price generation.



**Figure 10:** Stock tree with strata,  $N^{Str} = 5, n = 2, T = 2$ .

With the idea of stratifying, we can generate an almost infinite number of stock prices as long as we have a reasonable number of strata. That is, if we have  $20 < n (= \text{number of strata}) < 50$ , then it will take almost the same amount of time to run the simulation as compared to Broadie and Glasserman's tree method. However, as we mentioned above, if  $N^{strata}$  (= total sample size) is large, then the stock price processes have the martingale property by the law of the large numbers. Therefore, we apply the optimal exercise policy to the new stock prices tree.

Suppose that we have  $n = b$  strata and generate  $N^{strata} = m$  sample stock prices. Denote these processes as  $\Omega_{t_k} = \{S_{t_k}^1, S_{t_k}^2, \dots, S_{t_k}^m\}$  at each time step  $t_k$ , where  $0 < k \leq T$ . Also, we use equiprobable strata, for instance. Let  $A^i$  for  $i = 1, \dots, b$  be subsets of  $\Omega_{t_k}$  and  $\Pr(A^i) = 1/b$ . We stratify the sample prices to their own strata. That is,  $A^i \cap A^j = \emptyset$  if  $i \neq j$ .

$$\begin{aligned}
 &\{S^i \in A^1 \subset \Omega_{t_k} \text{ for some } i\} \\
 &\{S^i \in A^2 \subset \Omega_{t_k} \text{ for some } i\} \\
 &\vdots \\
 &\{S^i \in A^b \subset \Omega_{t_k} \text{ for some } i\}.
 \end{aligned}$$

Then we can calculate the sample mean of each subset,

$$\bar{A}^j = \frac{\sum_{i \in A^j} S^i}{|A^j|} \quad \text{for } j = 1, \dots, b.$$

When we generate the next time step prices, we use the  $\bar{A}^j$  as the initial prices. That is,

$$S_n = \bar{A}_{n-1}^j e^{(r-\sigma^2/2)\Delta t + \sigma W_{\Delta t}} \quad \text{for } j = 1, \dots, b.$$

Therefore, the simulated price structure looks exactly the same as that without the strata. With this framework, we can not only derive similar price estimates but also reduce the variance. At this time, we do not provide numerical results.

## CHAPTER IV

### INTRODUCTION

A *trading system* is a method of trading that uses objective entry and exit criteria based on parameters that have been validated by historical testing on quantifiable data. Trading the futures and equity markets using a trading system does many things that may help improve performance of earnings. A trading system provides the discipline to overcome the fear and greed that in many cases paralyzes a trader, which prevents him from making proper decisions. Each order is determined by a set of rules that does not deviate based on anything other than market action.

There are many securities that are highly correlated with each other, for example, correlations  $|\rho| > 0.9$ . The high correlation between two securities gives a good idea how one security's price movements ought to move along with its counter-security's price movements. Therefore, we can observe the process of the set of correlated securities and catch any extreme behavior from the constituents of the ensemble. Below we motivate this approach.

Making a profit consistently by trading individual stocks is difficult. The main component of the price signal is memoryless and overwhelmingly large. One could argue that it may in fact be easier to make profits by trading in accordance with the oscillating rhythmic component or the fluctuating elastic signal in the price differential of two or more stocks. To overcome the distorting influence of the large random market signal for a single security, we structure a combination of trading positions in a pair or group of similar stocks so that the random market signal components of all the stocks in the overall trading position are at least partially cancelled out. This leads to an oscillating and mean reverting price signal which people often try to model.

One of the early trading systems is based on the assumption that the spread itself is lognormally distributed ([67]). This assumption has the downside of excluding the possibility of negative spreads; its primary motivation is computational efficiency because it lets us use all the standard Black-Scholes formulas.

Making the lognormality assumption is not very realistic but the idea of modeling the spread

directly is tempting in itself. It lets us avoid the thorny problem of explicitly understanding the dependence structure (correlation).

In assuming that the spread follows a given process while the component prices follow another, one concern might be the potential for inherent inconsistency. For example, when individual prices follow geometric Brownian motions (GBM), it is impossible for the spread to follow a GBM. On the other hand, if we assume that the individual prices follow arithmetic Brownian motion, there is no consistency in assuming arithmetic Brownian motion for the spread. Furthermore, in general, even if there is some inconsistency in the assumptions about the underlying process, we are usually able to choose the parameters of the spread process that ensure an approximate matching of the distributions.

Karguine [42] tried to model spread process as an Ornstein-Uhlenbeck process, and he derived the optimal differentiable and threshold policies. However, even if we succeed in choosing a process for the spread (whether one factor or multifactor), using it for pricing and hedging the spread option is not usually feasible ([27]). The fundamental problem is that the hedging instruments are the individual underlying components. Correlation and leg volatilities can have a dramatic impact on hedges. Eydeland [27] showed that the difference between the components' "deltas" can widen and narrow significantly with changes in volatility or correlation. The alternative delta hedge with a constant ratio between the individual underlying contracts generated by the method that models spreads directly, will usually result in significant losses.

In addition, the statistical properties of the spread might be more stable than those of the individual underlies. In such a situation, the estimation of the model can be much easier and the hedging issues may assume lesser importance. We look at the stock prices from a statistical point of view (nonparametric). It is easy to see that stock price processes are correlated and do not have the martingale property when we see the price spreads themselves in the long run. However, percentage spreads seem to have a martingale property based on our empirical testing. For the rest of the thesis, we investigate this process from a statistical point of view, try to set a reasonable strategy, and explore arbitrage opportunities.

**PART II**  
**INTRA-MARKET STATISTICAL TRADING MODEL**

## CHAPTER V

### STATISTICAL TRADING MODEL

Sometimes stock prices appear to remain in a range for extended period of time. A good way to describe this situation is to define a moving range around the stock prices. Some people use an upper boundary and a lower boundary to define the range; the upper bound is calculated as a moving average of a chosen period plus 5% of the price, and the lower boundary is the moving average minus 5%. These boundaries have the drawback of being too narrow to accommodate price levels when volatility is high and too wide when volatility is low.

Bollinger [11] defines the upper boundary as a chosen moving average plus twice the corresponding standard deviation, with the lower boundary as the moving average minus twice the standard deviation. Our approach for setting up the arbitrage trading model is very similar to Bollinger's approach. The main difference, however, is that we use two different highly correlated securities and we apply statistical process control methods.

#### *5.1 Modeling*

We can claim that the equity values are dependent on two factors. One corresponds to common macroeconomic conditions and the other is firm-specific conditions. If the macroeconomic conditions are changed, then the equity prices move in the same direction if they are under the same economic conditions. However, firm-specific condition changes influence an equity's own price movement and do not affect other companies' values. By continuously monitoring the macroeconomic conditions influencing the equity values, we can catch any odd behavior, i.e., the prices of two related assets do not react the same way even though they may eventually converge to the same direction later. Then we may find an arbitrage opportunity for those cases by selling high and buying low or buying low and selling high.

Let  $V$  be an equity value. Then we can describe its behavior as

$$dV = \xi dE + \zeta dF$$

where  $E$  is due to common macroeconomic conditions,  $F$  is due to firm-specific conditions, and  $\xi$  and  $\zeta$  are magnitudes of changes. Once we can decompose the cause of value changes, we can utilize the following equations,

$$\begin{aligned}dV^{(1)} &= \xi_1 dE + \zeta_1 dF^{(1)} \\dV^{(2)} &= \xi_2 dE + \zeta_2 dF^{(2)},\end{aligned}$$

where  $V^{(1)}$  and  $V^{(2)}$  are the values of assets 1 and 2 respectively, and  $F^{(1)}$  and  $F^{(2)}$  are firm-specific conditions for assets 1 and 2. Then the difference of two assets,  $V = V^{(1)} - V^{(2)}$ , can be rewritten as

$$dV = (\xi_1 - \xi_2) dE + \zeta_1 dF^{(1)} - \zeta_2 dF^{(2)}. \quad (38)$$

Since  $(\xi_1 - \xi_2)$  is a deterministic value as well as a major indicator of changing equity values in the long run, we can easily monitor whether two stock prices are converging or diverging. In addition, if  $\xi_1$  and  $\xi_2$  are very close each other, then the first term in the right-hand side of Equation (38) is negligible and we can identify arbitrage opportunities. Note that  $\text{Corr}(F^{(i)}, F^{(j)}) = 0$  if  $i$  and  $j$  are different assets.

To concretize our main idea, we analyze two highly correlated securities and catch their deviation from normal behavior (against macroeconomic condition changes). If we utilize this deviation, we may be able to get a good idea how to trade those securities. First, we use a 10-day *moving average* (MA(10)) of two highly correlated stocks' closing prices. Let  $X_n$  and  $Y_n$  be the  $n$ th day closing prices of stocks  $X$  and  $Y$ , respectively, and let  $D_n$  be the difference between the two underlying stocks' closing prices, i.e.,  $D_n = X_n - Y_n$ . Then we set up a threshold to establish appropriate trading criteria.

The main calculation follows. For simplicity, we assume that the price processes  $\{X_n\}$  and  $\{Y_n\}$  are stationary even though an empirical test may fail to show that. Section 5.2 discusses how to avoid the stationarity assumption by using *percentage spreads*.

Before we use the moving average, we need to specify lower moments of the price difference process. That is, the expected value of the difference process  $D$  is

$$E[D] = E[X - Y] = E[X] - E[Y], \quad (39)$$

and the variance of  $D$  is

$$\text{Var}(D) = \text{Var}(X) + \text{Var}(Y) - 2\rho\sigma_X\sigma_Y, \quad (40)$$

where  $\rho$  is a correlation coefficient between two stocks. The variances of  $X$  and  $Y$  are obtained by the following method under the GBM assumption for the underlying process. Let  $U_i = \ln(X_i) - \ln(X_{i-1})$ ,

$$S_X^2 = (N - 1)^{-1} \sum_{i=1}^N (U_i - \bar{U})^2,$$

and

$$\hat{\sigma}_X^2 = S_X^2 / \Delta t, \quad (41)$$

where  $\bar{U}$  is the sample mean of stock  $X$  and  $\Delta t$  is a time increment for stock price observation. We use  $\Delta t = 1/252$  in this paper since we observe the daily closing price and there are 252 trading days for one year. In practice, the implied volatility can be used instead of Equation (41). Even though we can use the implied volatilities, there may still exist an estimation error on the correlation coefficient,  $\rho$ . In addition, the assumptions on the underlying process are not always correct, e.g., the GBM process may not be appropriate although estimation errors are negligible. Therefore, we directly calculate the variance of the  $\{D_i\}$  process from the data.

First, we fit an MA( $q$ ) model of the  $\{D_i\}$  process. That is,

$$Z_{q+i} = \frac{1}{q} \sum_{j=1}^q D_{i+j} \quad \text{for } 0 \leq i \leq N - q. \quad (42)$$

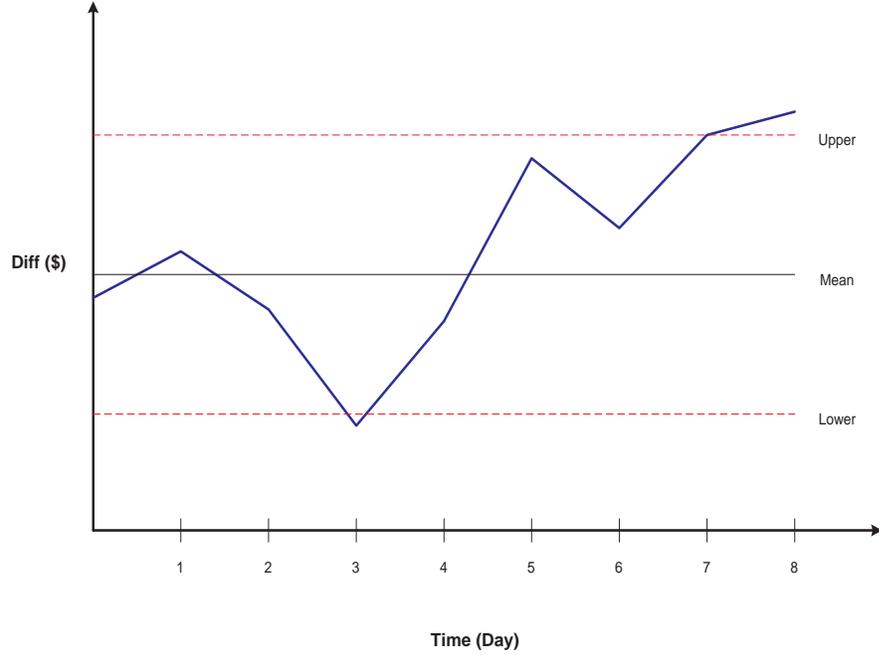
Then we calculate the mean and variance of the  $\{Z_i\}$  process. Since the  $D_i$ 's are serially correlated, we need to be careful when computing the variance. We again assume that the  $\{Z_i\}$  process is stationary. The mean is estimated by

$$\widehat{\mathbb{E}[Z]} = \bar{Z} = \frac{1}{N - q + 1} \sum_{j=0}^{N-q+1} Z_{q+j}, \quad (43)$$

and the variance of  $Z$  is estimated via the following set of equations.

- Empirical Estimation

$$\begin{aligned} \hat{\sigma}_{N-q+1}^2 &= (N - q + 1) \text{Var}(\bar{Z}) \\ &= R_Z(0) + \frac{2}{N - q + 1} \sum_{k=1}^{N-q} (N - q + 1 - k) R_Z(k), \end{aligned} \quad (44)$$



**Figure 11:** A sample path of a price difference process. Days 3 and 7 are out of bounds. We need to take an action on these events.

where

$$R_Z(k) = \frac{1}{q^2} \left( (q - k)R_D(0) + 2 \sum_{j=1}^{q-1-k} (q - j)R_D(j) \right),$$

and  $R_D(k)$  can be estimated by

$$\hat{R}_D(k) = \frac{1}{N - k} \sum_{j=1}^{N-k} (D_j - \bar{D})(D_{j+k} - \bar{D}). \quad (45)$$

Using the usual market convention, we assume that  $\text{Cov}(D_i, D_{i+k}) = 0$  if  $k \geq q$ .

Since we know the mean and variance of the  $\{Z_i\}$  process, we can use control charting ideas to detect extreme events. In practice, market technicians typically use  $\pm 1.5\sigma$  limits; we will also use that threshold for the initial stage of our investing strategy. To maximize the total profit, however, we need to commit additional study on the relationship between the severity of correlations and the gains and losses.

Figure 11 is a sample path of the  $D$  process. Since days 3 and 7 are out of the threshold, we may need to take an action depending upon the pattern of the underlying securities. For example, at day 3, we could short security  $Y$  and long security  $X$  since we assume that securities are going to

regress toward their mean points. On the other hand, at day 7, we could short security  $X$  and long security  $Y$ .

We now analyze further the frequency of these events and the magnitude of each event. We assume that the events that affect gains and losses form a renewal process and the gains and losses are independent of the times of arrival of the events. The total profit is the summation of the each gain or loss on each arrival. Let  $N(t)$  be the total number of events that are out of bounds,  $N_1(t)$  be the number of events that are out of the upper bound, and  $N_2(t)$  be the number of events that are out of the lower bound. Then  $N(t) = N_1(t) + N_2(t)$ . Let  $\phi_n^{(1)}$  be the gain or loss on the  $n$ th out-of-upper-bound event and  $\phi_n^{(2)}$  be the gain or loss on the  $n$ th out-of-lower-bound event. Also, we assume that  $\{\phi_n^{(i)}, n \geq 1, i = 1, 2\}$  are i.i.d. Hence, total gains or losses are

$$G(t) = \sum_{n=1}^{N_1(t)} \phi_n^{(1)} + \sum_{n=1}^{N_2(t)} \phi_n^{(2)}.$$

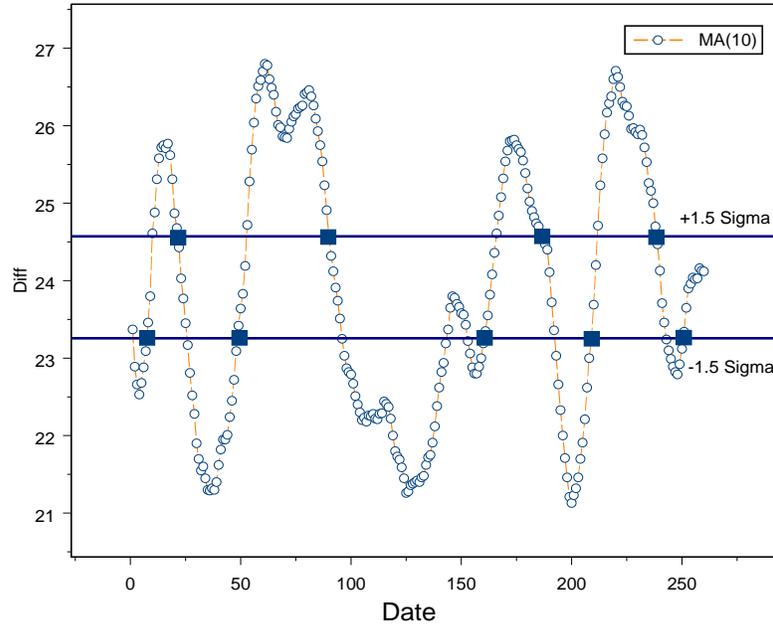
We also assume that  $\{N_i(t), i = 1, 2\}$  are renewal processes with inter arrival rates  $\lambda_i$ , and the corresponding gains/losses follow normal distributions with means  $\mu_i$  and standard deviations  $\sigma_i$ . Then we can calculate the mean and standard deviation of total gains/losses ([54]) under the independence assumption between  $\{N_i(t)\}$  and  $\{\phi^{(i)}\}$ . That is,

$$\begin{aligned} \mathbf{E}[G(t)] &= \mathbf{E}[N_1(t)]\mathbf{E}[\phi^{(1)}] + \mathbf{E}[N_2(t)]\mathbf{E}[\phi^{(2)}] \\ &= t(\lambda_1\mu_1 + \lambda_2\mu_2) \end{aligned}$$

and

$$\begin{aligned} \text{Var}[G(t)] &= \text{Var}\left(\sum_{n=1}^{N_1(t)} \phi_n^{(1)}\right) + \text{Var}\left(\sum_{n=1}^{N_2(t)} \phi_n^{(2)}\right) \\ &= \mathbf{E}[N_1(t)]\text{Var}[\phi^{(1)}] + \text{Var}[N_1(t)]\mathbf{E}[\phi^{(1)}]^2 \\ &\quad + \mathbf{E}[N_2(t)]\text{Var}[\phi^{(2)}] + \text{Var}[N_2(t)]\mathbf{E}[\phi^{(2)}]^2 \\ &= \lambda_1 t (\sigma_1^2 + \mu_1^2) + \lambda_2 t (\sigma_2^2 + \mu_2^2). \end{aligned}$$

In order to conduct a preliminary study, we applied this methodology to real market data. As in Figure 11, we first set the upper and lower thresholds with  $\mu \pm 1.5\sigma$ . However, the best threshold may be different depending on pairs and we may need to further analyze the situation to maximize the profit.



**Figure 12:** KO and PBG’s difference moving average, MA(10), and upper and lower thresholds. Date 1 is August 2, 2002 and Date 260 is July 5, 2001. The solid square boxes are the points out of the bounds where we may need to take a proper action. Table 13 shows overall gains and losses by trading corresponding to this graph.

### 5.1.1 Numerical Example

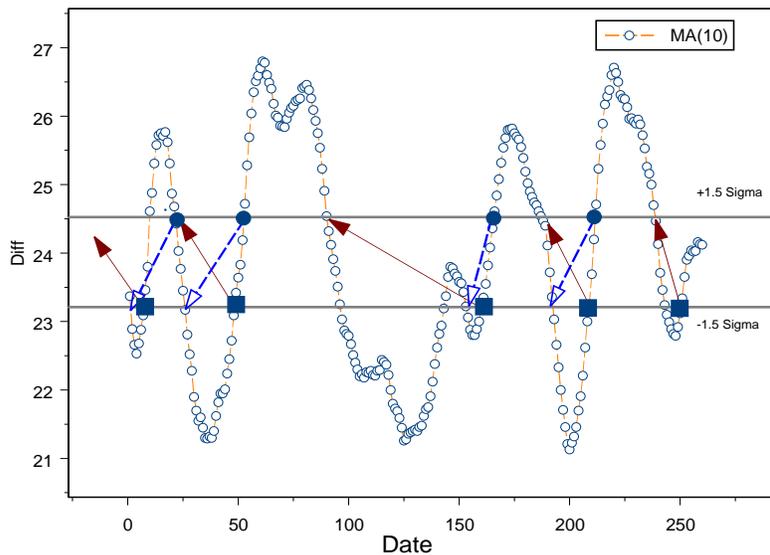
We select two highly correlated common stocks traded in the market actively, for example, the Coca-Cola Company and the Pepsi Bottling Group; or the Home Depot Inc. and Lowe’s Companies Inc., and so on. For this example, we use the Coca-Cola Company (NYSE:KO) and the Pepsi Bottling Group (NYSE:PBG). We collect the daily closing prices of the two stocks from July 5, 2001 to August 2, 2002. Based on these data, we estimate the mean and variance of the 10-day moving average difference process using Equations (43) and (44). The sample mean is 23.88 and the sample standard deviation is 0.4261. Figure 12 shows the overall 10-day moving average of the differences process. Based on the figure, we notice that several data points are out of the boundary.

We set a simple trading strategy. That is, if any  $Z_i$ ,  $1 \leq i \leq 260$ , are out of the upper bound, we short PBG and long KO. If the  $Z_i$  are out of the lower bound, we short KO and long PBG. The solid boxes represent these transaction points. We first long 2295 (= \$100000 / closing price of KO) shares of KO and short 4613 (= \$100000 / closing price of PBG) shares of PBG on August

**Table 13:** Trading summary based on Figure 12. D and U correspond to the square boxes on the graph. D is the first day of out of the lower threshold and U is the first day of out of the upper threshold, prices are daily closing prices, and the numbers of shares are rounded up to be integer. The numbers of shares are calculated by assuming that we have \$100,000.

	Date	KO(\$)	PBG(\$)	# KO	# PBG	Gain/Loss
D	25-Jul-02	47.56	25.27	-2103	3957	0
U	5-Jul-02	56.47	31.43	1837	-3087	6693
D	28-May-02	54.44	32.39	-1837	3087	0
U	27-Mar-02	52.10	26.03	2180	-4327	952
D	13-Dec-01	45.87	23.11	-2180	4327	0
U	5-Nov-01	48.73	23.77	2224	-4243	7513
D	5-Oct-01	44.97	23.57	-2224	4243	0
U	17-Aug-01	46.99	21.37	2295	-4613	9279
D	1-Aug-01	43.57	21.68	-2295	4613	0

1, 2001; then we short 2295 shares of KO and long the 4613 shares of PBG on August 17, 2001. These two transactions produced a profit of \$9280. In the same manner with all other possible trading cases, we made a total of \$24,437 during the period under study. We can also increase the



**Figure 13:** KO and PBG's difference moving average, MA(10), and upper and lower thresholds. Date 1 is August 2, 2002 and Date 260 is July 5, 2001. In Figure 12, we just had the solid boxes; the current graph includes solid circles to indicate the new trading dates.

**Table 14:** Trading summary based on Figure 13. D\* and U\* correspond to square boxes on the graph. D\* is the first day out of the lower threshold and U\* is the first day out of the upper threshold, D and U correspond to solid circles on the graph. D is the first day out of the lower bound and U is the first day returning from the upper bound. Prices are daily closing prices, and the numbers of shares are rounded up to an integer. The numbers of shares are calculated by assuming that we have \$100,000 for each trading method.

	Date	KO(\$)	PBG(\$)	# KO	# PBG	Gain/Loss
D	1-Aug-02	49.93	24.87	2179	-4219	3867
D*	25-Jul-02	47.56	25.27	-2103	3957	0
U	22-Jul-02	45.89	23.7	-2179	4219	0
U*	5-Jul-02	56.47	31.43	1837	-3087	6693
D	27-Jun-02	56.09	30.75	1786	-2910	10711
D*	28-May-02	54.44	32.39	-1837	3087	0
U	20-May-02	55.99	34.37	-1786	2910	0
U*	27-Mar-02	52.1	26.03	2180	-4327	952
D	24-Dec-01	47.99	23.63	2139	-4261	1971
D*	13-Dec-01	45.87	23.11	-2180	4327	0
U	5-Dec-01	46.75	23.47	-2139	4261	0
U*	5-Nov-01	48.73	23.77	2224	-4243	7513
D	26-Oct-01	48.57	23.91	2195	-4330	3083
D*	5-Oct-01	44.97	23.57	-2224	4243	0
U	1-Oct-01	45.56	22.99	-2195	4350	0
U*	17-Aug-01	46.99	21.37	2295	-4613	9279
D*	1-Aug-01	43.57	21.68	-2295	4613	0

number of transactions by trading when the moving average returns between the upper and lower thresholds. In fact, we use the same strategies for trading. That is, the last day of out of the upper threshold, we long KO and the last day out of the lower threshold, we short PBG. Figure 13 shows the dates of trading. The solid boxes are the same as those of Figure 12, and the solid circles have been added to indicate the new trading dates.

Table 13 shows the overall transaction process based on our first trading strategy corresponding to Figure 12. Table 15 shows the gain and loss (G&L) from the trading. The second trading strategy is used at this time. With this trading strategy, we made \$44069.

This works very well given the specific period. But do the mean and thresholds remain the same for a long time? The reasonable answer would be no. The next question is when and how to change the mean and thresholds. There are two approaches for detecting a small shift of mean — the cumulative sum (CUSUM) charts and exponentially weighted moving average (EWMA) charts.

**Table 15:** Trading summary based on Figure 12. D\* and U\* correspond to square boxes on the graph. D\* is the first day out of the lower threshold and U\* is the first day out of the upper threshold, D and U correspond to solid circles on the graph. D is the first day out of the lower bound and U is the first day returning from the upper bound. Prices are daily closing prices, and the numbers of shares are rounded up to an integer. The numbers of shares are calculated by assuming that we have \$100,000 for each trading method.

	Date	Intc(\$)	AMD(\$)	# INTC	# AMD	G/L
D*	29-Aug-01	27.98	14.2	-3573.98	7042.254	0
U*	19-Nov-01	30.88	13.65	3573.981	-7042.25	14237.82
U	31-Dec-01	31.34	15.86	-3190.81	6305.17	0
D	14-Jun-02	21.23	9.6	3190.81	-6305.17	7211.264

Also, the threshold depends on the variance of the process. That means that we need to monitor the variability of the process as well.

As stated above, a mean shift can be detected by either CUSUM or EWMA charts. We attempt to use EWMA because it is regarded as a nonparametric (distribution-free) procedure ([50]). The exponentially weighted moving average  $z_i$  is defined as

$$z_i = \lambda x_i + (1 - \lambda)z_{i-1},$$

where  $0 \leq \lambda \leq 1$  is a constant and the starting value is the process target, so that  $z_0 = \mu_0$ . We use the average of preliminary data as the starting value of the EWMA, so that  $z_0 = \bar{x}$ . If the observations  $x_i$  are independent random variables with variance  $\sigma^2$ , then the variance of  $z_i$  is

$$\sigma_{z_i}^2 = \sigma^2 \left( \frac{\lambda}{2 - \lambda} \right) [1 - (1 - \lambda)^{2i}].$$

Therefore, the EWMA control chart would be constructed by plotting  $z_i$  versus the sample number  $i$ . The center line and control limits for the EWMA control chart are as follows.

$$UCL = \mu_0 + L\sigma \sqrt{\frac{\lambda}{(2 - \lambda)} [1 - (1 - \lambda)^{2i}]} \quad (46)$$

$$\text{Center line} = \mu_0$$

$$LCL = \mu_0 - L\sigma \sqrt{\frac{\lambda}{(2 - \lambda)} [1 - (1 - \lambda)^{2i}]}, \quad (47)$$

where  $L$  is the width of the control limits (see [16]). Note that the term  $[1 - (1 - \lambda)^{2i}]$  in the above equation approaches unity as  $i$  gets larger. That means that after the EWMA control chart has been

running for several time periods, the control limits will approach the steady-state values given by

$$UCL = \mu_0 + L\sigma\sqrt{\frac{\lambda}{(2-\lambda)}} \quad (48)$$

$$LCL = \mu_0 - L\sigma\sqrt{\frac{\lambda}{(2-\lambda)}}. \quad (49)$$

There have been numerous extensions and variations of EWMA control charts. We can also use EWMA to monitor the process standard deviation. However, we did not attempt to implement this method since we used short trading periods and it unlikely to detect such changes of mean. If we use a long period of data, we recommend that a check be performed on whether the mean and/or standard deviation change.

## 5.2 SPC with Autocorrelated Data

So far, we have dealt with the price spread itself. By using a price spread, we face several problems. First, the spread is not a stationary process for many cases. Therefore, we need to change the mean and thresholds depending on changes of mean and standard deviation. That is a cumbersome problem with this approach, and it is not straightforward to detect such changes of mean and standard deviation. Now, we use the ratio changes of price between two securities rather than price spreads themselves. Using the same notation, let  $\tilde{X}_n = X_n/X_{n-1}$  and  $\tilde{Y}_n = Y_n/Y_{n-1}$ . Also, let  $D_n$  be the difference of two ratio changes, i.e.,  $D_n = \tilde{X}_n - \tilde{Y}_n$ . With these changes from the previous section, we now implement the statistical process control approach.

The standard assumptions that are usually cited in justifying the use of control charts are that the data generated by the process when it is in control are normally and independently distributed with mean  $\mu$  and standard deviation  $\sigma$ . Both  $\mu$  and  $\sigma$  are considered fixed and unknown. An out-of-control condition is a change or shift in  $\mu$  or  $\sigma$  (or both) to some different value. Therefore, we could say that when the process is in control the quality characteristic at time  $t$ ,  $x_t$ , is represented by the model

$$x_t = \mu + \epsilon_t, \quad t = 1, 2, \dots, \quad (50)$$

where  $\epsilon_t$  are i.i.d. normal with mean zero and standard deviation  $\sigma$ . This is often called the Shewhart model of the process. When these assumptions are satisfied, one may apply conventional control

charts and draw conclusions about the state of statistical control of the process. Furthermore, the statistical properties of the control chart, such as the false-alarm rate with  $3\sigma$  control limits, or the average run length, can be easily determined and used to provide guidance for chart interpretation. Even if the normality assumption is violated to a slight or moderate degree, these control charts will still work reasonably well.

Montgomery [50] argues that the most important assumption made concerning control charts is that of independence of the observations, for conventional control charts do not work well if the quality characteristic exhibits even low levels of correlation over time. Specifically, these control charts will give misleading results in the form of too many false alarms if the data are correlated.

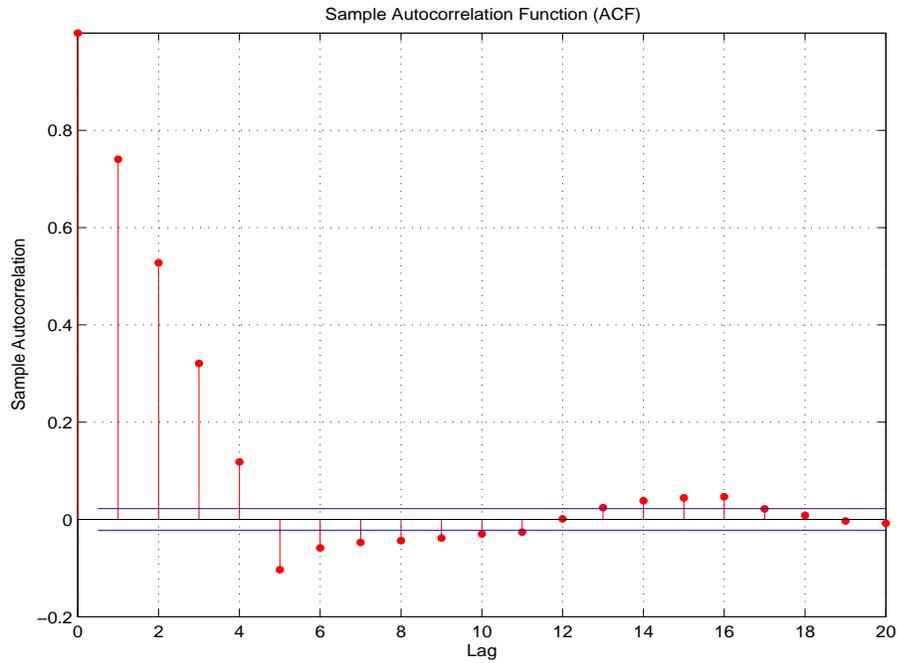
We have already mentioned that the stock prices are serially correlated over the time (see Figure 14). An approach that has proved useful in dealing with autocorrelated data is to directly model the correlation structure with an appropriate time series model, use that model to remove the autocorrelation from the data, and apply control charts to the residuals.

We collected the 1-minute tick data for KO and PBG for 21 trading days from 04/28/2004 to 05/26/2004. We first look at the autocorrelation function (ACF) of the differences from the raw data and check the normality of this price difference data. Figure 14 shows the ACF of KO-PBG and Figure 15 deals with the normality of the difference process. Figure 14 tells us that the difference process is highly autocorrelated and Figure 15 shows that the data is not normally distributed. Therefore, we can conclude from these two figures that we cannot directly apply classical SPC these data sets.

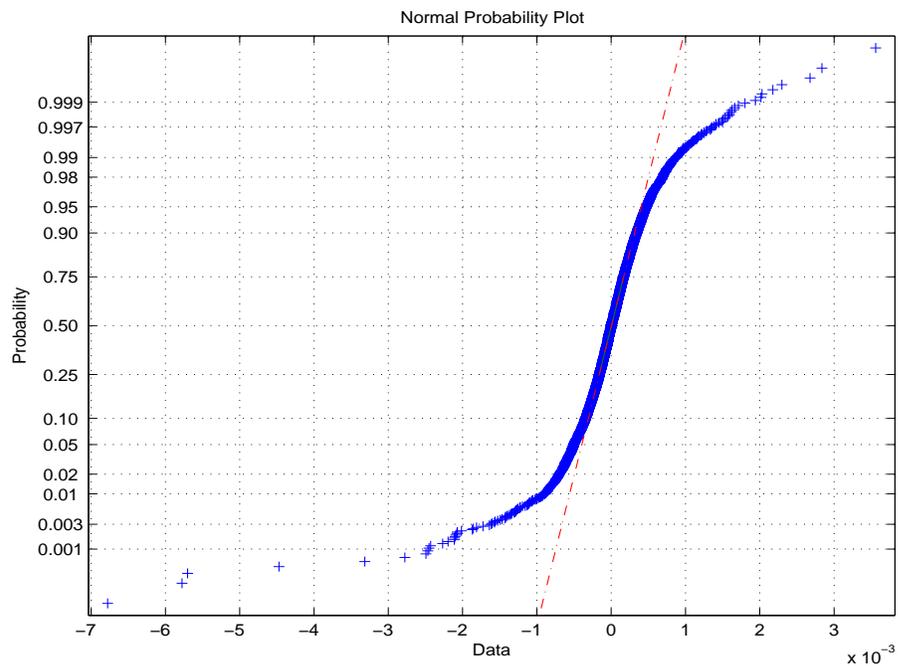
One way to cure this problem is to use the *batch means* approach proposed by Runger and Willemain [57]. The batch means approach has been used extensively in the analysis of the output from computer simulation models, another area where highly correlated data often occurs. This approach works because as the batch size becomes large, batch means become approximately i.i.d. normal. The unweighted batch means (UBM) chart breaks successive groups of sequential observations into batches, with equal weights assigned to every point in the batch. Let the  $j$ th unweighted batch mean be

$$\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{(j-1)m+i}, \quad j = 1, 2, \dots, b, \quad (51)$$

where  $m$  is the batch size and  $b$  is the number of batches. The important implication of Equation (51)



**Figure 14:** Autocorrelation plot for the KO-PBG difference process. Moving average length is 5 minutes.



**Figure 15:** Normality plot of KO-PBG difference process. Moving average length is 5 minutes.

is that although one has to determine an appropriate batch size  $m$ , it is not necessary to construct an ARMA model of the data.

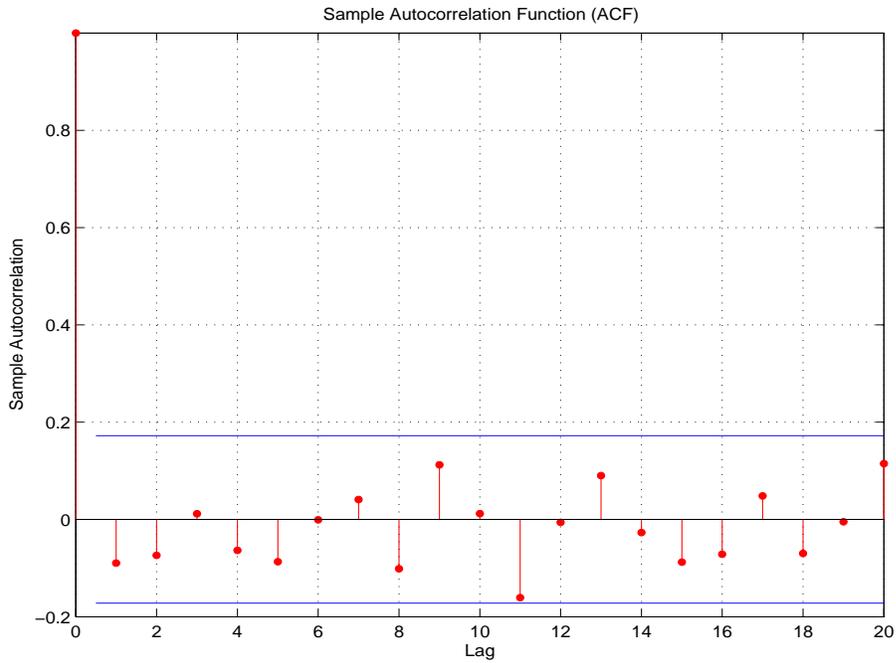
Runger and Willemain showed that the batch means can be plotted and analyzed on a standard individual chart. With UBM, the control chart averaging is used to dilute the autocorrelation of the data.

Procedures for determining an appropriate batch size have been developed in the simulation area. These procedures are empirical and do not depend on identifying and estimating a time series model. Of course, a time series model can guide the process of selecting the batch size and also provide analytical insights. Also, note that if batch size gets big, the batch means become approximately i.i.d. normal.

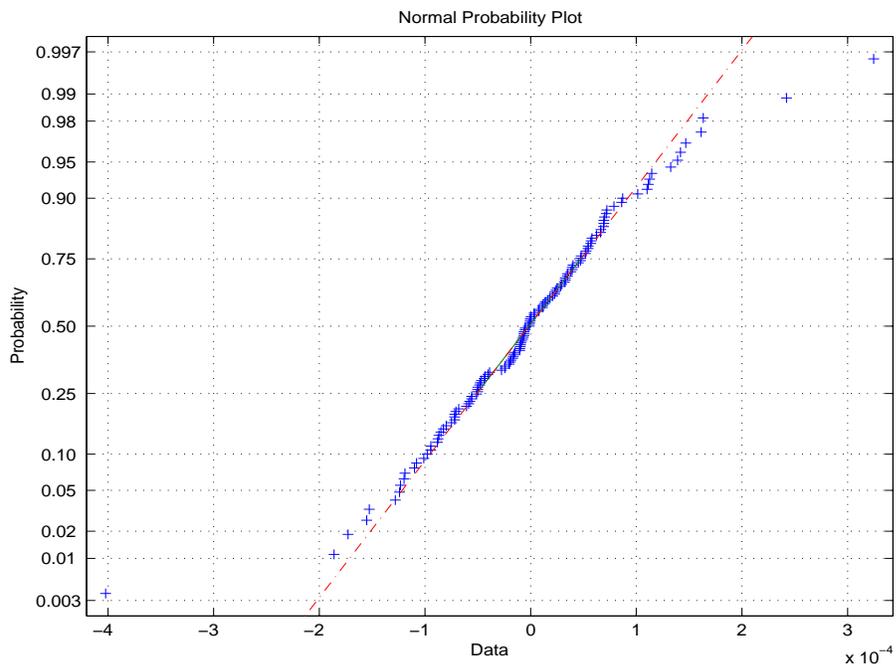
We apply the UBM approach with different batch sizes  $m$  to deal with autocorrelated data. We have also checked for normality. Again, there are no specific rules to choose the right batch size. We picked a batch size of 60 since it gives 1-hour segmentation of data. Further, it provides data that are approximately normally distributed. As we expected, the batched data exhibit much less autocorrelation than without batching. Figure 17 tells us that the data are approximately normally distributed except for a few outliers. Outliers can be explained by observation of the market. For example, some individual investors or financial institutions might make occasional errors when they trade and we can see this phenomenon easily in the US market.

With this change, we apply the standard SPC method. Let  $A_t$  and  $B_t$ , where  $t = 1, 2, \dots$ , be stock prices. From there, we calculate the moving average processes of minute data,  $MA(q)$ . Let them be *movingA* and *movingB* respectively. Then, we calculate the percentage change on each stock price. Let them be *percentA* and *percentB*. From these two, we obtain the  $D$  process,  $D = \textit{percentA} - \textit{percentB}$ . We set  $q = 5$  and  $m = 60$ . We obtain the sample mean and standard deviation of  $D$ . With that, we have a control chart using batch means. We set two thresholds. One is  $\pm 3\sigma$  and the other is  $\pm 1.645\sigma$ . Figure 18 shows the batch means of percent change of the difference process with the two different thresholds.

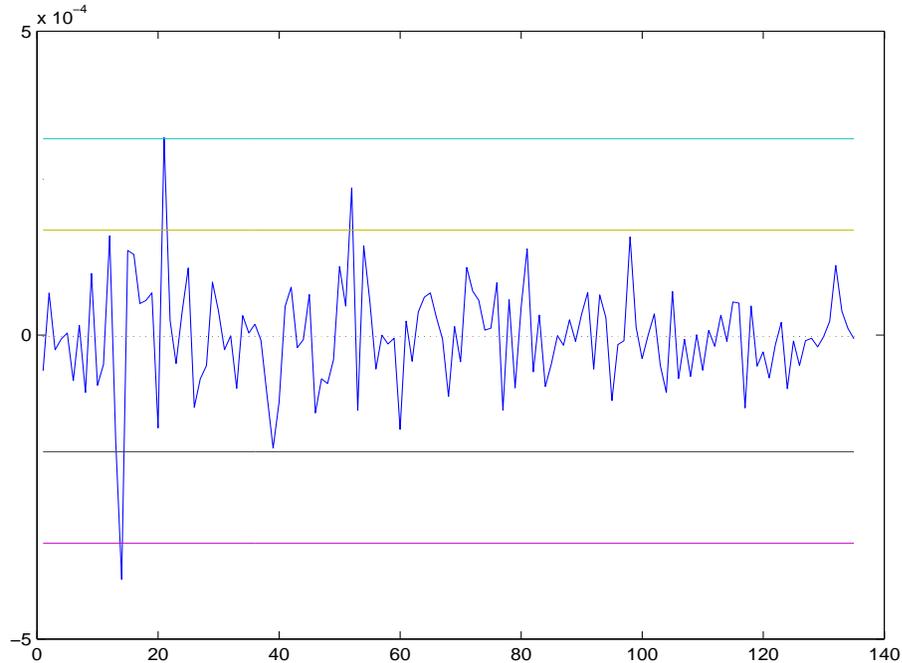
We have set two thresholds for the following reason. With  $\pm 1.645\sigma$ , we simply take an action, but with  $\pm 3\sigma$ , we can imagine that the underlying process's mean has actually changed. Therefore, for  $\pm 3\sigma$ , we need to recalculate the mean and standard deviation of underlying process. As can



**Figure 16:** Autocorrelation plot for batch mean difference process for KO and PBG. Moving average length is 5 minutes and batch size is 60 minutes.



**Figure 17:** Normality plot for batch mean difference process for KO and PBG. Moving average length is 5 minutes and batch size is 60 minutes.



**Figure 18:** Percent change of difference process. Mean and standard deviation is calculated with the first 50 data points and trading strategy is applied after that. Overall profit was \$1330.

be seen in Figure 18, there are a few points out of the thresholds. In particular, batches 21 and 52 are out of the upper inside threshold ( $+1.645\sigma$ ) and batch 14 is out of the lower outside threshold ( $-3\sigma$ ). Note that we may have to re-estimate the mean and standard deviation since batch 14 is out of the lower outside threshold.

Setting up the right strategy is another arduous task in this area. Even though we identify odd behaviors using SPC, bad strategies can render our information useless. For now, we simply use the following strategy. We take actions at the points where the process goes out of the threshold and returns. For example, batch 77 is out of the upper threshold for the KO and PBG pair and batch 78 bounces back in the threshold from the outside. Therefore, we short KO and long PBG at the batch 77 point and long KO and short PBG at batch 78. On the other hand, batch 59 is out of the lower threshold and batch 60 bounces back to the threshold. Therefore, we long KO and short PBG at batch 59 and we short KO and long PBG at batch 60.

With this simple strategy described above, we have tested five different pairs from 4/28/2004 to 05/26/2004. We first compute unweighted batch means data with a batch size of 20 for each batch

**Table 16:** Trading summary on the five different pairs. The first 50 unweighted batch means of data are used to compute the mean and thresholds. Numbers of shares are computed via \$100000/share price. One-minute tick data was collected from 4/28/2004 to 05/26/2004. A batch size of 20 is used for each batch.

Pairs	Number of Transactions	G/L
KO, PBG	3	1330
NT, LU	5	3222
LOW, HD	1	-301
INTC, AMD	6	1776
FDX, UPS	0	0

and moving average length of 5. With these settings, we check the normality and independence assumptions and the two tests do not fail. We use 50 unweighted batch means worth of data to compute the mean and thresholds. With these settings, we apply the trading strategy. For each pair, the numbers of shares are computed using a \$100000/share price. Table 16 shows the overall gain and loss from the 21 trading days. The number of transactions varies depending on the pairs. Note that there are only 5 examples and we need to investigate further with more pairs.

### 5.3 Variance Estimation

As we have seen previously, standard deviation plays a crucial role in our trading model. Therefore, we investigate different variance estimators in this section.

When i.i.d. samples are used to calculate a sample average, the variance of that average is related to the variance of the individual samples by  $1/n$  (i.e.,  $\text{Var}[\bar{Y}_n] = \text{Var}[Y_i]/n$  or  $\text{Var}[Y_i] = \sigma_n^2 \equiv n\text{Var}[\bar{Y}_n]$ ). In this case, the classical variance estimator

$$\widehat{\text{Var}}[\bar{Y}_n] = \frac{S^2}{n} = \frac{1}{n(n-1)} \sum_{i=1}^n (Y_i - \bar{Y}_n)^2$$

is often used and is generally adequate. When the independence assumption is not satisfied, the relationship  $\text{Var}[Y_i] = n\text{Var}[\bar{Y}_n]$  no longer holds, complicating the estimation of the variance of the sample average  $\text{Var}[\bar{Y}_n]$  (or function of  $\text{Var}[\bar{Y}_n]$ , such as  $\sigma_n^2$ ). When the sample is correlated, the variance of the sample average is well known to be

$$\text{Var}[\bar{Y}_n] = \frac{1}{n} \left[ R_0 + \frac{2}{n} \sum_{l=1}^{n-1} (n-l)R_l \right]$$

where  $R_l \equiv \text{Cov}(Y_i, Y_{i+l})$ ,  $l = 0, \pm 1, \pm 2, \dots$ , is the autocovariance function. In contrast,

$$\mathbb{E} \left[ \widehat{\text{Var}}[\bar{Y}_n] \right] = \frac{1}{n} \left[ R_0 - \frac{2}{n(n-1)} \sum_{l=1}^{n-1} (n-l) R_l \right],$$

resulting in a biased estimate of  $\text{Var}[\bar{Y}_n]$  (the magnitude of the bias can be found in Marshall [46]).

It is easy to see that one-minute tick data for stock prices are highly correlated. Therefore, using the right variance estimator is very crucial for setting up the threshold of our trading scheme. We look at three different variance estimators: batch means, standardized time series area, and standardized time series Cramér-von Mises.

### 5.3.1 Batch-Means Estimator

Variance estimation using the batch means approach is popular among experimenters. This approach has been explored by Conway [18], Fishman [21], Schmeiser [58], and many other authors. In the batch-means approach, a sample set of size  $n$  is divided into sub-groups of samples and each sub-group is reduced to a single average value. These averages are then used to compute the batch-means estimator of the variance of the grand sample average,  $\text{Var}[\bar{Y}_n]$ .

Suppose the stationary process  $Y_1, Y_2, \dots, Y_n$  with finite mean  $\mu$  is divided into  $b$  non-overlapping batches, each of size  $m$  (assuming that  $n = mb$ ).

$$\begin{aligned} \text{Batch 1:} & \quad Y_1, Y_2, \dots, Y_m \\ \text{Batch 2:} & \quad Y_{m+1}, Y_{m+2}, \dots, Y_{2m} \\ & \quad \vdots \\ \text{Batch } b: & \quad Y_{(b-1)m+1}, Y_{(b-1)m+2}, \dots, Y_n \end{aligned}$$

The observations  $Y_{(i-1)m+1}, \dots, Y_{im}$  comprise batch  $i$ . For  $i = 1, \dots, b$  and  $j = 1, \dots, m$ , let

$$\bar{Y}_{i,j} \equiv \frac{1}{j} \sum_{k=1}^j Y_{(i-1)m+k} \quad \text{and} \quad \bar{Y}_n \equiv \frac{1}{n} \sum_{k=1}^n Y_k.$$

The  $i$ th batch mean is given by

$$\bar{Y}_{i,m} \equiv \frac{1}{m} \sum_{k=1}^m Y_{(i-1)m+k}, \quad i = 1, 2, \dots, b.$$

If we choose the batch size  $m$  large enough, it is reasonable to treat the  $\bar{Y}_{i,m}$ 's as if there are i.i.d. normal random variables with mean  $\mu$ . Then for sufficiently large  $m$ , the variance of the batch means can be estimated by their sample variance,

$$\begin{aligned}\widehat{\text{Var}}[\bar{Y}_m] &= \frac{1}{b-1} \sum_{i=1}^b (\bar{Y}_{i,m} - \bar{Y}_n)^2 \\ &= \frac{1}{b-1} \left( \sum_{i=1}^b \bar{Y}_{i,m}^2 - b\bar{Y}_n^2 \right).\end{aligned}$$

Therefore, the non-overlapping batch means (NBM) estimator of  $\sigma_n^2$  is given by

$$\widehat{\sigma}_n^2 = m\widehat{\text{Var}}[\bar{Y}_m],$$

where we note that  $m\widehat{\text{Var}}[\bar{Y}_m] \doteq n\widehat{\text{Var}}[\bar{Y}_n] \doteq \sigma^2 (\equiv \lim_{n \rightarrow \infty} \sigma_n^2)$  for sufficiently large  $m$ . That is, for large batch size  $m$ , one assumes that the batch means are approximately i.i.d. normal random variables with mean  $\mu$  and unknown variance  $\sigma_m^2/m \doteq \text{Var}(\bar{Y}_{i,m})$ . Hence, we estimate  $\sigma^2 \doteq \sigma_m^2$  by  $m$  times the sample variance of the batch means. Thus, the NBM estimator for  $\sigma^2$  is

$$\begin{aligned}\hat{V}_B \equiv \widehat{\sigma}_n^2 &= m\widehat{\text{Var}}[\bar{Y}_m] \\ &= \frac{m}{b-1} \sum_{i=1}^b (\bar{Y}_{i,m} - \bar{Y}_n)^2 \\ &= \frac{m}{b-1} \left( \sum_{i=1}^b \bar{Y}_{i,m}^2 - b\bar{Y}_n^2 \right).\end{aligned}$$

The main problem with the application of the batch means method in practice is the choice of the batch size  $m$ . If  $m$  is too small, the batch means  $\bar{Y}_{i,m}$  can be highly correlated. Alternatively, a large batch size can result in the very few batches and potential problems with high variability of resulting confidence interval half length ([15]).

Several variants of the batch-means estimation approach have been investigated by various authors. Meketon and Schmisser [49] introduced the overlapping batch means method which has been explored further by Sargent, Kang, Goldsman [61] and Song and Schmisser [62]. The overlapping batch means estimator is generally offered as a variance reduction modification to the NBM estimator.

### 5.3.2 Standardized Time Series Weighted Area Estimator (STS)

The STS methodology defined by Schruben [59] uses a continuous-time random process to represent the original sequence of samples (i.e.,  $Y_i, i = 1, \dots, n$ ) in a particularly useful form. Let

$$\mathcal{T}_{i,n} = \bar{Y}_i - \bar{Y}_n \quad i = 1, 2, \dots, n \quad \mathcal{T}_{0,n} \equiv 0,$$

where  $\bar{Y}_i$  is the average of the first  $i$  samples in the sequence i.e.,  $\bar{Y}_i \equiv \sum_{j=1}^i Y_j / i$ . Thus,  $E[\mathcal{T}_{i,n}] = 0$  for  $i = 0, 1, \dots, n$ . Then scale the sequence by dividing by  $\sqrt{n}\sigma/i$  and scale the time index of the sequence to the unit interval. The final STS form is

$$T_n(t) \equiv \frac{\lfloor nt \rfloor \mathcal{T}_{\lfloor nt \rfloor, n}}{\sqrt{n}\sigma} = \frac{\lfloor nt \rfloor (\bar{Y}_{\lfloor nt \rfloor} - \bar{Y}_n)}{\sqrt{n}\sigma}, \quad 0 \leq t \leq 1,$$

where  $\lfloor \cdot \rfloor$  is the greatest integer function. Schruben points out that the original time series can be reconstructed from  $T_n(t)$  and  $\bar{Y}_n$ . Therefore, no information is lost by the transformation. The following Assumption 1 is called the Functional Central Limit Theorem (FCLT), and is sufficient to guarantee that the standardized time series converges to a process that we can exploit.

**Assumption 1 (FCLT)** (Billingsley [5]) *There exist constants  $\mu$  and positive  $\sigma$  such that*

$$X_n \Rightarrow \sigma \mathcal{W} \quad \text{as } n \rightarrow \infty,$$

where  $\mathcal{W}$  is a standard Brownian motion,  $\Rightarrow$  denotes weak convergence, and

$$X_n(t) \equiv \frac{\lfloor nt \rfloor (\bar{Y}_{\lfloor nt \rfloor} - \mu)}{\sqrt{n}} \quad \text{for } 0 \leq t \leq 1.$$

The sample paths of  $X_n$  lie in  $\mathcal{D}[0, 1]$ , the space of functions on  $[0, 1]$  that are right continuous and have left-hand limits, while the sample paths of  $\mathcal{W}$  lie in  $\mathcal{C}[0, 1]$ , the space of continuous functions on  $[0, 1]$ .

Assumption FCLT leads to a result involving a standard Brownian bridge process. Let  $\mathcal{B}(t)$  denote a standard Brownian bridge process defined by  $\mathcal{B}(t) = \mathcal{W}(t) - t\mathcal{W}(1)$ . Then  $\mathcal{B}(t) \sim N(0, t(1-t))$  and  $\text{Cov}[\mathcal{B}(s), \mathcal{B}(t)] = \min(s, t) - st, 0 \leq s, t \leq 1$ . Notice that  $\mathcal{W}(1)$  and  $\mathcal{B}(\cdot)$  are independent.

**Theorem 5** *Under Assumption FCLT,*

$$(\sqrt{n}(\bar{Y}_n - \mu), \sigma T_n) \Rightarrow (\sigma \mathcal{W}(1), \sigma \mathcal{B}).$$

**Proof** See Foley and Goldsman [23]. ■

**Remark 5** We have three useful properties from above theorem:

- $\sqrt{n}(\bar{Y}_n - \mu)$  is asymptotically  $\sigma N(0, 1)$ ,
- $\sigma T_n$  is asymptotically  $\sigma$  times a Brownian bridge, and
- $\sqrt{n}(\bar{Y}_n - \mu)$  and  $\sigma T_n$  are asymptotically independent; thus, all information gleaned from  $\sigma T_n$  will be asymptotically independent of  $\sqrt{n}(\bar{Y}_n - \mu)$ .

The (weighted) area estimator for  $\sigma^2$  is based on the statistic

$$S(w; n) \equiv \frac{1}{n} \sum_{k=1}^n w \left( \frac{k}{n} \right) \sigma T_n \left( \frac{k}{n} \right), \quad (52)$$

where  $w(t)$  is a certain weighting function.

The limiting functional of  $S(w; n)$  is

$$S(w) \equiv \int_0^1 w(t) \sigma \mathcal{B}(t) dt,$$

where the weighting function  $w(t)$  is continuous on  $[0, 1]$  and chosen to satisfy  $\text{Var}[S(w)] = \sigma^2$ , so that  $S(w) \sim N(0, \sigma^2)$ . In addition, let  $A(w; n) \equiv S^2(w; n)$  and  $A(w) \equiv S^2(w)$ . Then under mild conditions, the continuous mapping theorem (CMT) (Billingsley [5], Theorem 5.1) implies that  $A(w; n) \xrightarrow{\mathcal{D}} A(w) \sim \sigma^2 \chi_1^2$ , and we call  $A(w; n)$  the *weighted area estimator* for  $\sigma^2$ .

We illustrate a few examples of weighting functions from Goldsman and Schruben [32]:

1.  $w_0(t) \equiv \sqrt{12}$  for all  $t \in [0, 1]$ .
2.  $w_1(t) \equiv \sqrt{45}t$  or  $w_1(t) \equiv \sqrt{45}(1-t)$  gives greater weight for large (small) values of  $t$ .
3. We can also use the “antisymmetric” weighting scheme,  $w_2(t) \equiv \sqrt{840}(3t^2 - 3t + 0.5)$ , which has good bias properties (See Theorem 6 below).

We have defined the STS of a sampled stochastic process for a single long run data samples. Thus, variance parameter estimators utilizing the STS do not need to be based on batched data.

Nevertheless, let us now consider the batched STS area estimator. This is the sample mean of the corresponding estimators from the individual batches, i.e.,

$$\bar{A}(w; b, m) \equiv \frac{1}{b} \sum_{i=1}^b A_i(w; m),$$

where  $A_i$  denotes an estimator from the  $i$ th batch of size  $m$  ( $n = bm$ ). Since the batched estimators are simply linear combinations of estimators from each batch of size  $m$ , we can produce the following results concerning  $E[A(w; n)]$  and  $\text{Var}[A(w; n)]$ .

**Theorem 6** (Goldsman et al. [34]) *Suppose  $Y_i$  is a stationary process for which Assumption FCLT holds,  $\sum_{k=1}^{\infty} k^2 |R_k| < \infty$ , and  $\sum_{k=-\infty}^{\infty} R_k = \sigma^2 > 0$ . Then*

$$E[\bar{A}(w; b, m)] = E[A_i(w; m)] = \sigma^2 + \frac{[(F - \bar{F})^2 + \bar{F}^2]\gamma}{2m} + o(1/m)$$

and

$$\text{Var}[\bar{A}(w; b, m)] \doteq \frac{\text{Var}[A_i(w; m)]}{b} \doteq \frac{2\sigma^4}{b},$$

where  $F(t) \equiv \int_0^t w(s)ds$ ,  $F \equiv F(1)$ ,  $\bar{F}(t) \equiv \int_0^t F(s)ds$ ,  $\bar{F} \equiv \bar{F}(1)$ , and  $\gamma \equiv -2 \sum_{i=1}^{\infty} kR_k$ .

**Remark 6** *It is possible to choose weights  $w(t)$  so that the first-order bias term in front of  $\gamma$  disappears. The antisymmetric function  $w_2(t) \equiv \sqrt{840}(3t^2 - 3t + 1/2)$  (see Goldsman et al. [31]) has this property.*

Therefore, we see that batching typically helps to decrease estimator variance (by a factor of  $b$ ), though this is achieved at a cost of a modest increase in estimator bias (since  $m$  now appears instead of  $n$  in the expected value expressions). Recall that  $A_i(w; m) \xrightarrow{\mathcal{D}} A_i(w) \sim \sigma^2 \chi_1^2$ , where  $\xrightarrow{\mathcal{D}}$  denotes convergence in distribution as  $m \rightarrow \infty$ . Further, under suitable moment and mixing conditions, the  $A_i(w; m)$ 's are asymptotically independent, and thus

$$\bar{A}(w; b, m) \xrightarrow{\mathcal{D}} \frac{\sigma^2 \chi_b^2}{b}.$$

We have discussed variance parameter estimators using the batch means and STS weighted area methods. Now, we review another variance parameter estimator combining both.

**Theorem 7** (Goldsman et al. [33]) As the batch size  $m \rightarrow \infty$ ,

$$\widehat{V}_C \equiv \frac{(b-1)\widehat{V}_B + b\widehat{V}_A}{2b-1} \xrightarrow{\mathcal{D}} \frac{\sigma^2 \chi^2(2b-1)}{2b-1},$$

so that

$$\mathbb{E}[\widehat{V}_C] \rightarrow \sigma^2$$

and

$$\text{Var}[\widehat{V}_C] \rightarrow \frac{2\sigma^4}{2b-1}.$$

Notice that  $\widehat{V}_C$  is asymptotically unbiased and has lower variance than  $\widehat{V}_A$  or  $\widehat{V}_B$ .

### 5.3.3 Cramér-von Mises Variance Estimator

The weighted Cramér-von Mises (CvM) estimator of the variance parameter  $\sigma^2$  has a number of desirable properties. For certain weighting functions, it is a first-order unbiased estimator of  $\sigma^2$ , and its variance is lower than that of many other estimators. Unfortunately, the CvM estimator has the unattractive property of occasionally assuming negative values [47]. We propose various ways to get around the negativity problem. The best trick involves batching, in which case the negativity problem essentially disappears.

We define the *weighted Cramér-von Mises* estimator for  $\sigma^2$  and its limiting functional as

$$C(g; n) \equiv \frac{1}{n} \sum_{k=1}^n g\left(\frac{k}{n}\right) \sigma^2 T_n^2\left(\frac{k}{n}\right) \quad (53)$$

and

$$C(g) \equiv \int_0^1 g(t) \sigma^2 B^2(t) dt, \quad (54)$$

respectively, where  $g(t)$  is a weighting function normalized so that  $\mathbb{E}[C(g)] = \sigma^2 \int_0^1 g(t) t(1-t) dt = \sigma^2$  and  $g''(t)$  is continuous and bounded on  $[0, 1]$ . Under Assumption FCLT, it can be shown that  $C(g; n) \xrightarrow{\mathcal{D}} C(g)$  as  $n \rightarrow \infty$ .

The next theorem gives the expected value and variance of the weighted CvM estimator.

**Theorem 8** (see [31]) Suppose  $\{Y_i, i \geq 1\}$  is a stationary process for which Assumption FCLT holds, and  $\sum_{k=1}^{\infty} k^2 |R_k| < \infty$ . Then

$$\mathbb{E}[C(g; n)] = \sigma^2 + \frac{\gamma}{n}(G-1) + o(1/n), \quad (55)$$

where  $G \equiv \int_0^1 g(t)dt$ . If we also assume uniform integrability of  $C^2(g; n)$ , then as  $n \rightarrow \infty$ ,

$$\text{Var}(C(g; n)) \rightarrow \text{Var}(C(g)) = 4\sigma^4 \int_0^1 g(t)(1-t)^2 \int_0^t g(s)s^2 ds dt. \quad (56)$$

If a weighting function  $g(t)$  can take on negative values, it may be possible for the CvM estimator (53) or its limiting functional (54) to become negative as well. This is a disconcerting property since a negative variance is intuitively displeasing ([47]). There exists a strategy for coping with the problem — applying a batched version of the CvM estimators. We have so far assumed that the CvM estimators for  $\sigma^2$  have been based on one long batch of size  $n$ . The use of batching usually results in estimators with lower variance and only modestly higher bias than their one-batch benchmark. We desire that this lower variance will render as negligible the probability of a negative realization of the batched version of CvM estimator. We briefly describe the batching rules.

- Divide the run into  $b$  contiguous, nonoverlapping batches, each of size  $m$  (assuming  $n = mb$ ). Batch  $i$  consists of observations  $Y_{(i-1)m+1}, Y_{(i-1)m+2}, \dots, Y_{im}$ ,  $1 \leq i \leq b$ .
- Calculate an estimator from each batch (instead of from the entire run). Using the obvious notation, we denote CvM estimator from batch  $i$  as  $C_i(g; m)$ .
- The *batched CvM* estimator for  $\sigma^2$  is the sample mean of the corresponding estimators from the individual batches, i.e.,

$$\bar{C}(g; b, m) \equiv \frac{1}{b} \sum_{i=1}^b C_i(g; m).$$

Under mild moment and mixing conditions (see e.g., [31]), we can show that as  $m \rightarrow \infty$ , with fixed  $b$ ,

$$\bar{C}(g; b, m) \xrightarrow{\mathcal{D}} \bar{C}(g; b), \quad (57)$$

where  $\bar{C}(g; b)$  is the average of  $b$  i.i.d. realizations of  $C_i(g; m)$ .

Since the batched estimators are simple linear combinations of estimators from each batch size  $m$ , Equations (55) and (56) immediately show that

$$\text{E}[\bar{C}(g; b, m)] = \text{E}[C_i(g; m)] = \sigma^2 + \frac{\gamma}{m}(G - 1) + o(1/m) \quad (58)$$

and

$$\text{Var}(\bar{C}(g; b, m)) = \frac{\text{Var}(C_i(g; m))}{b} \rightarrow \frac{\text{Var}(C(g))}{b}. \quad (59)$$

Hence, we see that batching typically helps decrease estimator variance (by a factor of  $b$ ), though this is achieved at the cost of a modest increase in bias (since  $m$  now appears instead of  $n$  in the expected value expressions).

If the probability is small that the estimator for one batch has a low probability of going negative, then it stands to reason that the average of approximately independent estimators from multiple batches will have an even-lower probability of going negative ([47]). Thus, we see from Table 17 that

- The probability of negative realizations is reduced by an order of magnitude simply by averaging estimators from  $b = 2$  batches, at least for all of the weighting functions under study.
- For all intents and purposes, negativity disappears when using  $b > 2$ .

**Table 17:** CvM variance estimators with different numbers of batches. Underlying stochastic process is AR(1) (autoregressive process) with  $\phi = 0.9$  and batch size 1024 is used. Frequencies are obtained with 1000000 replications. We use weighting functions from ([47]).

# of Batch	Frequency			Probability		
	$W_{g_0}$	$W_{g_2^*}$	$W_{g_4^*}$	$P(W_{g_0} < 0)$	$P(W_{g_2^*} < 0)$	$P(W_{g_4^*} < 0)$
1	0	84403	17732	0	0.084	0.018
2	0	9325	245	0	0.031	0.001
3	0	559	1	0	0.002	0.000
4	0	19	0	0	0.000	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0

We can put things on a more sound theoretical footing. Since the  $C_i(g; m)$ 's,  $1 \leq i \leq b$ , are asymptotically independent as  $m \rightarrow \infty$ , we can argue that Equation (56)–(59) and the central limit theorem imply

$$\bar{C}(g; b, m) \xrightarrow{\mathcal{D}} \bar{C}(g; b)$$

$$\begin{aligned} &\approx \text{Nor}(\sigma^2, \text{Var}(C(g))/b) \\ &\sim \sigma^2 \text{Nor}(1, 4\nu(g)/b), \end{aligned}$$

where  $\nu(g) \equiv \int_0^1 g(t)(1-t)^2 \int_0^t g(s)s^2 ds dt$ , and the notation  $\approx$  is taken to mean “is approximately distributed as”. Hence, for large  $m$ , we have the approximation

$$\begin{aligned} \text{P}(\bar{C}(g; b, m) < 0) &\doteq \text{P}(\text{Nor}(1, 4\nu(g)/b) < 0) \\ &= \Phi\left(-\frac{1}{2}\sqrt{\frac{b}{\nu(g)}}\right), \end{aligned} \tag{60}$$

where  $\Phi(\cdot)$  is the standard normal cumulative distribution function. Note that the probability in Equation (60) clearly goes to zero as  $b$  becomes large, proving our point.

## 5.4 Conclusion

Modern portfolio theory says that we can eliminate idiosyncratic risks by including many assets in the portfolio. This process reflects the maxim “Do not put all your eggs on one basket.” However, it admits that we have to live with the systematic risks. Within this portfolio, we can somehow take advantage of arbitrage opportunities using the reactions of asset price movement against macroeconomic condition changes. In our examples, we have shown the potential to find arbitrage opportunities; and we can further study how our approach can combine with modern portfolio theory. With elimination of non-systematic risks by portfolio theory and utilization of systematic risks with our methodology, we might be able to further maximize the profit.

Since data are highly correlated and firm-specific conditions (risk) are unpredictable, we filter the data so that we can significantly reduce the above effects. Thus, we could apply traditional statistical process control concepts here. One of major tasks in setting up the right strategy is to estimate the variance of difference process. Therefore, various variance estimators are studied. In particular, the Cramér-von Mises variance estimator is an excellent choice (in terms of low variance); but it sometimes yields negative values. We investigate how to eliminate this problem. The use of batching results in estimators with lower variance and only moderately higher bias than their one-batch benchmark.

We need to further study in several parts of the thesis. First, more examples should be studied for the trading model. We have used five pairs to test our new approach due to the difficulty of

getting tick data. Once data is available, we need to investigate the trading criteria and profitability. With good trading strategies, we can maximize the profit. This approach can be further implemented with options and indices. We just used this trading systems in the equity market only; however, this approach can be used in other financial markets such as fixed income, utilities, or commodities. In addition, we have ignored transaction costs so far. But, these costs should be considered together. Second, we briefly mentioned a few variance estimators. The threshold play a important role in determining the gains and losses and estimating the variance of the serially correlated data is one of the most challenging tasks. With a good variance estimator, we can maximize the profit by setting up the good threshold which detect the odd behaviors with high probability. We also need to study which variance estimator does the better performance in terms of profitability. Lastly but not least, choosing the batch size in batch means is another big task in our problem. More extensive works are necessary in choosing the batch size.

## CHAPTER VI

### APPENDIX

**Proof** [Lemma 2.] Let  $\rho_x^* := \mathbf{P}_\nu(\mathcal{H}(\Lambda, \mathcal{G}) \geq x)$ . By Assumption A,  $\rho_x^* > 0$ . Let  $\rho \in (0, \rho_x^*)$  be arbitrary. By the definition of  $\gamma$ , we have that

$$\mathbf{P}_\nu(\mathcal{H}(\Lambda, \mathcal{G}) \geq \gamma(\nu, \rho)) \geq \rho$$

and

$$\mathbf{P}_\nu(\mathcal{H}(\Lambda, \mathcal{G}) \leq \gamma(\nu, \rho)) \geq 1 - \rho > 1 - \rho_x^*.$$

Suppose that  $\gamma(\nu, \rho) < x$ . Then

$$\mathbf{P}_\nu(\mathcal{H}(\Lambda, \mathcal{G}) \leq \gamma(\nu, \rho)) \leq \mathbf{P}_\nu(\mathcal{H}(G) \leq x) = 1 - \rho_x^*,$$

which is a contradiction. Thus,  $\gamma(\nu, \rho) \geq x$ . ■

**Proof** [Proposition 4.] See [39]. ■

**Proof** [Proposition 5.] Notice that a  $(1 - \rho)$  quantile of a random variable  $Y$  can be expressed as the optimal solution of the problem  $\min_\eta \mathbf{E}\phi(Y, \eta)$ , where

$$\phi(Y, \eta) = \begin{cases} (1 - \rho)(Y - \eta) & \text{if } \eta \leq Y \\ \rho(\eta - Y) & \text{if } \eta \geq Y. \end{cases}$$

To see this, notice that the subdifferential  $\partial_\eta \mathbf{E}\phi(Y, \eta)$  can be expressed as

$$\partial_\eta \mathbf{E}\phi(Y, \eta) = [\rho - P(Y \geq \eta), -(1 - \rho) + P(Y \leq \eta)].$$

Therefore,  $\eta$  satisfies the optimality condition  $0 \in \partial_\eta \mathbf{E}\phi(Y, \eta)$  if and only if

$$\rho - P(Y \geq \eta) \leq 0$$

and

$$-(1 - \rho) + P(Y \leq \eta) \geq 0,$$

i.e., if and only if  $\eta$  is a  $(1 - \rho)$  quantile of  $Y$ . A similar argument shows that the sample  $(1 - \rho)$  quantile of a sample  $Y_1, \dots, Y_N$  (call it  $\hat{\eta}$ ) is the solution to the sample average approximation problem  $\min_{\eta} N^{-1} \sum_{i=1}^N \phi(Y_i, \eta)$ . Since the objective function  $E\phi(Y, \eta)$  is convex in  $\eta$ , it follows that the distance between  $\hat{\eta}_\rho$  and the set of  $(1 - \rho)$  quantiles of  $Y$  goes to zero as  $N$  goes to infinity with probability one. There are numerous examples in [39].

Let  $\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_{N_G}$  be i.i.d. samples from  $f(\cdot, \nu)$ . We look at two different cases to prove.

- Case 1:  $P_\nu(\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) > a) > 0$ .

By Lemma 2, we have that  $\gamma(\nu, \rho^*) > a$  for any  $\rho^* \in (0, \rho_a^+)$ , where  $\rho_a^+ = P_\nu(\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) > a) > 0$ . As discussed earlier, the distance between the sample  $(1 - \rho^*)$  quantile  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho^*)$  of  $\mathcal{H}(\mathbf{\Lambda}_1), \dots, \mathcal{H}(\mathbf{\Lambda}_{N_G})$  and the set of  $(1 - \rho^*)$  quantile of  $\mathbf{\Lambda}$  goes to zero as  $N_G$  goes to infinity with probability one. Since  $\hat{\gamma}(\nu, \rho^*) > a$ , it follows that  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho^*) > a$  w.p.1 for  $N_G$  large enough. Moreover, the probability that  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho^*) > a$  for a given  $N_G$  goes to one exponentially fast.

- Case 2:  $P_\nu(\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) > a) = 0$ .

This case is that  $a$  is the maximum value achieved by  $\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})$ . By Assumption A, this implies that  $\rho_a^0 := P_\nu(\mathcal{H}(\mathbf{\Lambda}, \mathcal{G}) = a) > 0$  and thus, for any  $\rho^* \in (0, \rho_a^0)$  we must have  $\gamma(\nu, \rho^*) = a$ . It follows that  $\gamma(\nu, \rho^*) = a$  is also the unique  $(1 - \rho^*)$  quantile of the random variable  $\mathcal{J} := a\mathbf{1}_{\{\mathcal{H}(\mathbf{\Lambda}, \mathcal{G})=a\}}$ . It is clear that  $\hat{\gamma}_{N_G}^a := a\mathbf{1}_{\{\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho^*)=a\}}$  is a sample  $(1 - \rho^*)$  quantile of  $\mathcal{J}_1, \dots, \mathcal{J}_{N_G}$ , where  $\mathcal{J}_i := a\mathbf{1}_{\{\mathcal{H}(\mathbf{\Lambda}_i)=a\}}$ . Since the distribution of  $\mathcal{J}$  has finite support, it follows from the result in [60] that  $\hat{\gamma}_{N_G}^a = \gamma(\nu, \rho^*) = a$  w.p.1 for  $N_G$  large enough, and, moreover, the probability that  $\hat{\gamma}_{N_G}^a = \gamma(\nu, \rho^*) = a$  for a given  $N_G$  goes to one exponentially fast since  $\hat{\gamma}_{N_G}^a = a$  if and only if  $\hat{\gamma}_{N_G}(\mathbf{\Lambda}, \rho^*) = a$ .

The proof is completed. ■

**Proof** [Lemma 3.] Consider a point  $\nu \in V$  and a sequence of points  $\nu_k \in V$  converging  $\nu$ . By the Lebesgue dominated convergence theorem and Assumption C,

$$\lim_{k \rightarrow \infty} \mathbb{E}[h(\Lambda, \nu_k)] = \mathbb{E}[\lim_{k \rightarrow \infty} h(\Lambda, \nu_k)]$$

and by Assumption B,

$$\lim_{k \rightarrow \infty} h(\Lambda, \nu_k) = h(\Lambda, \nu) \quad \text{w.p.1.}$$

Hence,  $l(\nu_k)$  tends to  $l(\nu)$ .

Now consider a new sequence  $X_k$  of neighborhoods of  $\nu$  in  $V$  shrinking to  $\{\nu\}$  and the function

$$d_k(\lambda) = \sup\{|h(\lambda, x) - h(\lambda, \nu)| : x \in X_k\}.$$

It follows from Assumption B that for almost every  $\lambda$ ,  $d_k(\lambda)$  tends to zero as  $k \rightarrow \infty$ . Furthermore, Assumption C implies that the family  $\{d_k(\lambda), k = 1, 2, \dots\}$  is dominated by an integrable function; therefore,

$$\lim_{k \rightarrow \infty} \mathbb{E}[d_k(\Lambda)] = \mathbb{E}[\lim_{k \rightarrow \infty} d_k(\Lambda)] = 0.$$

Now we have that

$$|\bar{l}_{N_G}(x) - \bar{l}_{N_G}(\nu)| \leq N_G^{-1} \sum_{i=1}^{N_G} |h(\lambda_i, x) - h(\lambda_i, \nu)|$$

and hence,

$$\sup_{x \in X_k} |\bar{l}_{N_G}(x) - \bar{l}_{N_G}(\nu)| \leq N_G^{-1} \sum_{i=1}^{N_G} d_k(\lambda_i).$$

This implies that for any given  $\epsilon > 0$ , there exists a neighborhood  $X$  of  $\nu$  such that w.p.1 for sufficiently large  $N$ ,

$$\sup\{|\bar{l}_{N_G}(x) - \bar{l}_{N_G}(\nu)| : x \in X\} < \epsilon.$$

Since  $V$  is compact, there exist a finite number of points  $\nu_1, \dots, \nu_m \in V$ , and corresponding neighborhoods  $X_1, \dots, X_m$  covering  $V$  such that w.p.1 for sufficiently large  $N_G$ ,

$$\sup\{|\bar{l}_{N_G}(x) - \bar{l}_{N_G}(\nu_j)| : x \in X_j, j = 1, \dots, m\} < \epsilon.$$

Furthermore, these neighborhoods can be chosen in such a way that

$$\sup\{|l(x) - l(\nu_j)| : x \in X_j, j = 1, \dots, m\} < \epsilon$$

since  $l(\nu)$  is continuous on  $V$ . Due to the strong law of large numbers,

$$|\bar{l}_{N_G}(\nu_j) - l(\nu_j)| < \epsilon, \quad j = 1, \dots, m.$$

Therefore, we can conclude that

$$|\bar{l}_{N_G}(\nu) - l(\nu)| < 3\epsilon$$

for all  $\nu \in V$ . ■

**Proof** [Theorem 4.] From Lemma 3, we know that for any  $\epsilon > 0$  and all sufficiently large  $N_G$  and all  $\nu \in V$ ,

$$|\bar{l}_{N_G}(\nu) - l(\nu)| < \epsilon \text{ w.p.1.}$$

Now suppose  $\nu^*$  is a unique and consider a neighborhood  $X$  of  $\nu^* \in V$ . Since  $l(\nu)$  is continuous and  $V$  is compact, there exists  $\epsilon > 0$  such that

$$l(\nu^*) - l(\nu) > 2\epsilon$$

for all  $\nu$  in  $V$  and  $\nu$  not in  $X$ . Then two above equations tell us that

$$l(\nu^*) > \bar{l}(\nu) + \epsilon$$

for all  $\nu$  in  $V$  and  $\nu$  not in  $X$ . On the other hand, we have

$$\bar{l}_{N_G}(\tilde{\nu}_{N_G}) = \tilde{\eta}_{N_G} > l(\nu^*) - \epsilon.$$

This then says that  $\tilde{\nu}_{N_G} \in X$ , and since the neighborhood  $X$  was arbitrary, we obtain that  $\tilde{\nu}_{N_G}$  converges to  $\nu^*$  w.p.1.

Since we now know that  $\tilde{\nu}_{N_G}$  is the optimal parameter vector, option price  $\tilde{P}$  with policy set sampled from  $f(\cdot, \tilde{\nu}_{N_G})$  is the true value of option  $P^*$  by the SLLN and the definition in Equation (21). ■

**Proof** [Proposition 6.] See [39]. ■

**Table 18:** Simulation prices on min-put options on two assets with the same volatility by  $\min(S_1, S_2)$  and the difference of  $S_1$  and  $S_2$ . Option parameters:  $K = 100$ ,  $r = 0.06$ ,  $T = 0.5$ , and  $\sigma_1 = \sigma_2 = 0.6$ .

$S_1$	$S_2$	Euro	FD	Lower	Upper	Point	Error (%)	CPU (s)
80	80	36.86	37.30	37.12	37.83	37.47	0.47	7.30
80	100	31.64	32.08	32.00	32.68	32.34	0.80	7.48
80	120	28.65	29.14	28.94	29.62	29.28	0.48	7.29
100	100	24.73	25.06	24.90	25.53	25.22	0.62	7.28
100	120	20.61	20.91	20.87	21.44	21.15	1.17	7.35
120	120	15.70	15.92	15.87	16.28	16.08	0.99	7.24

**Table 19:** Simulation prices on min-put options on two assets with different volatilities by  $\min(S_1, S_2)$  and the difference of  $S_1$  and  $S_2$ . Option parameters:  $K = 100$ ,  $r = 0.06$ ,  $T = 0.5$ , and  $\sigma_1 = 0.4$ ,  $\sigma_2 = 0.8$ .

$S_1$	$S_2$	Euro	FD	Lower	Upper	Point	Error(%)	CPU(s)
80	80	37.55	38.01	37.84	38.55	38.19	0.48	7.43
80	100	31.81	32.23	32.01	32.60	32.31	0.23	7.27
80	120	28.09	28.54	28.34	28.97	28.66	0.41	7.17
100	80	32.86	33.34	33.22	33.98	33.60	0.78	7.64
100	100	25.47	25.81	25.64	26.30	25.97	0.63	7.36
100	120	20.48	20.75	20.56	21.11	20.84	0.41	7.11
120	80	30.69	31.21	31.10	31.88	31.49	0.89	7.03
120	100	22.44	22.77	22.63	23.29	22.96	0.84	7.42
120	120	16.76	16.98	16.94	17.37	17.15	1.01	7.52

## REFERENCES

- [1] AIT-SAHALIA, F. AND P. CARR 1997. American Options: A Comparison of Numerical Methods. In *Numerical Methods in Finance*, eds. L.C.G. Rogers and D. Talay, Cambridge University Press, Cambridge.
- [2] ANDERSEN, L. 2000. A Simple Approach to the Pricing of Bermudan Swaptions in the Multi-Factor Libor Market Model. *Journal of Computational Finance*, 3: 5-32.
- [3] ANDERSEN, L. AND M. BROADIE 2001. A Primal-Dual Simulation Algorithm for Pricing Multi-Dimensional American Options. Working paper, Columbia University.
- [4] BARRAQUAND, J., AND D. MARTINEAU 1995. Numerical Valuation of High Dimensional Multivariate American Securities. *Journal of Quantitative and Computative Analysis*, 30:383-405.
- [5] BILLINGSLEY, P. 1968. *Convergence of Probability Measures*. John Wiley & Sons, New York.
- [6] BOYLE, P.P. 1977. Options: A Monte Carlo Approach. *Journal of Financial Econometrics*, 4: 323-338.
- [7] BROADIE, M., AND J. DETEMPLE Winter 1996. American option valuation: new bounds, approximations, and a comparison of existing methods. *The Review of Financial Studies*, 9(4): 1211-1250.
- [8] BROADIE, M., AND P. GLASSERMAN 1997a. Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8/9): 1323-1352.
- [9] BROADIE, M., AND P. GLASSERMAN 1997b. Monte Carlo methods for pricing high-dimensional American options: an overview. *Net Exposure*, 3: 15-37.
- [10] BROADIE, M., AND P. GLASSERMAN 1997c. A Stochastic Mesh Method for Pricing High-dimensional American Options. Working paper, Columbia University.
- [11] <http://www.bollingerbands.com/>
- [12] CARR, P 1998. Randomization and the American Put. *The Review of Financial Studies*, 11(3):597-626.
- [13] CARR, P., AND G. YANG 1997. Simulating American Bond Options in an HJM Framework. Working paper, Morgan Stanley.
- [14] CARRIERE, J. F. 1996. Valuation of the Early-Exercise Price for Derivative Securities using Simulations and Splines. *Insurance: Mathematics and Economics*, 19: 19-30.
- [15] CHANG, B. 2004. *Estimation Techniques for Non-linear Functions of the Steady-State Mean in Computer Simulation* Ph.D. dissertation, Georgia Tech.
- [16] CROWDER, S.V. 1989. Design of Exponentially Weighted Moving Average Schemes. *Journal of Quality Technology*, 21.

- [17] CROWDER, S.V., AND M. HAMILTON 1992. An EWMA for Monitoring a Process Standard Deviation. *Journal of Quality Technology*, Vol 21.
- [18] CONWAY, R.W. 1963. Some Tactical Problems in Digital Simulation. *Management Science*, 43: 1288-1295.
- [19] EMANUEL DERMAN, IRAJ KANI, DENIZ ERGENER, INDRAJIT BARDHAN (1996). Enhanced Numerical Methods for Options with Barriers. *Financial Analysts Journal*, (November-December 1995): 65-74.
- [20] FAN, J., Q. YAO, H. TONG 1996. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83: 189-206.
- [21] FISHMAN, G.S. 1978. Grouping Observations in Digital Simulation. *Management Science*, 24: 510-521.
- [22] FIGLEWSKI, S., B. GAO. 1999. The adaptive mesh model: a new approach to efficient option pricing. *Journal of Financial Economics*, 53: 313-351.
- [23] FOLEY, R.D., D. GOLDSMAN 2000. Confidence Intervals Using Orthonormally Weighted Standardized Time Series. *ACM Transactions on Modeling and Computing Simulation*, 9: 297-325.
- [24] FU, M. C., S. B. LAPRISE, D. B. MADAN, Y. SU, AND R. WU. Spring 2001. Pricing American options: a comparison of Monte Carlo simulation approaches. *Journal of Computational Finance*, 4(3): 39-88.
- [25] KAPUR J.N. AND H.K. KESAVAN 1992. *Entropy Optimization Principles with Applications*. Academic Press, San Diego.
- [26] KARATZAS, I. AND S. SHREVE 1988. *Brownian Motion and Stochastic Calculus*. Springer-Verlag, New York.
- [27] EYDELAND, A, AND KRZYSZTOF WOLYNIEC. 2002. *Energy and Power Risk Management: New Developments in Modelig, Pricing, and Hedging*. John Wiley & Sons.Inc, New York.
- [28] GARCIA, D. 2003. Convergence and Biases of Monte Carlo Estimates of American Options Using a Parametric Exercise Rule. *Journal of Economic Dynamics and Control*, 27: 1855-1879.
- [29] GATEV, EVAN G., GOETZMANN, WILLIAM N, ROUWENHORST, K. GEERT 2003. Pairs Trading: Performance of a Relative Value Arbitrage Rule. Working paper.
- [30] GLASSERMAN, P. 2003. *Monte Carlo Methods in Financial Engineering*. Springer, New York.
- [31] GOLDSMAN, D., K. KANG, AND A. SEILA. 1999. Cramér-von Mises Variance Estimators for Simulations. *Operations Research*, 47: 299-309.
- [32] GOLDSMAN, D. AND L. SCHRUBEN 1990. New Confidence Interval Estimators Using Standardized Time Series. *Management Scicence*, 36(3): 393-397.
- [33] GOLDSMAN, D., L. SCHRUBEN, AND J. SWAIN 1990. Tests for Transient Means in Simulated Time Series. *Naval Research Logistics*, 41: 171-187.

- [34] GOLDSMAN, D., K. KANG, S. KIM, A.F. SEILA, AND G. TOKOL 2003. *Combining Standardized Time Series Area and Cramér-von Mises Variance Estimators*. Technical Report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.
- [35] GRANT, D., G. VORA, D. WEEKS. 1996. Simulation and the early-exercise option problem. *Journal of Financial Engineering*, 5(3): 211-227.
- [36] HADAMARD, J.S. 1902. *Sur les problèmes aux dérivées partielles et leur signification physique*. Princeton University Bulletin, 49-52.
- [37] HARTLEY, P. M. 2000. Pricing a multi-asset American option. Working paper, University of Bath.
- [38] HAUGH, M AND KOGAN, L. 2001. Pricing American Options: A Duality Approach. Working paper, MIT.
- [39] HOMEN-DE-MELLO, T. AND R.Y. RUBINSTIN 2003. Rare Event Probability Estimation for Static Models via Cross-Entropy and Importance Sampling. Working paper.
- [40] HULL, J. C. . *Options, Futures, and Other Derivatives*. 4th Ed. Prentice Hall, NJ.
- [41] IBÁÑEZ, A. AND ZAPATERO FERNANDO 2004. Monte Carlo Valuation of American Options through Computation of the Optimal Exercise Frontier. *Journal of Financial and Quantative Analysis*, 2: 253-275.
- [42] KARGUINE, VLADISLAV 2003. Optimal Convergence Trading. Working paper, Boston University and Cornerstone Research.
- [43] KERTZ, R. 2001. *Stochastic Processes in Finance II*. Class Note, Georgia Institute of Technology.
- [44] LAPRISE, S. B., M. C. FU, S. I. MARCUS, AND A. E. B. LIM. 2001. *A linear interpolation approach for pricing American-style derivatives*. Technical Report, Institute for Systems Research, University of Maryland.
- [45] LONGSTAFF, F. A., AND E. S. SCHWARTZ. 2001. Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14: 113-147.
- [46] MARSHALL, W 2000. *Robust Variance Estimation For Ranking and Selection*. Ph.D. dissertation, Georgia Institute of Technology.
- [47] MARSHALL, W, D. GOLDSMAN, B. CHANG, AND LEE, S. 2004. Cramér-von Mises Variance Estimators Don't Have to Be Negative. Working paper, Georgia Institute of Technology.
- [48] MERTON, R.C. 1976. Option Pricing When Underlying Stock Returns Are Discountinuous. *Journal of Financial Economics*, 3: 125-144.
- [49] MEKTON, M.S., AND B. SCHMEISER 1984. *Overlapping Batch Means: Something for Nothing?* In proceedings of the 1984 Winter Simulation Conference, pages 227-230. The Institute of Electrical and Electronic Engineers(IEEE).
- [50] MONTGOMERY, D. 2001. *Introduction to Statistical Quality Control*. John Wilson & Sons. New York.

- [51] MYER, G. 2002. *Numerical Method in Finance*. Class Note, Georgia Institute of Technology.
- [52] ØKSENDAL, B. 2003. *Stochastic Differential Equations*. Fifth Ed., Springer New York.
- [53] ROGERS, L.C.G. 2001. Monte Carlo valuation of American options. Working paper, University of Bath.
- [54] ROSS, S. 1996. *Stochastic Processes*. John Wiley & Sons, New York.
- [55] RUBINSTEIN, R. Y. 1999. The Simulated Entropy Method for Combinatorial and Continuous Optimization. *Methodology and Computing in Applied Probability*, 2: 122-190.
- [56] RUBINSTEIN, R. Y. AND A. SHAPIRO 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization via the Score Function Method.*, John Wiley & Sons, New York.
- [57] RUNGER, G.C. AND T.R. WILLEMAIN 1996. Batch Means Control Charts for Autocorrelated Data. *IIE Transactions*, 28.
- [58] SCHMEISER, B.W. 1982. Batch Size Effects in the Analysis of Simulation Output. *Operations Research*, 30: 556-568.
- [59] SCHRUBEN, L.W. 1983. Confidence Interval Estimation Using Standardized Time Series. *Operations Research*, 31: 1090-1108.
- [60] SHAPIRO, A. AND T. HOMEM-DE-MELLO. 2000. On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, 11: 70-86.
- [61] SARGENT, B.W., K. KANG, AND D. GOLDSMAN 1992. An Investigation of Finite Sample Behavior of Confidence Interval Estimators. *Operations Research*, 40(5): 898-913.
- [62] SONG, W.T., AND B.W. SCHMEISER 1993. Variance of the sample mean: Properties and graphs of quadratic-form estimators. *Operations Research*, 41(3): 501-517.
- [63] STEPHEN FIGLEWSKI, BIN GAO (1999). Adapted Mesh Method: Option Pricing. *Journal of Financial Economics*, 53: 313-351.
- [64] TILLEY, J. 1993. Valuing American options in a path simulation model. *Transactions of the Society of Actuaries*, 45: 83-104.
- [65] TSITSIKLIS, J. N., AND B. VAN ROY. 2001. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, forthcoming.
- [66] WU, R., AND M. FU. 2003. Optimal Exercise Policies and Simulation Based Valuation for American-Asian Options. *Operations Research*, 1: 52-66.
- [67] WILCOX, D. 1990. *Energy Futures and Options: Spread Options in the Energy Markets*. Goldman Sachs.
- [68] <http://www.mit.edu/6.454/gew/CEtutorial.pdf>

## VITA

Sungjoo Lee received the B.S. in Industrial Engineering from Iowa State University, Ames, Iowa, in 1999. He started his Ph.D program in the School of Industrial and Systems Engineering at the Georgia Institute of Technology in Spring 2000. While he was pursuing his Ph.D. degree, he obtained his M.S.O.R. degree in Industrial and Systems Engineering in 2002. He also started the Quantitative and Computational Finance program from the School of Mathematics, Dupree School of Management, and School of Industrial and Systems Engineering. His research interests include pricing path-dependent derivative securities, especially using Monte Carlo simulation, statistical analysis of the equities market, and setting up arbitrage trading models.