Pricing Path-Dependent Derivatives

Tian-Shyr Dai
Department of Computer Science and Information Engineering
National Taiwan University

Contents

List of Figures

Abstract

A financial derivative is a financial instrument whose payoff is based on other elementary financial instruments, such as bonds or stocks. With the rapid growth and deregulation of many economies, more derivatives are being designed by the financial institutions to satisfy the needs of their clients. This also gives rise to new problems in pricing and hedging.

It has been proved that pricing arbitrary European options is a $\sharp$ P-hard problem [2]. Even now, there are some notorious derivatives, such as Asian options, that can not be efficiently priced. These sophisticated derivatives are playing important roles in financial markets.

Pricing path-dependent derivatives with tree models combined with state variables is a standard numerical approach, especially when we can not get proper closed form. Monte Carlo simulation is also a good alternative, but it is less efficient than the tree methods in general. [7]

A systematic approach to constructing data structures and algorithms for pricing is the first goal of this thesis. I will first introduce how the idea works by illustrating the underlying ideas. Then I will apply the ideas to European-style path-dependent options, such as barrier options, geometric average-rate option and Asian-like interest rate options. For American-style options, the early exercise property of these options is critical, and the numerical data from Asian-like derivatives also suggests that it is a monotone curve rather than an oscillated one. The second goal of this thesis is therefore about demonstrating these properties of the pricing data.

Pricing the arithmetic average rate options is a hard problem. This is because we can't derive a proper formula for describing the distribution of the sums of log-normal random variables. A new lattice model is designed for pricing the arithmetic average-rate options. This efficient approach can give a more reliable answer than other approaches.

Chapter 1

Introduction

1.1 Setting the Ground

A financial derivative is a financial instrument whose payoff is based on other elementary financial instrument, such as bonds or stocks. With the rapid growth and deregulation of economies, more derivatives are being designed by the financial institutions to satisfy the needs of their clients. More sophisticated derivatives created by financial institutions become so complex and hard to be understood. On the one hand, the financial innovations make the market more efficient. On the other hand, they also give rise to new problems in pricing and hedging. These problems will become more important as these sophisticated derivatives start play to important roles in financial markets.

A new discipline, named *financial engineering*, is founded under such circumstances. This new discipline involves the design, development, and implementation of innovative financial instruments and processes, through which we can meet the requirement of risk management. In order to solve the finance problems, I will combine knowledge from different subjects in finance, computer science and mathematics.

Pricing arbitrary derivatives has been proved to be a $\sharp$-P problem [2]. Usually, we can price some complex derivatives via Monte Carlo method. But it's hard for pricing American-style options by this method, the efficiency of it is also poor. This situation give us strong intuition for finding appropriate systematic approach about pricing.

This thesis discusses solving path-dependent derivatives with tree methods and state variables, and provides a systematic and efficient approach for these sophisticated derivatives. We can easily and intuitively build up a computer program for pricing these sophisticated derivatives following our approach. From the pricing data, we examine some interesting properties, like the early exercise property of the American-style options and the oscillating curves of pricing data. These observations give us insights into derivatives pricing.

In order to solve the strongly path-dependent arithmetic average-rate options, a new pricing tree model, using the closure property on natural number, is investigated. This model provides a good approach for pricing these options. The experimental data in this thesis show that this approach is efficient and can offer satisfactory results.

1.2 Structures of the Thesis

I organize this thesis as follows. In Chapter two, I will introduce some underlying knowledge about financial derivatives, including the properties of derivatives, pricing models and methods. In Chapter 3, I will describe how to implement a computer program for pricing the derivatives efficiently and systematically. Some complex examples like the geometric average-rate options and arithmetic average interest-rate options are priced in Chapter 4 by the approach described in Chapter 3. Some observations about the pricing results are also made in this chapter. An new pricing tree model, used for pricing the arithmetic average-rate options, is described in Chapter 5. We will compare this algorithm with numerical results from other papers.

Chapter 2

Fundamental Concepts

In this chapter, I would like to introduce some background knowledge you about financial world, basic rules of derivatives and pricing methods. Some concepts, like stopping time, will be introduced later when needed.

2.1 Basic Assumptions

In this section, I would like to introduce basic assumptions in finance and mathematical models. Survey on the background mathematics is also given in this section

2.1.1 Basic Assumptions in Finance

The following statements are *needed* for all the models in this thesis.

Rational Behavior

People in this ideal market all behave *rationally*. That is to say, they try to maximize their benefit. They like to gain more and avert risk (risk averters). This is also a basic assumption used in most economic models.

Efficient Market

All derivatives are *priced correctly*. You can trade at the market price. This assumption implies that there is no liquidity problem in this ideal market.

Complete competitive market

All people behave like *price takers* in this market. Trading behaviors do not influence the prices in the market. So traders in this market do not care about the *side effects* of their activities, such as price movements caused by their trading.

No Arbitrage Opportunity

Arbitrage is any trading strategy that requires no cash investment and has some probability of making profits without any risk of loss. In our *ideal* environment, there should be no arbitrage opportunity for any trading strategy. That is to say, you can not make excess return without suffering any risk. This important assumption implies that the return of any riskless portfolio is the risk-free rate.

No Transaction Cost

No tax and shoes leather cost need to be taken into consideration. This assumption will make our models become simpler.

The Markov Property

A Markov process is a particular type of stochastic process where only the present value of a variable is relevant for predicting the future. To simplify the models for computation, all processes in this thesis are Markov processes.

2.1.2 Survey of Mathematical Tools

In this subsection, I will introduce stochastic processes needed to model the financial variables and some important tools needed for handling these models.

2.1.2.1 Stochastic Process

Any variable whose value changes over time in an uncertain way is called a *stochastic process.* Stochastic processes can be classified as *discrete-time processes* or *continuous-time ones.* A discrete-time stochastic process is one in which the value of the variable can change only at some certain time, whereas a continuous-time stochastic process allows changes can take place at any time.

Formally, a stochastic process $X = \{X(t)\}$ is a time series of random variables. In other words, $X(t)$ is a random variable for time t, and it is usually called the *process state* at time t. We often write $X(t)$ as X_t in shorthand. If the time t comes from a countable set, we call X_t a *discreet-time* stochastic process. If the time t forms a continuum, we call it a *continuous-time* stochastic process. Any *realization* of X is called a *sample path* or *trajectory.* Note that a sample path is but an ordinary function of t. Figure 2.1 plots a sample realization of a *Brownian motion process.*

Wiener Process *Wiener process* is a particular type of Markov stochastic process. It is sometimes referred to as *Brownian motion* in physics. It is often used for simulating stochastic variables in physics and finance.

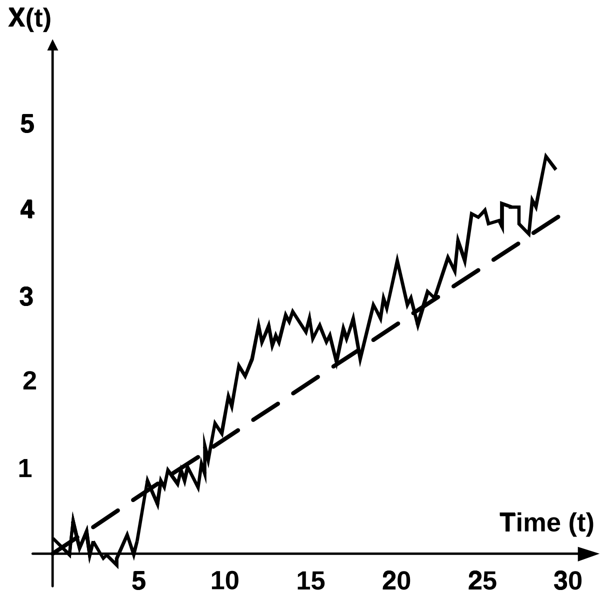


Figure 2.1: SAMPLE PATH OF A BROWNIAN MOTION PROCESS. The stochastic process with volatility is testified by the jittery of the path. The related deterministic process with randomness been taken out is also plotted for reference.

I will illustrate the idea with a standard Wiener process. Assume the behavior of Z_t follows a standard Wiener process. Let's consider the change of its value in a small interval of time $\triangle t$. Let $\triangle z$ be the change in z during $\triangle t$. Then the following properties must hold :
Property 1

$$\triangle z = \epsilon \sqrt{\triangle t}$$

where ξ is a random drawing from the standardized normal distribution. [1]
Property 2

The value of $\triangle z$ for any two disjoint time intervals are independent.

Thus $\triangle z$ is a normal distribution with zero mean and its standard deviation is equal to $\sqrt{\triangle t}$ by property 1. Property 2 implies that z follows a Markov process.

Generalized Wiener Process The standard Wiener process is a stochastic process with mean zero and variance 1. A *generalized process* can be defined in terms of a standard Wiener process dz as follows:

$$dx = a\,dt + b\,dz \tag{2.1}$$

[1] Standard normal distribution is a normal distribution with mean zero and standard deviation 1.

where a and b are constants.

2.1.2.2 Ito Process

In this subsection, I will introduce a powerful tool, developed by Ito [3], to handle stochastic processes. An *Ito process* is a stochastic process $X = \{X_t, t \geq 0\}$ satisfying

$$X_t = X_0 + \int_0^t a_s + \int_0^t b_s dW_s ds, t \geq 0, \tag{2.2}$$

where X_0 is *"starting point,"* and a_t and b_t are two stochastic processes satisfying $\int_0^t |a_s|ds < \infty$ and $\int_0^t |b_s|ds < \infty$, respectively, almost surely for all $t \geq 0$. A shorthand for (2.2) is the following Ito differential,

$$dX_t = a_t dt + b_t \sqrt{d_t}\xi \tag{2.3}$$

where ξ is again a random variable from the standard normal distribution. From (2.3), it is easy to find that dW in (2.2) is a normal distribution with mean zero and variance dt. It is easy to see that (2.3) reduces to (2.1) when a_t and b_t are all constants.

Ito's Lemma The central tool in the Ito integral is Ito's lemma. It says that a smooth function of an Ito process is also an Ito process. Assume X_t is an Ito process of (2.1), and f is a smooth function, then the following follows from Ito's lemma:

$$df(X) = f'(x)adt + f'(x)bdW + \frac{1}{2}f''(x)b^2 dt \tag{2.4}$$

Ito's process can be generalized to higher dimensions for handling multi-dependent or independent Winner processes. Consult [3] for more information.

2.1.3 Log-normal Model for Stock Price

A log-normal distribution for the stock price is the standard model used in financial economics. This is because its properties can satisfy reasonable assumptions about the random behavior of stock prices. The stochastic log-normal model for the non-dividend-paying stock is

$$\frac{dS}{S} = \mu dt + \sigma dz \tag{2.5}$$

Equation (2.5) is also known as *geometric Brownian motion* where S is the value of stock. The variables μ and σ are referred to as the expected return and volatility, respectively.

Clearly, the return rate[2] of stock is a random variable with a normal distribution. That is why we call it *log-normal.* The stock price realized by this model will never become negative, and the percent changes of S are independent and identically distributed. These nice properties make it a good model for simulating the stock price.

2.1.4 Term Structure Models

While there is a standard model for the stock prices, there is still no standard model for the interest rate. This is because of the complex nature of the term structure. In this section, I will just introduce two interest rate models, says Vasicek model [18] and Hull and White model [8], which are related to this thesis. This is not because that these models have better performance for simulating exact term structure, but we can utilize these models for pricing some sophisticated derivatives, says Asian-like interest rate derivatives.

Before surveying the term structure model, I will introduce the underlying mathematics model first. The numerical methods applying to these term structure models will also be introduced in this chapter.

2.1.4.1 The Ornstein-Uhlenbeck Process

The Ornstein-Uhlenbeck process has the Ito differential,

$$dX = -kXdt + \sigma dW$$

where $k, \sigma \geq 0$. Given $X(t_0) = x_0$, it can be shown that

$$E[X(t)] = e^{-k(t-t_0)}E(x_0)$$

$$Var[X(t)] = \frac{\sigma^2}{2k}(1 - e^{-2k(t-t_0)}) + e^{-2k(t-t_0)}Var[x_0]$$

for $t_0 \leq s \leq t$. It can be shown that $X(t)$ is normally distributed if x_0 is a constant or normally distributed while X_t is also stationary. See figure 2.2 for a plot.

A good property of mean reversion is in the Ornstein-Uhlenbeck model. When $X > 0$, the dx term tends to be negative, pulling dx towards zero. If $X < 0$, dx tends to be positive, pulling X towards zero again. It is also an important property found in real world term structures. [3] For term structure models, the following version is used,

$$dX = k(\mu - X)dt + \sigma dW$$

[2] Using continuous compounding formula.

[3] The interest rate in the real world appears to pulled back to some long-run average level over time.

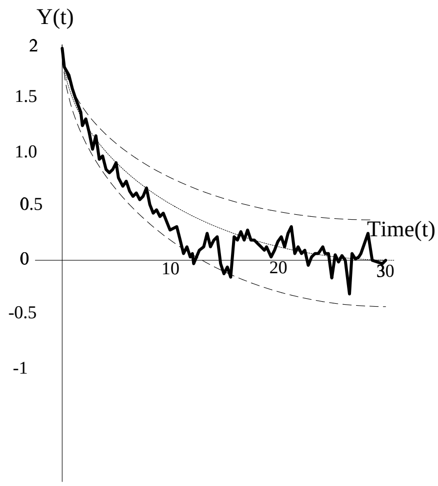


Figure 2.2: SAMPLE PATH OF ORNSTEIN-UHLENBECK PROCESS. Assume the underlying stochastic process is $dY = -0.15Y dt + 0.15dW$ with the initial condition $Y(0) = 2$. The envelope is used to show the standard derivation. This process will converge to a stationary distribution

where $\sigma \geq 0$. The mean and variance of this process are:

$$E[X(t)] = \mu + (x_0 - \mu)e^{-k(t-t_0)}$$

$$Var[X(t)] = \frac{\sigma^2}{2k}(1 - e^{-2k(t-t_0)})$$

2.1.4.2 The Vasicek Model

Two short rate models are introduced in these two subsections. The short rate will be the only source of uncertainty in these models. Both models, following the Ornstein-Uhlenbeck model, have some good properties in their discrete-time versions [9]. These properties will be helpful in solving Asian-like American-style derivatives.

The risk-neutral process in this model for r is

$$dr = a(b - r)dt + \sigma dz$$

where a, b and σ are constants [18]. This model incorporates mean reversion. The short rate is pulled to a level b at rate a. Superimposed upon this pull is a normally distributed stochastic term σdz.

Let $P(t, s)$ denotes the price at time t of a discount bond maturing at time s $(t \leq s)$. It can be shown that the yield to maturity, $R(t, T)$,[4] will follow

[4] It as internal rate of return at time t on a zero-coupon bond maturing date at time $s = t + T$

$$R(t,T) = \frac{-1}{T}\log P(t,t+T), T > 0.$$

The relation between the short rate $r(t)$ and $R(t,T)$ is

$$R(t,T) = E_t(\frac{-1}{T}\int_t^{t+T} r(\tau)d\tau) + \pi(t,T,r(t)), \tag{2.6}$$

It may be noted that different term structure theories, like expectation hypothesis, market segmentation hypothesis and liquidity preference hypothesis, can all satisfy equation 2.6 with various specifications for the function π. Consult [15] for more information about term structure theories.

The fatal disadvantage in the Vacicek model is that this model can not fit today's term structure automatically. Even choosing the parameters judiciously, significant errors may be caused in some case when we try to fit the exact term structure. In the following subsection, another term structure model by Hull and White will be introduced that exactly matches the real world term structure.

2.1.4.3 The Hull and White Model

This term structure model [8] can be treated as an extension of the Vasicek model. The Ito differential of this model they suggest is

$$dr = (\theta(t) - ar)dt + \sigma dz$$

or

$$dr = a[\frac{\theta(t)}{a} - r]dt + \sigma dz$$

where a and σ are constants. This model is basically the Vasieck model with a time-dependent reversion level. At time t the short rate reverts to $\frac{\theta(t)}{a}$ at rate a. The $\theta(t)$ function can be calculated from the initial term structure:

$$\theta(t) = F_t(0,t) + aF(0,t) + \frac{\sigma^2}{2a}(1 - e^{-2at}) \tag{2.7}$$

where $F(t,T)$ denotes the instantaneous forward rate as seen at time t for a contract maturity at time T. $F_t(t,T)$ denotes the differential with respect to t. The last term in (2.7) are usually fairly small in practice, which implies that the drift of the process for r at time t is approximately $F_t(0,t) + aF(0,t)$. This shows that on average r approximately follows the slope of the initial instantaneous forward rate curve.

These two short rate models are normally distributed with some good properties described before. Hull and White provided a general two steps tree-building procedures [10], which can be used to represent some one-factor term structure models.

2.2 Derivatives Basics

In this section, I will introduce fundamental knowledge on derivatives. This includes the payoffs of various derivatives, like *standard options, exotic options and interest rate options.*

2.2.1 Option Basics

An option, as the name implies, is the right to buy or sell the underlying asset for a limited time span with a specific price. Generally speaking, the options can be classified into three groups:*call options, put options* and the combination of the above two. Call options gives the holder the right to buy the underlying asset with a specific price at some certain time, while put options give the holder the right to sell it. The price which holder can buy or sell something is called the *exercise price* or the *strike price.* The date on the contract is known as the *expiration date*, *exercise date* or *maturity.*

The options can also be classified based on the time period in which they can be exercised. An *American option* can be exercised at any time up to maturity; in contrast, a *European option* can be exercised only at maturity. Thus, an American option gives all the advantages that a European option possesses, plus the advantage of early exercise. For this reason, the value of American options is at least as great as that of European ones, other conditions being equal.

There are two sides to every option contract. On the one side is the investor who take the long position (i.e., he buys the option), while on the other side is the investor who takes the short position (i.e., he sells the option).

2.2.2 Payoffs on Standard Options

An option provides its holder the right of gaining benefit without any obligation. Options will be exercised only when the best choice for the holder to gain maximum benefit [5]. Let me illustrate the standard European option as a example. Assume the value of the underlying asset is S, the strike price is X, and the premium of option is represented by O. Then the payoff for the long position at expiration is $\max(0, S-X)$ for call options; $\max(0, X-S)$ for put options. So the profit for a long position in call options at expiration is approximately

$$\max(0, S-X) - O$$

The profit for a long position in put options is approximately

$$\max(0, X-S) - O$$

[5]See page 3, we assume that all individual will behave rationally, and they will try to maximize their benefit as possible.

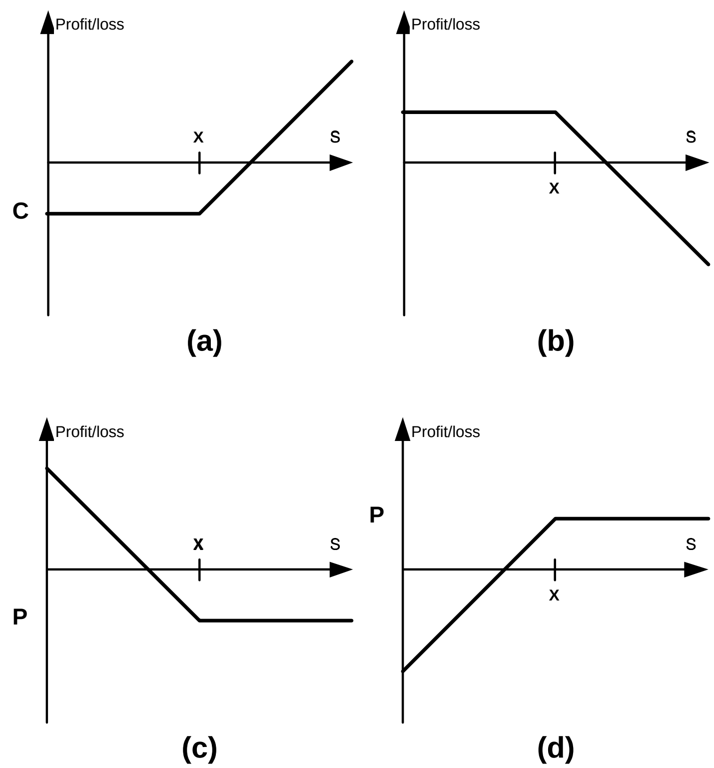


Figure 2.3: PROFIT/LOSS OF OPTIONS. (a) Long a call. (b) Short a call. (c) Long a put. (d) Short a put.

So the profit for a short position in call options is

$$-(max(0, S - X) - O) = min(0, X - S) + O$$

while the profit for a short position in put options is

$$-(max(0, X - S) - O) = min(0, S - X) + O$$

Figure2.3 illustrates profit/loss graphically.

2.2.3 Payoffs on Exotic Derivatives

Exotic derivatives have complicated payoffs than the standard derivatives. These sophisticated derivatives are usually designed by financial institutions to meet the requirements of their clients. Pricing these exotic derivatives are usually very hard because the payoff functions are usually ergodic (path-dependent.) I will introduce the payoff rules of some exotic derivatives related to this thesis.

2.2.3.1 Barrier Options

Barrier options are options whose payoff depends on whether the underlying asset's price reaches a certain level during a certain period of time. There are two types of barrier options: knock-out and knock-in options. A knock-out option is similar to a regular option except that when the underlying asset's price reaches a certain barrier, H, the option ceases to exist. The knock-in comes to existence only when the underlying asset price reaches the barrier.

2.2.3.2 Geometric Average-Rate Options

Just as the name suggests, the payoff of a geometric average-rate option depends on the geometric average of the underlying asset's values. Assume [6]

$$S_{avg} = \sqrt[N+1]{\prod_{t=0}^{N} S_t} \tag{2.8}$$

where N is the number of periods and S_t is the value of the underlying asset at period t. The payoff of a European-style call is

$$\max(S_{avg} - X, 0) \tag{2.9}$$

while the payoff of a put is

$$\max(X - S_{avg}, 0) \tag{2.10}$$

2.2.3.3 Arithmetic Average-Rate Options

They are very similar to geometric average-rate options except that we use arithmetic average instead of geometric average. Rewrite S_{avg} in (2.8) as

$$S_{avg} = \frac{\sum_{t=0}^{N} S_t}{N+1} \tag{2.11}$$

The payoff of call and put options are as (2.9) and (2.10) for European-style options.

2.3 Pricing Methods

Some important pricing methods used in this thesis will be introduced in this section. They include the Black-Scholes formula, tree simulation like Jarrow and CRR models for stocks, and Hull and White tree constructions [10] for term structure models.

[6] There are other types of geometric average option with different payoffs.

2.3.1 The Balck-Scholes Formula

In the early 1970s, Fischer Black and Myron Scholes made a major breakthrough by deriving a differential equation that must be satisfied by the price of any derivative security dependent on a non-dividend-paying stock. They solved this equation and obtained the closed-form solution for European call and put options on stock. This formula, known as the Black-Shcoles formula, is one of the most significant tools for pricing financial instruments. This formula will be treated as a benchmark for pricing some European options in this thesis.

2.3.1.1 Assumptions

The assumptions used to derive the Black-Scholes differential equation are listed below:

1. The value of the underlying assets follows the log-normal distribution.
2. The rate of return on stock, μ, and the volatility of stock price, σ, are constant throughout the option's life.
3. The short selling of securities with full use of proceeds is permitted.
4. The are no transaction costs or taxes. All securities are perfectly divisible.
5. No dividends are paid during the life of the derivative security.
6. No arbitrage opportunity.
7. Security trading is continuous.
8. The risk-free rate of interest, r, is constant and unchanged during the life of the security.

2.3.1.2 The Black-Scholes Differential Equation

By eliminating the random source of the underlying stochastic process [5], the final equations emerges as

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \tag{2.12}$$

where f is the price of a derivative security, S is the stock price, σ is the volatility of the stock price, and r is the continuously compounded risk-free rate.

2.3.1.3 The Closed Form Solution for Black-Scholes Formula

The closed form solutions for the price of European calls and puts by solving (2.12) can be described as below,[7]

$$C = SN(d_1) - Xe^{-rT}N(d_2)$$

$$P = Xe^{-rT}N(-d_2) - SN(-d_1)$$

where

$$\begin{aligned} d_1 &= \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \\ d_2 &= \frac{\ln(S/X) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \end{aligned}$$

The notations for the above equation are described as below.
$N(x)$ = Probability distribution function for standard normal distribution
σ^2 = Annualized variance of the continuously compounded return on stocks
r = Continuously compounded risk-free rate
T = The time to maturity

Closed-Form Solution for Geometric Average-rate Options. The price for geometric average-rate European-style options can be expressed easily with the above formula [12]. The stocks expected return is set at $(r-\sigma^2/6)/2$ with its volatility set at $\sigma/\sqrt{3}$. In others words, a geometric average-rate option can be treated like a regular option with the volatility set equal to $\sigma/\sqrt{3}$ and the dividend yield equal to

$$r - \frac{1}{2}(r - \frac{\sigma^2}{6}) = \frac{1}{2}(r + \frac{\sigma^2}{6})$$

2.3.2 Tree Models

The basic idea behind *tree model* is that we construct a tree simulating the movements of some underlying stochastic financial variables at discrete times steps $\triangle t$, $2\triangle t$, $3\triangle t$, ... If the toal time span is T and the periods is n, then $\triangle t = \frac{T}{n}$. In each period, we need to *calibrate* the first and the second moments. Use Figure 2.4 as the example. Assume the interest rate model follows (2.7) with $\theta(t)$ equal to zero. Then the variables in Figure 2.4 satisfy the following equations,

[7] C denotes the call price, P denotes the put price.

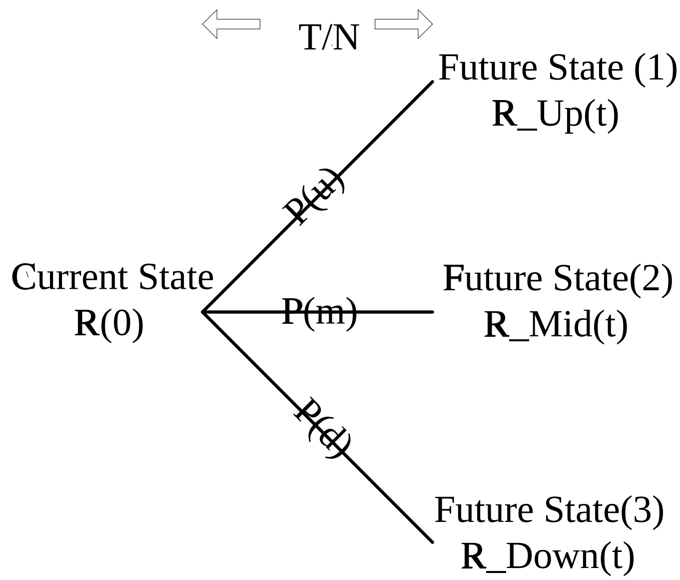


Figure 2.4: A One Period Trinomial Tree Example. Assume the interest rate today is $R(0)$ in the current state. There are three possible future states at next time step. Each state represents the up, middle or down movement of the interest rate. $P(u)$, $P(m)$ and $P(d)$ denote the respective probabilities.

$$-a\triangle tR(0) = P(u)\text{R\_Up}(t) + P(m)\text{R\_Mid}(t) + P(d)\text{R\_Down}(t) - R(0) \tag{2.13}$$

$$\sigma^2 = P(u)[\text{R\_Up}(t) - R(0)M]^2 + P(m)[\text{R\_Mid}(t) - R(0)M]^2 + P(d)[\text{R\_Down}(t) - R(0)M]^2 \tag{2.14}$$

where $M = 1 + a\triangle t$,

$$1 = P(u) + P(m) + P(d) \tag{2.15}$$

and $0 \leq P(u), P(m), P(d) \leq 1$

Equation (2.13) and (2.14) calibrate the mean and the variance of the original stochastic process. Equation (2.15) is the basic axiom of probability. There are more unknown variables than the equations. This gives us chance add the other constraints to make the tree models satisfying other requirements.

Pricing the derivatives can be illustrated by dynamic programing as follow (use Figure 2.4 as example),

$$V(\text{Current\_State}) = P(u)V(F.S.1)D_1 + P(m)V(F.S.2)D_2 + P(d)V(F.S.3)D_3 \tag{2.16}$$

where V denote the value of the derivative at a specific state, $F.S.$ is the abbreviation of *Future State*, and D_i denotes the discount factor for each state.

2.3.3 A General Tree Building Procedure for Term Structure

This tree structure is a good approach for constructing no-arbitrage short rate models of the term-structure [10]. This approach, making use of the trinomial tree, is appropriate for models where there is some function x of the short rate r that follows a mean-reverting arithmetic process. The key element of this process is that it produces a tree that is symmetrical about the expected value of x. The Vasicek and Hull and White models, for example, will take advantage of this tree building procedure.

There are three types of sub-trees, illustrated at Figure 2.5, for the tree building procedure. Assume the length for each time step is $\triangle t$ and the variance for each time

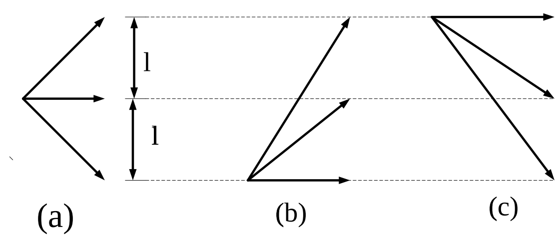


Figure 2.5: ALTERNATIVE BRANCHING PROCESS FOR INTEREST RATE TREE.

step is V. We can set the size of the interest rate step, $\triangle r$, at $\sqrt{3V}$. Then the tree can be built by the following two steps.

First stage: building a preliminary tree

Setting $\theta(t)$ in (2.7) and the initial value of r at zero suggest the following equation,

$$dr = -ardt + \sigma dz \tag{2.17}$$

Building an interest rate tree for (2.17) is the goal for first stage. This can be illustrated an example. Define (i, j) as the node for which $t = i\triangle t$ and $r = j\triangle r$. Assume we need to compute the type-a sub-tree. Then we can set R_Up(t) to $R(0) + \triangle r$, R_Mid(t) to $R(0)$ and R_Down(t) to $R(0) - \triangle r$ in equations (2.13), (2.14) and (2.15). The solutions for *P(u), P(m) and P(d)* can be solved as

$$P(u) = \frac{1}{6} + \frac{j^2M^2 + jM}{2}$$

$$P(m) = \frac{2}{3} - j^2M^2$$

$$P(d) = \frac{1}{6} + \frac{j^2M^2 - jM}{2}$$

The probability for the type-b sub-tree can be solved by setting R_Up(t) to $R(0) + 2\triangle r$, R_Mid(t) to $R(0) + \triangle r$ and R_Down(t) to $R(0)$to get

$$P(u) = \frac{1}{6} + \frac{j^2M^2 - jM}{2}$$

$$P(m) = -\frac{1}{3} - j^2M^2 + 2jM$$

$$P(d) = \frac{7}{6} + \frac{j^2M^2 - 3jM}{2}$$

For the type-c sub-tree, we set R_Up(t) as $R(0)$, R_Mid(t) as $R(0)-\triangle r$ and R_Down(t) as $R(0) - 2\triangle r$ respectively and obtain

$$P(u) = \frac{7}{6} + \frac{j^2M^2 + 3jM}{2}$$

$$P(m) = -\frac{1}{3} - j^2M^2 - 2jM$$

$$P(d) = \frac{1}{6} + \frac{j^2M^2 + jM}{2}$$

In order to make sure that the inequalities $0 \leq P(u), P(m), P(d) \leq 1$ hold, the range of j can be shown as follow,

$$\frac{-0.816}{M} \leq J \leq \frac{-0.184}{M}$$

and $-J \leq j \leq J$. A sample tree constructed by this step is illustrated in Figure 2.6.

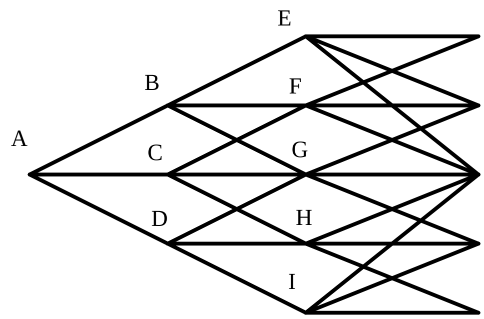
2.3.3.1 Second stage: calibration with the real term structure

Fitting today's term structure is the main goal of this stage. I will provide a shortcut method for fitting the spot rate curve. The exact method for this problem can be found in [10].[8] Assume the term structure function today is

$$0.08 - 0.05e^{-0.18t} \tag{2.18}$$

and the interest rate tree being calibrated is illustrated in Figure 2.6. Obviously, the average interest rate for the first period can be obtained by subtracting t in (2.18)

[8] Because of the symmetry of probability at the tree we construct in the first stage (see Figure 2.6), we can use this shortcut method instead of the original complex method.



Node	A	B	C	D	E	F	G	H	I
r(%)	0.00	1.73	0.00	-1.73	3.46	1.73	0.00	-1.73	-3.46
P(u)	0.167	0.122	0.167	0.222	0.887	0.122	0.167	0.222	0.087
P(m)	0.666	0.656	0.666	0.656	0.026	0.656	0.666	0.656	0.026
P(d)	0.167	0.167	0.122	0.122	0.087	0.222	0.167	0.122	0.887

Figure 2.6: SIMPLE TRINOMIAL TREE FOR THE HULL-WHITE MODEL. Parameters are set as follows, $a = 0.1$, $\sigma = 0.01$ and $\triangle t$ =one year.

with $\triangle t$. We can get the average interest rate for period-n by the following recursive formulas,

$$R_n = 0.08 - 0.05e^{-0.18n\triangle t}$$

$$S_n = R_n n - \sum_{i=1}^{n-1} S_i$$

The *calibrated* interest rate for each node can be obtained by adding the original interest rate of that node and the average interest rate of the period the node belongs to. The difference between the calibrated interest rate tree and the original interest rate tree is illustrated below(compared with Figure 2.6).

Node	A	B	C	D	E	F	G	H	I
r(%)	3.82	6.93	5.20	3.47	9.71	7.98	6.25	4.52	2.79

2.3.4 Binomial Tree Building Method

In this section, I will show how to build a binomial tree for the log-normal distribution. Just as described above, we calibrate the first and the second moments similar to equation (2.13) and (2.14). The generalized three-period binomial tree is illustrated in Figure 2.7.

Let P_u and P_d indicate the probability of upward and downward moving probability, respectively, then the equations can be described as follow,

$$R_f = \ln uP_u + \ln dP_d$$

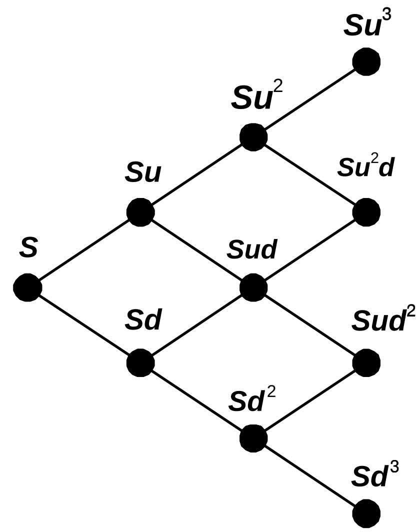


Figure 2.7: BINOMIAL MODEL FOR THREE PERIODS. Stock price movements over three time periods using binomial model. S is the stock price at period 0 and u and d are constants indicating the upward and downward ratios of the stock movement.

$$V = (\ln u - R_f)^2 P_u + (\ln d - R_f)^2 P_d$$

$$P_u + P_d = 1$$

where R_f denotes the risk-free rate and V denotes the variance of the stock return. Certainly, these four unknown variables can not be determined by the above three equations. For the convenience of pricing in tree models, Proper constraints are selected to achieve some good properties. Some examples are: the constraints we add in the CRR model is $u \times d = 1$, and $P_u = 0.5$ in the Jarrow's model.

Chapter 3

Combination of Pricing Tree with State Variables

In this chapter, I will investigate state variables into a *pricing tree*.[1] The systematic method about constructing special pricing trees for pricing general path-dependent options is also provided. It can be shown that most of these numerical pricing methods can be computed in *polynomial time*. This is a important property for an algorithm, otherwise, its usefulness will become limited if it takes so much time.

3.1 Discuss State Variables

In this section, I will discuss why we need state variables. In order to describe how the state variables work, I will give a simple example using the *forward-tracking method* on the CRR tree. Trying to keep the functionality for American-style options, I will show how to convert it to a *backward-tracking method*. See [6] for some similar examples.

3.1.1 Forward-Tracking Method on a Simple Example

Let's use barrier option as a example (See Figure 3.1). Assume the tree we simulated has just 4 periods long. The stock price at node A is denoted by S_A. Assume this option is a down-and-out call option. The payoff for a path start from A to B is

$$V(B) = \begin{cases} S_B - X, & if\ S_{\min} > H. \\ C, & if\ S_{\min} \leq H. \end{cases} \tag{3.1}$$

while H is the value of barrier, $S_{\min} = \inf_{0 \leq t \leq T}(S(t))$, S_B is the stock price at node B, and C is any constant. It is hard to price these options with the dynamic programming

[1]Pricing tree is used to indicate the tree structure described in Chapter 2 for pricing derivatives.

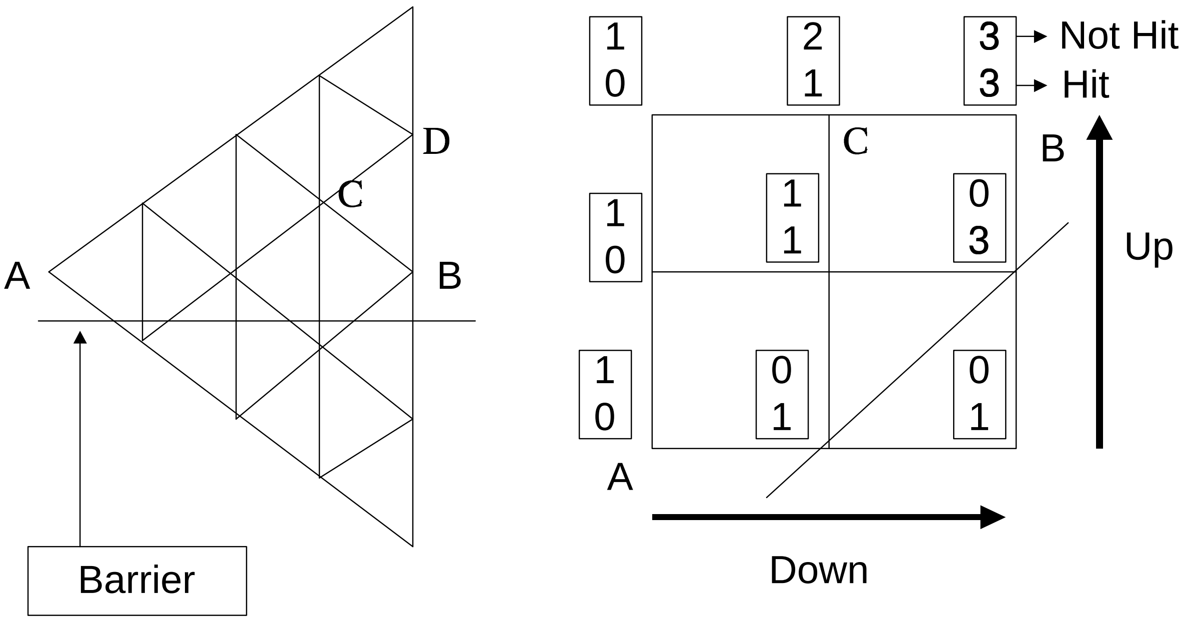


Figure 3.1: STATE VARIABLES IN THE CRR TREE MODELS The left part of this figure illustrates a CRR tree model with the barrier. The right part counts all the paths from A to B. The up movements in the right figure correspond to the up movements in the left figure while right movements correspond to the down movements.

methods in Chapter 2. This is because we don't know the derivative value at node B. The forward-tracking method below provides a good solution for this problem.

The barrier H, which is a constant, will be hit if the stock price starts form S and moves down for one period.[2] Each node has two states. The upper state represents the number of paths which start from A and don't hit the barrier, while the lower state is the number of paths which hit the barrier. The state variables for each node, says N_c, can be computed by Algorithm 1. In that algorithm, N_d is the node that

Algorithm 1 State Variables Evaluation

if $S(N_c) > H$ **then**
  N_c.State[0]=N_d.State[0]+N_u.State[0];
  N_c.State[1]=N_d.State[1]+N_u.State[1];
end if
if S(N_c)<=H **then**
  N_c.State[0]=0;
  N_c.State[1]=N_d.State[1]+N_u.State[1]+N_d.State[0]+N_u.State[0];
end if

can reach N_c with a up movement, while N_u is the node that can reach N_c with a

[2]The pricing tree used here follows the CRR model.

down movement, $S(N_c)$ represents the stock price at N_c, $State[0]$ represents the upper state, and $State[1]$ represents the lower one.

The total sum of the payoff of the barrier option for the paths from A to B, denoted by $V(A \to B)$, is $V(B) \times N_B.State[0]/(N_B.State[0] + N_B.State[1])$. The contribution of $V(A \to B)$ to the price of the barrier option is then $V(A \to B) \times P(A \to B)$ where $P(A \to B)$ is the probability of a path from A to B occurring. The undiscounted option price is finally

$$\sum_{\forall S \in \Re} V(A \to S) * P(A \to S)$$

where $\Re$ denotes the set of terminal nodes.

3.1.2 Backward-Tracking Method

The barrier option we described above can be solved in $O(n^2)$ time with the forward-tracking method. This is an acceptable performance for the European-style barrier options. But this algorithm can't price American-style options. For American-style options, we will exercise the options early if the benefit we get from early exercise exceeds the expected return by keeping them alive. The backward-tracking strategy provides the solutions for this problem.

Algorithm 2 A Backward-Tracking algorithm on Single Barrier Options

if $N_c \le H$ **then**
  N_c.State[0]=Don't care.;
  N_c.State[1]=$\frac{P_u \times N_u.State[1] + P_d \times N_d.State[1]}{R(N_c)}$;
end if
if $((N_c > H) \&\& (N_d \le H))$ **then**
  N_c.State[0]=$\frac{P_u \times N_u.State[0] + P_d \times N_d.State[0]}{R(N_c)}$;
  N_c.State[1]= $\frac{P_u \times N_u.State[1] + P_d \times N_d.State[1]}{R(N_c)}$;
end if
if $N_d > H$ **then**
  N_c.State[0]=$\frac{P_u \times N_u.State[0] + P_d \times N_d.State[0]}{R(N_c)}$;
  N_c.State[1]=$\frac{P_u \times N_u.State[1] + P_d \times N_d.State[1]}{R(N_c)}$;
end if

Using backward-tracking strategy with dynamic programing similar to (2.16) can we get Algorithm 2. N_c is the node we want to evaluate, and N_u and N_d can be reached form N_c by the up and down movements, respectively. $R(N_c)$ is the riskless yield for $\triangle t$ period time at node N_c. *Don't care* denotes that the state is useless.

The American-style options can be evaluated by adding the following statement at the bottom of Algorithm 2, [3]

$$N_c.State[0] = \max(N_c.State[0], S_{N_c} - X) \tag{3.2}$$

Constructing a pricing tree systematically will be the main focus in the next section. We will get some ideas about how state variables are defined and used.

3.2 Systematic Approach

State variables design and discussion on reasonable trading strategies are two main concepts in this section. With these two ideas we can construct different pricing trees for pricing general path-dependent derivatives[4] by just applying sophisticated dynamic programing techniques to these pricing trees.

3.2.1 Determination of State Variables

How many state variables does a node of a pricing tree need to keep for pricing arbitrary options? It goes without saying that for an n-periods binomial pricing tree, any terminal node V need to keep at most C^n_k states where k represents the number of up movements form the beginning node to V and n is the total number of the periods. That means we need a total of 2^n states at maturity because $\sum_{k=0}^{n} C^n_k = 2^n$. This pricing problem has been proved to be a $\sharp$ P-hard problem in [2]. So we can only hope to construct different pricing trees for specific derivative(s) because there should be no acceptable pricing tree algorithms for arbitrary derivatives.

The state variables we need on any node of the pricing tree depends on how many pay-off functions we need for that node. Before describing this idea, I would like to define an information set I_N for node N, as follows,[5]

Definition 3.2.1 *(Information set I_N)*
The information set I_N (for node N) includes the following information.

- *The information which we get at the derivatives issuing day, such as the value of the underlying asset, the volatility, the risk free rate, the strike prices, the number of the periods of this pricing tree and the maturity date.*
- *Specific information that belong to node N, like the interest rate at that node and the value of the underlying asset at that node.*

[3]Assume the options holders lose the right to exercise the options early if S has hit the barrier.

[4]There are still some derivatives, like Asian options, that can not be priced by simply applying these tricks. The technique of reducing the complexity for pricing these derivatives will be discussed later.

[5]This definition is incomplete due to some facts I will introduce in the future. See page 27 for the modification.

- *User's assumptions.*
- *(For Backward tracking method) The state values we can get from the nodes of next period by dynamic programming.*
- *Nothing else belongs to I_N.*

Define *proper pay-off function* for node N as follows,

Definition 3.2.2 *(Proper pay-off function for node N)*
A pay-off function is said to be proper if we can get a constant value by applying I_N to that pay-off function.

We can show a simple example about the proper pay-off function. If there is no user's assumption in set I_N, then (3.1) is *not* a proper pay-off function since we can't determine the relationship between S_{min} and H. But it will become a proper pay-off function by adding a user's assumption $S_{\min} > H$ to I_N.

After defining proper pay-off function, we can define the *node pay-off function set* for Node N as follow,

Definition 3.2.3 *(Node pay-off function set F_N)*
A node pay-off function set F_N for node N satisfies the following constraints.

- *All the elements in F_N must be proper pay-off functions for node N.*
- *Any pay-off value generated by any path reaching node N can be determined by one and only one pay-off function which belongs to F_N.*

Clearly, the number of state variables we need for node N can be described as $|F_N|$. We can use the node C in Figure 3.1 as a example. Assume that we are pricing a European-style single barrier option. We can construct a set F_C including the following proper pay-off functions.

- $\frac{V_D.State[0]*P_u}{R(C)}$
  User's assumptions$\rightarrow$ The paths don't hit the barrier, and they will reach D.
- $\frac{V_B.State[0]*P_d}{R(C)}$
  User's assumptions$\rightarrow$ The paths don't hit the barrier, and they will reach B.
- $\frac{V_D.State[1]*P_u}{R(C)}$
  User's assumptions$\rightarrow$ The paths hit the barrier, and they will reach D.
- $\frac{V_B.State[1]*P_d}{R(C)}$
  User's assumptions$\rightarrow$ The paths hit the barrier, and they will reach B.

where $R(C)$ is the riskless yield to maturity at node C, $V_i.State[0]$ denotes the payoff at node i if the historical path doesn't hit the barrier, $V_i.State[1]$ represents the payoff at node i if the paths hit the barrier, P_u and P_d represent the probabilities of upward and downward movements, respectively. Since we need four proper pay-off functions for describing the pay-off at node C, we will need four states for keeping all possible values at node C.

3.2.2 Reasonable Trading Strategies

Before discussing this issue, let's see a simple example about a two periods pricing tree used in pricing a standard call option. For simplicity, the risk-free rate is set to zero, the probabilities for up movement and down movement are equal to 0.5, and the exercise price is equal to 102. The value of underlying asset for each node is set arbitrarily so that you needn't make so much effect checking this example.[6] This example is illustrated in Figure 3.2. The state variables in the left part of the figure can be trivially described as follows,

$$N_c = \begin{cases} \frac{N_u+N_d}{2}, & European-style \\ \max(S_{N_c}-X, \frac{N_u+N_d}{2}), & American-style \end{cases}$$

where N_c is the node we want to evaluate, N_u and N_d can be reached form N_c by an up or a down movement, respectively. S_{N_c} is the value of the underlying asset at node N_c.

The difference between the pricing tree in the left part of Figure 3.2 and that in the right part of that figure is that node the X in the right part will contain one more state variable than that in the left part. Let me give definitions for the two states of node X. The upper state of X is the value of node X if S moves up (i.e., $S = 110$ in the next period), while the lower state of X is the value of the node X if S moves down (i.e., $S = 110$ in the next period). For European-style options, the value of the upper state is equal to $0.5 * 8$, where 0.5 is the probability of the up movement and 8 is the benefit by exercising the call option. The value of the lower node is equal to zero since nobody will exercise the call option when $S = 100$. Then we can get the same value as we computed in Figure 3.2(A). This can be described by the following formula,

$$OptionValue = (X.State[0] + X.State[1] + V(Y)) * 0.5$$

where X.State[0] and X.State[1] represent the up and down states of node X and $V(Y)$ denotes the state value for node Y. For American-style options, we will just need to check whether we can make more benefit by early exercise. It can be shown that we will exercise the call option early if S reaches the lower state of node X. The value for the lower node is equals to $3*0.5$, where 3 is equal to the benefit we exercise

[6]Notice that the tree may not satisfy the constraints in Chapter 2.

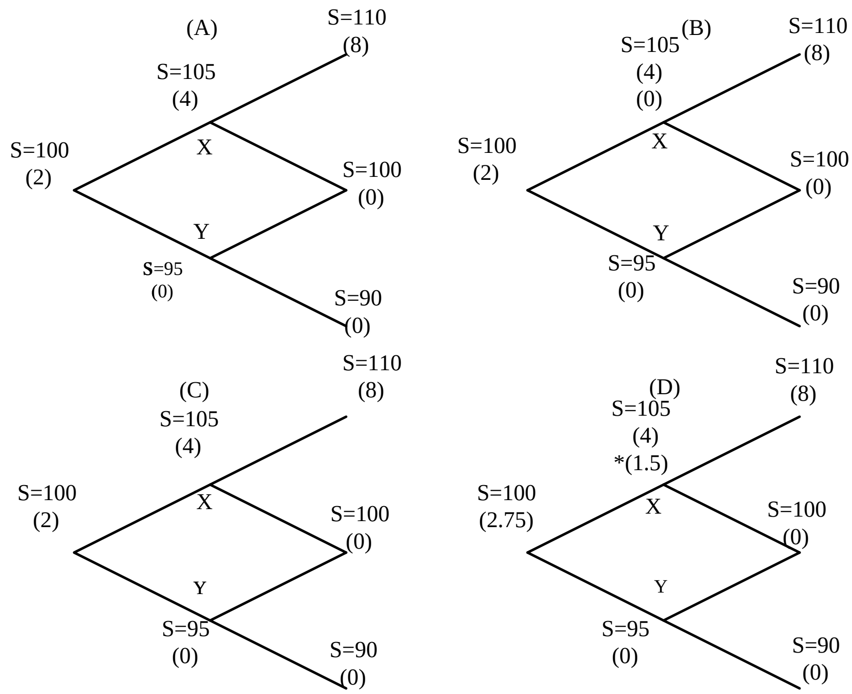


Figure 3.2: A Simple Example of a non-Reasonable Trading Strategy (A) Pricing European call options by applying a traditional pricing tree methods. (B) Pricing European call options by applying a multi-state-variables pricing tree. (C) Pricing American call options by applying a traditional pricing methods. (D) Pricing American call options by applying a multi-state-variables pricing tree. I only add one more state variable to the node X in the right part of this figure so that only the node X will contain two state variables. All the state variables are put in the parentheses. The value of the American call is different between (C) and (D). This is because the trading strategy is not reasonable in the node X of (D).

the options, 0.5 is the probability that S starts from X and moves down at period 2. We will get 2.75 by applying (3.3). This answer is different from the answer we may get by applying traditional pricing method.

Why could this happen? Formally speaking, our trading strategy for early exercise is not a *stopping time*. In other words, we have used the information which we shouldn't know at some stage when making early exercise decisions. In this case, we shouldn't know exactly S will move up or down from node X. So we can't make early exercise decisions with this information. Before introducing the theorem on the stopping time and trading strategy, I would give some definitions you need for reading this theorem.

Definition 3.2.4 *($\sigma-algebra$[13])*
A $\sigma-algebra$, says F, is a field which is closed with respect to countable intersections of its members, that is a collection of subsets of ω that satisfies

- $\phi,\ \omega \in F$
- $\alpha \in F \Rightarrow \bar{\alpha} \in F$
- $\alpha_1, \alpha_2 .....\alpha_n, ... \in F \Rightarrow \bigcup_{N=1}^{\infty} \alpha_N \in F$

The above definition is very important for probability theorem, but the detail of this issue is not our main point. You may just treat F_t, a $\sigma - algebra$, as an information set which contains all the information about the stochastic process (of the value of underlying asset) up to time t. For example, F_{today} is an information set containing all the information about the events happened up to today. But we can't know what will happen tomorrow with F_{today}.

A stopping time can be defined as follows,

Definition 3.2.5 *(Stopping time [13])*
A random time T is called a stopping time for some stochastic process $B(t)$, $t \geq 0$, if and only if for any t it is possible to decide whether T has occurred or not by observing $B(s)$ where $0 \leq s \leq t$. More rigorously, for ant t, the set $\{T \leq t\} \in F_t$, the $\sigma - algebra$ generated by $\{B(s)$, where $0 \leq s \leq t\}$.

Stopping time can be treated as decision making by simply observing the information set F_t, and we can make a decision by the information we know up to now. The following theorem describes an important property of reasonable trading strategies.

Theorem 3.2.6 *(Rational trading Strategies)*
A trading strategy must be a stopping time. That is to say, all trading behaviors B_t must be predictable by observing F_t.

This theorem can be explained more intuitively as follows. The decision we have to make today[7] should only depend on the information we can get up to now. For example, we can't decide a trading strategy as follows,

> I would like to early exercise the options today if the stock price will fall under the exercise price at maturity, otherwise, I would hold the options.

The reason is that we can't get *predictable* trading behavior by applying above trading strategy.

Do you find the error I make on purpose in Figure 3.2? The value of European-style options is the same whether by applying the traditional pricing tree or by applying the multi-state variables version. This is because we don't make any trading decisions during the life of the options. So violation of theorem 3.2.6 has no influence on the value. But it does influence the value of American-style options since we can't know exactly whether S will go up or down at node X.

Theorem 3.2.6 gives us some constraints in defining the state variables. To satisfy this theorem, one item, *User's Assumptions*, in definition 3.2.1 needs to be modified as follow,

[7] The decision is to exercise the options early or not in the above example.

- User's Assumptions must be determined by F_t, where t is the time that we could reach the node.

3.2.3 Creating Proper Recursive Formulas

The tricks about evaluating the state variables by recurrence are discussed as follow.

Definition 3.2.7 *(Action set)*
An action set A_N is composed of all actions that the underlying asset value S, which starts from node N, could take during the next period. For examples, A_N contains the up movement and the down movement in a binomial tree model.

With this definition, we introduce the following notations. Assume a is an action that belongs to A_N. Then N_a is the node that will be reached in the next period by taking action a. $P(a)$ is the probability that this action would be taken. Assume the state we want to evaluate is u (at node N). $S(N_a, u)$ is the state at node N_a that may be reached from state u by taking action a, and $Value(i)$ is the value for state i. Then the value of u can be calculated as follows,

$$Value(u) = \sum_{a \in A_N} P(a) * Value(S(N_a, u))$$

3.2.4 Creating a Sample Pricing Tree

I will give a simple example on constructing a pricing tree and an algorithm for pricing a specific derivative below. General examples will be introduced in the next chapter. The option I use here is the barrier option of (3.1).

1. Determine the proper pay-off functions we need for each node.
   - For the nodes at maturity, we need two proper pay-off functions described in (3.1). We need one proper pay-off function for the node which the value is below the barrier.
   - For the nodes we may reach before maturity, we divide them into two classes.
     - The value of that node is below or equal to the barrier:
       This case is trivial, since the path of the underlying asset value must hit the barrier at least once. We just need one proper pay-off function to describe the pay-off when hitting the barrier.
     - The value of that node is above the barrier:
       Two proper pay-off functions are required in this case. This is because there may be two kinds of paths that may reach this node. Some paths might hit the barrier before reaching this node, while others don't. The pay-off for these two kinds of paths are different. So we need two proper pay-off functions.

2. Design proper recursive formulas for evaluating the value of each state variable. See Algorithm 2.

3. Implement this algorithm by programming.

Chapter 4

Pricing Geometric and Arithmetic Interest Rate Average-Rate Derivatives

Some complex path-dependent derivatives, which are hard to solve by applying the algorithms in Chapter 3, will be solved in this chapter. They include geometric average-rate options and arithmetic average interest rate options. The pricing methods for arithmetic average-rate option, whose underlying asset value follows the log-normal distribution, will be introduced in the next chapter.

Besides giving the solutions to these derivatives, there are two important phenomena worth discussing. One is the early exercise property, and the other is the characteristics of the convergence behavior. Some reasonable explanations will be given for these unexpected properties.

4.1 Some Background Knowledge

Some background knowledge, like the early exercise property of call options and the convergence characteristics of pricing trees [17] is surveyed in this section. Discussions on these properties will give us more insights into these derivatives.

4.1.1 The Early Exercise Property

Before discussing the properties of early exercise, let's list some needed definitions.

Definition 4.1.1 *(Martingale)*
A stochastic process $\{X(t), t \leq 0\}$ *adapted to a* $filtration$ F *is a martingale if for any* t *it is integrable,* $E|X(t)| < \infty$*, and for any* $s < t$
$E(X(t)|F_s) = X(s)$

In a risk-neutral environment without arbitrage opportunity, the following equation holds [11],

$$\frac{S(t)}{A(t)} = E_t^Q[S(T)/A(T)] \tag{4.1}$$

where $S(t)$ is the value of the underlying asset at date t, $A(t)$ represents the value of the money market account at date t, E^Q denotes that we compute the expected value by equivalent martingale probabilities. In simple words, the stochastic process of discounted stock price follows a martingale process.

The early exercise property for the standard call option can be described as follows.

Theorem 4.1.2 *(Early Exercise Property of Standard Call Options)*
Given no dividends on the underlying stock and positive interest rates, a standard American call option will never be prematurely exercised, so an American option will be priced the same as a European option.

The following equations can be derived by applying Jensen's inequality[1] and (4.1),

$$A.C \geq E.C. = E^Q(\frac{\max(0, S_T - K)}{R}) \geq \max(0, E^Q(\frac{S_T}{R}) - \frac{K}{R}) = \max(0, S_0 - \frac{K}{R})$$

where "A.C." is the value of a standard American call, "E.C." is the value of a standard European call, S_t is the value of underlying asset at time t, K is the strike price, and R is the risk-free yield form today ($t = 0$) to maturity. If we exercise the option immediately, the payoff is $S_0 - K$. Since the following inequality holds under the assumptions in Theorem 4.1.2,

$$S(0) - \frac{K}{R} > S(0) - K$$

an option holder would like to keep or sell the option than exercise the option early.

Because different payoff functions are used by various sophisticated options, the property in Theorem 4.1.2 may not hold general. The discussion on this phenomenon will be postponed.

4.1.2 Convergence of Pricing Trees

The oscillation of the values produced by the pricing tree are discussed by Edward Omberg [17]. He argues that the oscillations are a by-product of approximating the stochastic process. He also shows a proof that this phenomenon will also occurs when pricing American-style options. Through the examples he used for pricing are standard options, this phenomenon also occurs on some path-dependent derivatives, like barrier options. The experiments provided in this chapter, however, show that the results produced by the pricing tree may converge monotonically on some derivatives. And this property is also verified by applying the combinatorial method.

[1] The interest rate is constant here.

4.2 Pricing Geometric Average-Rate Options

For pricing geometric average-rate options, a sophisticated design for the pricing tree is introduced here. The figure for describing the early exercise behavior is also illustrated here. Some explanations will be given for this phenomenon. A much faster $O(n^3)$ combinatorial approach is then introduced for European-style options. As argued in [16], this type of algorithm is useful for pricing European-style geometric average-rate options which have non-standard payoff functions.

4.2.1 Generation of Pricing Tree

Constructing a proper pricing tree for geometric average-rate options is harder than the examples given in Chapter 3. Selecting a proper pricing tree is the key step to this problem. The underlying model we used in this case is the CRR binomial tree. We construct this pricing tree following the approach in Chapter 3.

Determining the proper payoff functions

Define $N(i,j)$ as the node of the pricing tree for which i is its time and j is the number of down movements needed to reach it. Assume S is the value of the underlying asset and u is the upward factor. Then the maximum geometric sum by path from time 0 to time i is $S^i u^{\frac{i(i+1)}{2}}$, while the minimum geometric sum is $S^i u^{-\frac{i(i+1)}{2}}$. With the properties provided by the CRR model,[2] the set of all possible geometric sums at time i is

$$G_i = \{S^{i+1}u^{\frac{i(i+1)}{2}}, S^{i+1}u^{\frac{i(i+1)}{2}-2}, S^{i+1}u^{\frac{i(i+1)}{2}-4}, \ldots, S^{i+1}u^{-\frac{i(i+1)}{2}+2}, S^{i+1}u^{-\frac{i(i+1)}{2}}\}$$

Note that the sum take the form of $S^{i+1}u^k$ for some integer k. For a node at time i, the payoff function for an American-style geometric average-rate call is

$$\max(\sqrt[i+1]{S^{i+1}u^k} - X, D) \tag{4.2}$$

where X is the exercise price and D is the value if the options is kept alive, the number of states for any node at time i is at most $\frac{i(i+1)}{2}+1$ since $|G_i| = \frac{i(i+1)}{2}+1$. The required space for the whole is therefore $O(n^3)$, and the computation time is $O(n^4)$.

It should be clear that some states are not necessary. For example, the state that represents $S^{i+1}u^{\frac{i(i+1)}{2}-2}$ is useless for node $N(i,0)$ because there can be no paths that reach $N(i,0)$ with geometric sum $S^{i+1}u^{\frac{i(i+1)}{2}-2}$. We can therefore cut useless states instead of keeping exactly $|G_i|$ states for each time-i node. The experiments

[2] $ud = 1$. d is the downward factor.

data listed in this section show the savings with this idea, respectively. Formally, the maximum and the minimum geometric sums at $N(i,j)$ can be narrowed down to

$$\begin{aligned} N_{\max}(i,j) &= S^{i+1}u^{i(i+1)/2-j(j+1)} \\ N_{\min}(i,j) &= S^{i+1}u^{-i(i+1)/2+(i-j)(i-j+1)} \end{aligned}$$

The set composed of the necessary states for $N(i,j)$ can be described as

$$G_{N(i,j)} = \{A : A \in G_i, A \leq N_{\max}(i,j), A \geq N_{\min}(i,j)\} \tag{4.3}$$

Creating proper recursive formulas

For the terminal nodes $N(n,j)$, where n is the number of periods, the call value for each state of $N(n,j)$ is

$$\max(\sqrt[n+1]{S^{n+1}u^k} - X, 0) \tag{4.4}$$

where k represents the exponent of u for that state. For non-terminal nodes, the value D for state $S^{i+1}u^k$ at $N(i,j)$ is

$$(P_u \times V_{N(i+1,j)}(S^{i+2}u^{k+i-2j+1}) + P_d \times V_{N(i+1,j+1)}(S^{i+2}u^{k+i-2j-1}))/R \tag{4.5}$$

where $V_n(s)$ represents the option value for state s at node n and R is the one-period risk-less return. American-style options and put options can be obtained by simply modifying the above formula.

4.2.2 The Combinatorial Method

European-style geometric average-rate options can be priced by a much faster algorithm. Since the property $P_u = P_d = 0.5$ is useful here, the Jarrow-Rudd tree model is employed here.

The number of paths of length n having the same geometric average is precisely the number of (unordered) partitions of some integer into unequal parts none of which exceeds n. This claim can be verified as follows. Let $q(m)$ denote the number of such a partition of integer m. Any legitimate partition of m, say $\lambda \equiv (x_1, x_2, \ldots, x_k)$, then satisfies $\sum_i x_i = m$, where we impose $n \geq x_1 > x_2 > \cdots > X_k > 0$ for convenience. Now, interpret λ as the path of length n that makes the ith up move at $n - x_i$. Each up move at step $n - x_i$ contributes x_i to the sum m. This path has a terminal geometric average of $SM^{1/(n+1)}$, where

$$M \equiv u^m d^{n(n+1)/2-m}$$

in which the ith up move contributes u^{x_i} to the u^m term.

It can be shown that, in fact,

$$(1+x)(1+x^2)(1+x^3)\cdots(1+x^n) = 1 + \sum_{i=0}^{n(n+1)/2} q(i)x^i.$$

The probability for each path is 2^{-n} in this model. So the derived value is the present value of $\sum_{i=0}^{n(n+1)/2} 2^{-n} q(i) \max(SM^{\frac{1}{n+1}} - X, 0)$ for the call. Since the $q(i)$ can be computed in $O(n^3)$ time, pricing European-style options can be solved in time proportional to n^3.

4.2.3 Experimental Data

Experimental data about pricing using trees and combinatorial methods is illustrated here. Simple discussion on the computation time, the convergence of the pricing value, and the behavior for early exercise are also given. The European-style options analytical value is obtained by the Black-Scholes formula listed on Page 14.

4.2.3.1 European-style options

The options priced in this section use the following assumptions: The underlying asset value is 100, the strike price is equal to 100, the volatility is 20%, the risk free rate is 10%, and the time from the issuing day to maturity is 1 year. The analytical value is 6.769955. The result of pricing are illustrated in Figures 4.1, 4.2 and 4.3. Skipping useless states can save the computational time dramatically. The tree and the combinatorial methods converge quickly and correctly, but the combinatorial approach requires much less computational time. This is because the tree algorithm is an $O(n^4)$ algorithm, which is one degree higher than the combinatorial approach.

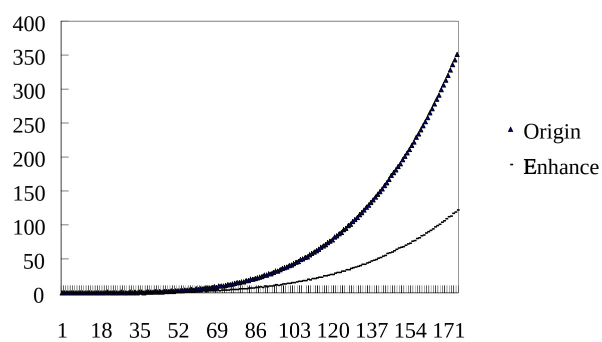


Figure 4.1: COMPUTATION TIME. The x-axis is the number of periods, and the y-axis is the computation time in seconds. “Origin” and “Enhance” denote the computational times whether we do not or do skip the useless states, respectively.

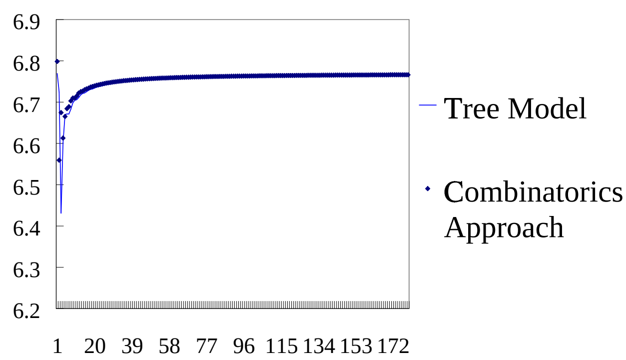


Figure 4.2: COMPUTATION TIME. TREE VERSUS COMBINATORICS. The x-axis is the number of periods, and the y-axis is the computation time in seconds.

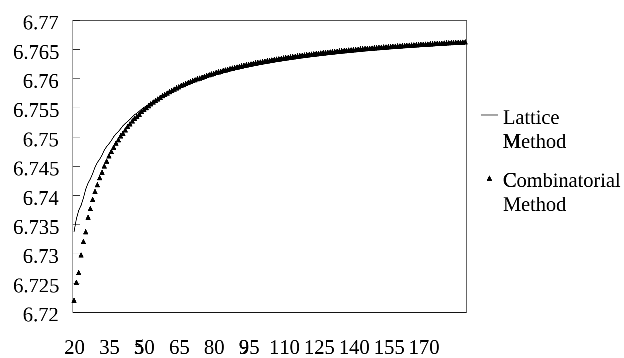


Figure 4.3: CONVERGENCE. The x-axis is the number of periods, and the y-axis is the option value.

Since the tree method performs well in pricing European-style options, it can be used for valuing American-style options.

4.2.3.2 American-style options

To price American-style options, only (4.2) needs to be added to the original pricing tree (for European-style options). The pricing results are illustrated in Figure 4.4. The pricing results also converge quickly and monotonically, which implies that this approach should be adequate for pricing American-style geometric average-rate options.

An interesting fact to observe is the early exercise strategy. For a standard call option, the holder may not exercise the option early (for non-dividend paying stock) or exercise the option when the value of the underlying stock is high. But the experimental results we get from pricing geometric average-rate options are nothing like

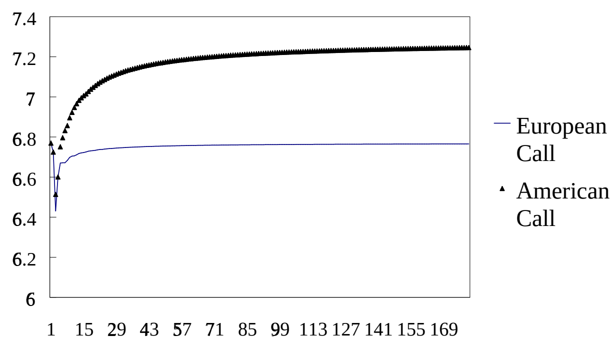


Figure 4.4: COMPARE THE OPTIONS VALUE The x-axis is the number of periods, and the y-axis is the option value.

that. The holder will not exercise the options just because the value of underlying stock is high. This interesting fact can be observed in Figure 4.5.

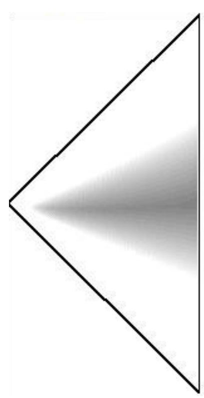


Figure 4.5: EARLY EXERCISE BEHAVIORS. The triangle denotes the tree. The darker the point is, the more likely the option will be exercised there.

Notice that the darkness for each point is not computed by the real probability measure. I compute the grey level of that point with the following formula:

$$V = 255 - \frac{|EG_{N(i,j)}|}{|G_{N(i,j)}|} \tag{4.6}$$

where $EG_{N(i,j)}$ is the states that the holders will exercise the options when $N(i,j)$ is reached. V denotes the grey level of that point (255 is pure white and 0 is pure black.)

Another interesting fact is that oscillations are not found in the results. It has be proven that oscillations are inherent when pricing standard American-style options with binomial tree. [17]. Edward Omberg claims that oscillations are a by-product of approximating the stochastic process. The reason that we get the different conclusion is that the payoff function of geometric average-rate options is far different from that of standard options; the geometric average-rate options are strongly path dependent. This phenomenon also happened when pricing other strongly path dependent derivatives, like arithmetic average-rate interest rate options. To converge monotonically or converge without oscillation means we lose the chance of getting a good estimate by taking the average of an upper and a lower bound.

4.3 Arithmetic Average-Rate Options

With the tree method described above, we can price arithmetic average-rate interest rate options under the Hull-White models [10]. The results we get from this model converge are monotonic but not quick convergence. The behavior of the early exercise is also different from what we observed before.

4.3.1 Generation of the Tree

Determine the proper payoff functions

Before describing this method, one interesting property is observed about the term structure tree model (see Figure 2.6). The difference of the short rates between adjacent nodes of the same period, say $\triangle R$, are all equal ($\triangle R = 1.73$ in this example). This is an important fact, since the possible arithmetic sums of interest rate at time i must be a subset of I_i. I_i can be described as

$$I_i \equiv \{S + \frac{i(i+1)}{2}\triangle R, S + (\frac{i(i+1)}{2} - 1)\triangle R, \ldots, S + \frac{-i(i+1)}{2}\triangle R\}$$

where S is the sum of interest rates from time 0 to i. It can be shown that $S = 9.02\%$ when $i = 1$ in this case. (see the table in Page 18) For each I_i, the proper payoff function for an American-style geometric-average call options is

$$\max(\frac{S + \triangle R \times k}{i+1} - X, D) \tag{4.7}$$

where X is the exercise price, D is the option value if we keep the option alive, and k is a given integer that makes $S + k\triangle R$ an item in I_i. It can be shown that the maximum number of states for an-i-time node need to keep is $i(i+1)+1$. The space is $O(n^3)$, and the computational time is $O(n^4)$.

Similar to what we do in pricing the geometric average-rate options, some unreachable states can be cut in the tree. Define $N(i,j)$ as the node for which $t = i\triangle t$ and $r = j\triangle r$. The set of necessary states for $N(i,j)$ can be described as

$$I_{N(i,j)} = \{A : A \in I_i; A \leq I_{N\max}(i,j); A \geq I_{N\min}(i,j)\}$$

where $I_{N\max}(i,j)$ and $I_{N\min}(i,j)$ are the maximum and the minimum arithmetic interest rate sums from the root to $n(i,j)$, respectively.

Creating proper recursive formula

For the terminal nodes $N(n, j)$, the call option value for each state of $N(n, j)$ is

$$\max(\frac{S + k\triangle R}{n+1} - X, 0)$$

where k is a given integer.

The value D (see (4.7)) for state $S + k\triangle R$ at $N(i, j)$ is[3]

$$\begin{aligned}(P_u \times V_{N(i+1,j+1)}(S + s + (k+j+1)\triangle R) + \\ P_m \times V_{N(i+1,j)}(S + s + (k+j)\triangle R) + \\ P_d \times V_{N(i+1,j-1)}(S + s + (k+j-1)\triangle R))/R\end{aligned} \tag{4.8}$$

where $V_n(S)$ represents the option value for S at node N, R is the discount rate for that period, and s is the calibrated interest rate at period i. American-style options and put options can be obtained by modifying the above formulas.

4.3.2 Experimental Results

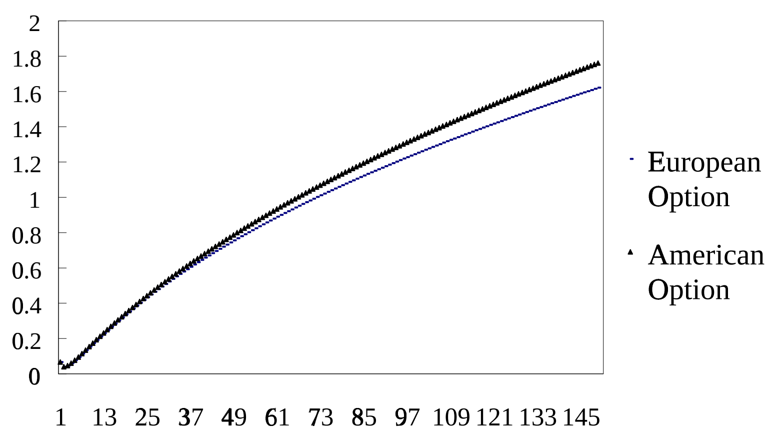


Figure 4.6: European and American Interest Rate Options under the Hull-White Model.

A numerical experiment is illustrated here. The parameters for the tree are: $a = 0.1$, $\sigma = 0.01$, the t-year continuous compounded zero coupon rate is $0.08 - 0.05e^{-0.18t}$, and the time from the issuing day to maturity is one year. (See [10] for the definitions of these parameters.) The payoff at maturity for this call option is defined as

$$100 \times \max(A(n) - X, 0)$$

[3]Modification is needed when the node is the root of the type B and C sub-trees (see Figure 2.5).

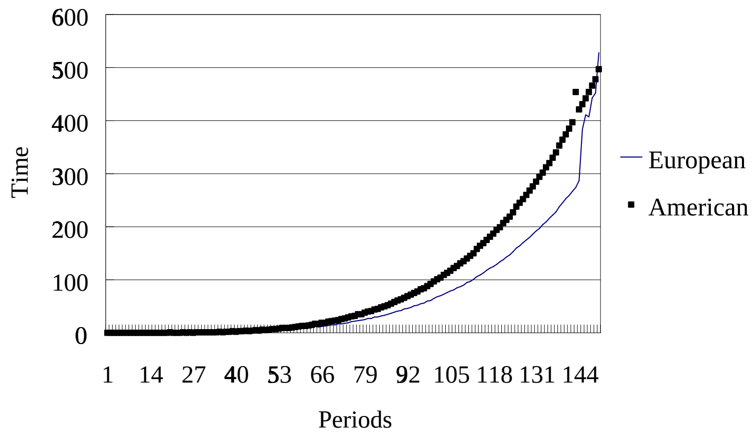


Figure 4.7: Computational Time of Pricing European and American Options

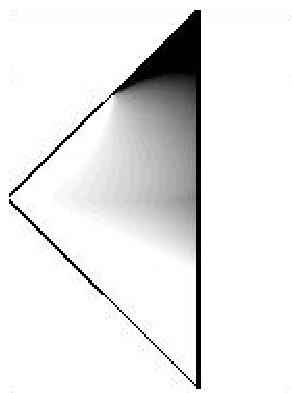


Figure 4.8: The Early Exercise Behavior. The triangle denotes the tree. The darker the point is, the more likely the option will be exercised there.

where X is the strike value, $A(i)$ is the arithmetic average of the interest rates from time 0 to time i. The payoff at time j if the option holder exercises the option is defined as $100 \times (A(j) - X)$. The numerical results of European and American options are illustrated in Figure 4.6.

The pricing results still converge monotonically but slowly. The computation time also grows dramatically. See Figure 4.7 for a plot. Since there is no oscillation in the pricing results, it may be hard to estimate the upper bound of the option value (in this case). Only the lower bound of the option value can be estimated.

The early exercise behavior for the arithmetic interest-rate options is shown in Figure 4.8.[4] This figure is similar as Figure 4.5 except that the options holders will

[4]The equation for the grey level of that point is computed by (4.6).

also exercise the options when the interest rate is high (for call options).

Chapter 5

Pricing Arithmetic Average-Rate Options

Pricing arithmetic average rate options is a well-known hard problem. This is because we can't derive a proper formula for describing the distribution of the sum of log-normal random variables. Various approximation approaches have announced. But there are at least two problems on most of these approaches. First, they may not be applicable to pricing American-style options. Secondly, most approaches fail to get acceptable results on some extreme cases [4]. A new lattice model, designed for pricing the arithmetic average-rate options is introduced in this chapter. This new approach can perform well and solve the American-style options accurately.

The method for building the lattice are discussed in first section. We try to calibrate the stochastic process of the underlying asset values. Experimental results are tabulated later. We will also examine the convergence of this algorithm. This algorithm also passes statistical tests of the extreme cases provided in [4].

5.1 Building a New Lattice Model

5.1.1 The Intuition

Let's begin with a simple problem, how many possible arithmetic sums may occur at node $N(i,j)$? [1]. There are C^i_j paths that would reach $N(i,j)$, which implies that they must be at most C^i_j different arithmetic sums. This is an unacceptable result since $\sum_{j=0}^{i} C^i_j = 2^i$, which implies totally 2^i states are needed for keeping all possible options values at period i. Can we decrease the number of states we need to an acceptable size? The answer is probably not, if we follows the current lattice models.

Another approach is to reconstruct a lattice model. We use the pesudo-polynomial technique to solve this problem. A new lattice is therefore constructed.

[1]Reference the definition in page 32

5.1.2 Building a New Lattice Model

The lattice here is a trinomial model. Redefine $N(i,j)$ as the node that has j-th biggest value at time i. The symbols $\mu(s), \nu(s), \omega(s)$ denote the three branches from s,

$$\mu(N(i,j)) = N(i+1,j)$$
$$\nu(N(i,j)) = N(i+1,j+1)$$
$$\omega(N(i,j)) = N(i+1,j+2)$$

Define $\triangle t = \frac{T}{n}$. $V(N)$ is the underlying value of node N, $M(N,\triangle t)$ and $Var(N,\triangle t)$ are the mean and variance at the next period if the current state is N, respectively, and $P_u(N)$, $P_m(N)$, $P_d(N)$ denote the up, flat and down probabilities from N. See Figure 5.1. Since we need to calibrate the first and second moments of the underlying asset values, the variables for any node N must satisfy the following equations:

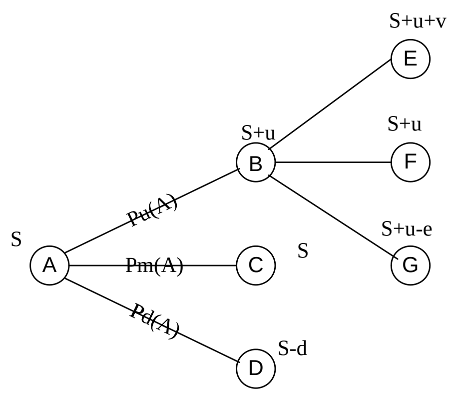


Figure 5.1: A New Lattice Model

$$V(\mu(N)) \times P_u(N) + V(\nu(N)) \times P_m(N) + V(\omega(N)) \times P_d(N) = M(N,\triangle t), \tag{5.1}$$

$$(V(\mu(N)) - M(N,\triangle t))^2 \times P_u(N) + (V(\nu(N)) - M(N,\triangle t))^2 \times P_m(N) + (V(\omega(N)) - M(N,\triangle t))^2 \times P_d = Var(N,\triangle t), \tag{5.2}$$

$$P_u(N) + P_m(N) + P_d(N) = 1$$

$V(N) = V(\nu(N))$ is imposed for simplifying algorithm design. Since $P_u(N)$, $P_m(N)$, $P_d(N)$ are probabilities, the following inequalities must hold,

$$0 \le P_u(N), P_m(N), P_d(N) \le 1$$

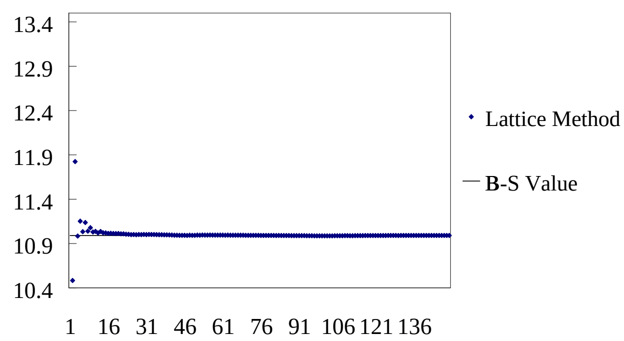


Figure 5.2: TEST ON STANDARD OPTIONS Use the lattice model on pricing the standard call options. The initial stock value is 100, the strike price is also equal to 100, the volatility is 0.2, the risk free rate is 0.06, and the time to maturity is equal to 1 year. The benchmark value derived from Black-Scholes formula is 10.989547

We finally impose the condition that the nodes combine.

For example, assume the initial node is A in Figure 5.1. Then u and d can be solved by imposing further that $u = d$. For the nodes whose underlying asset value is larger then S, like node B, the equation $e = u$ must hold. The value of v can therefore also be determined. [2] For the nodes whose underlying asset value is smaller than S, similar steps must be applied.

5.2 Experimental Results

First, we price a standard option with our method. It is a good benchmark to see if it works at least for the simplest problem. Figure 5.2 shows performance is good. The convergence is quick which implies that the lattice model can approximates the distribution of the underlying asset value well.

Let's test the convergence of this algorithm. The numerical data are the same as what in Figure 5.2. The pricing results of the European and the American-style call options are illustrated in Figure 5.3. We find that the pricing results of the European-style options converge quickly and stably. The converge speed of the American-style options are slower then the European-style options. But the results still converge *almost* uniformly and stably. So this approach should be a reliable approach in this case.

We will compare this algorithm with the similar algorithm announced by Hull and White [7]. These data are listed in Table 5.1. The numerical data about the Hull and White method and Monte Carlo simuations are copied from [1]. The number of the periods we use on the new lattice model is only 30. It takes almost 2 seconds

[2] You have to modify the up and the down displacements lightly if one of the probabilities we get by applying the above equations is negative.

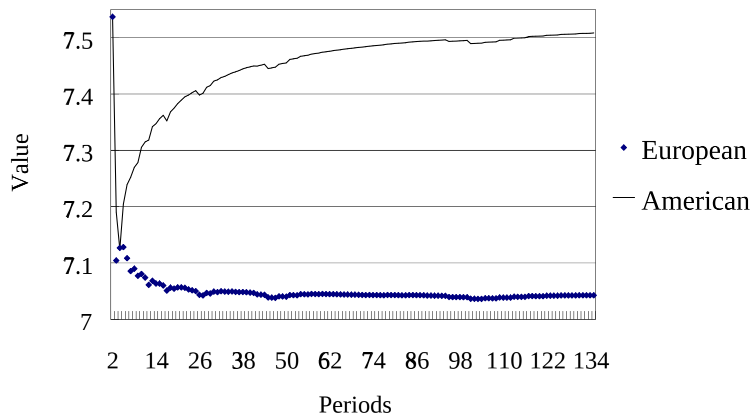


Figure 5.3: TEST ON EUROPEAN AND AMERICAN-STYLE ASIAN OPTIONS

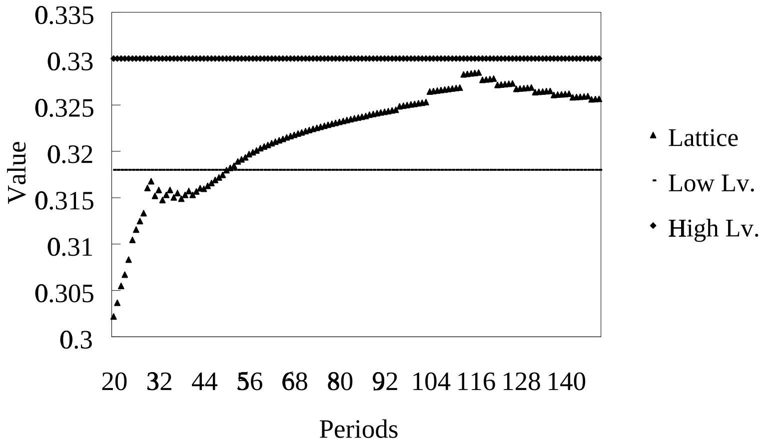


Figure 5.4: EXPERIMENT ON THIS NEW LATTICE MODEL(OPTION VALUES)

for a pentium-pro computer valuing a option. Most of the values computed by this algorithm are approximate to the value computed by Monte Carlo simulations. Only two value are out of the range of 95% confidential interval(These value are marked with "*").

To verify the convergence of this algorithm, we will test this algorithm by selecting one of the worst pricing result listed in Table 5.1. See Figure 5.4 for a plot. You may find that the algorithm still converge well. The pricing results approach the bench mark value when the number of periods is large. We also find that the pricing results are within the range of the 95% confidence interval when the number of period is larger then 50, which implies that the oscillations of the pricing results are acceptable.

The computation time grows dramatically when the number of periods go large. See Figure 5.5 for a plot. By the numerical results in Table 5.1, we know that this algorithm converge quickly. But we will get into great trouble if we try to seek more

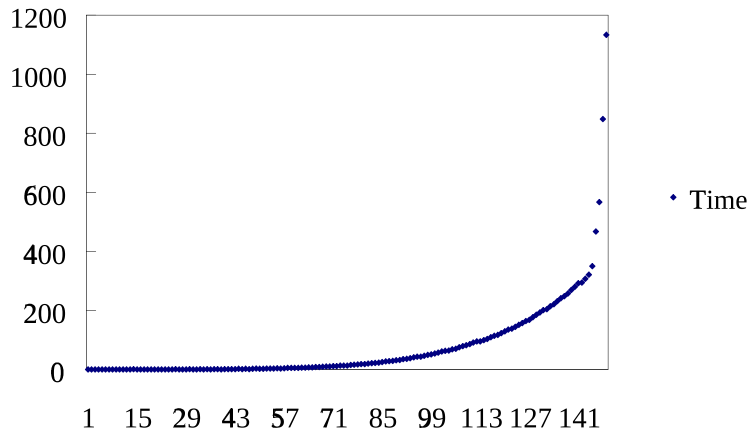


Figure 5.5: COMPUTATION TIME USED BY THE NEW LATTICE MODEL.

accurate answer. A tricky method for speeding up this algorithm might be useful here.

Another experiment will focus on some extreme cases mentioned in [4]. In this paper, The authors claim that some approximate algorithms may fail at extreme cases. We will test these extreme cases listed in Table 5.2 and show that our lattice model performs well at these cases.

Table 5.1: **The Value of Arithmetic Average-Rate Options Derivated by Various Algorithms.**

Maturity (Years)	Algorithm	Exercise Price=40	Exercise Price=45	Exercise Price=50	Exercise Price=55	Exercise Price=60
0.5	H-W	10.755	6.363	3.012	1.108	0.317
	M.C.	10.759	6.359	2.998	1.112	0.324
	S.D.	0.003	0.005	0.007	0.005	0.003
	A.(30)	10.754	6.356	2.997	1.104	0.317*
	Levy	10.765	6.386	3.024	1.105	0.313
1.0	H-W	11.545	7.616	4.522	2.420	1.176
	M.C.	11.544	7.606	4.515	2.401	1.185
	S.D.	0.006	0.008	0.01	0.009	0.007
	A.(30)	11.547	7.616	4.517	2.412	1.170*
	Levy	11.576	7.662	4.557	2.431	1.172
1.5	H-W	12.285	8.670	5.743	3.585	2.124
	M.C.	12.289	8.671	5.734	3.577	2.135
	S.D.	0.008	0.01	0.012	0.012	0.01
	A.(30)	12.284	8.674	5.750	3.585	2.118
	Levy	12.337	8.738	5.801	3.619	2.133
2.0	H-W	12.953	9.582	6.792	4.633	3.057
	M.C.	12.943	9.569	6.786	4.639	3.055
	S.D.	0.01	0.013	0.014	0.015	0.013
	A.(30)	12.944	9.577	6.786	4.625	3.045
	Levy	13.024	9.671	6.874	4.691	3.087

The initial underlying asset value is 50; the risk free rate is 10% per year; the volatility is 0.3 per year; averaging is between the beginning of the life of the options to maturity. H-W denotes the Hull and White algorithm based on 40 time steps and $h = 0.005$. Monte Carlo simulations are based on 40 time steps and 100,000 trials. A.(30) is our lattice method with the number of periods equal to 30. Levy denotes Levy's approach described in [14].

Table 5.2: **Testing the Lattice Model under Some Extreme Cases**

r	σ	T	$S(0)$	GE	$Shaw$	$Euler$	PW	TW	$MC10$	$MC100$	$S.E.$	$A.(30)$
0.05	0.5	1	1.9	0.195	0.193	0.194	0.194	0.195	0.192	0.196	0.004	0.193
0.05	0.5	1	2.0	0.248	0.246	0.247	0.247	0.250	0.245	0.249	0.004	0.246
0.05	0.5	1	2.1	0.308	0.306	0.307	0.307	0.311	0.305	0.309	0.005	0.306
0.02	0.1	1	2.0	0.058	0.520	0.056	.0624	.0568	.0559	.0565	.0008	0.056
0.18	0.3	1	2.0	0.227	0.217	0.219	0.219	0.220	0.219	0.220	0.003	0.218
.125	.25	2	2.0	0.172	0.172	0.172	0.172	0.173	0.173	0.172	0.003	0.172
0.05	0.5	2	2.0	0.351	0.350	0.352	0.352	0.359	0.351	0.348	0.007	0.351

The exercise price is 2.0, r is the risk-free rate, T is the life of the options from the issuing day to maturity, σ is the volatility, $S(0)$ is the initial price of the underlying asset, and $A.(30)$ denotes our method. The other approximation methods for comparison are: Geman-Eydeland (GE), Shaw, Euler, Post-Widder(PW) and Turnbull-Wakeman (TW). The benchmark values ($MC10$ and $MC100$) and the approximation values are copied from [4]. S.E. is the standard error, also from [4].

Chapter 6

Conclusion and Future Work

This thesis presents a systematic approach for pricing path-dependent derivatives. We also show how this approach works through some examples. Some interesting properties are found by reviewing the experimental data, such as the early exercise property and the convergence of the results. These properties might be worth further study. The pricing results on the arithmetic average interest-rate options converge monotonously but slowly. Speeding up the algorithm is also a topic worth studying. A sophisticated pricing tree model for arithmetic average rate options is also investigated. The experimental data show that this algorithm converges quickly and correctly, but the required computation time grows dramatically when the number of periods become large. A method for speeding up this algorithm will be very useful.

My troops are always in the front line, run the biggest risks, suffer the greatest loss. The price of victory is never cheap.
—by a Great Panzer General in Germany

Bibliography

[1] He Youn Cho and Hee Yong Lee "A Lattice Model for Pricing Geometric and Arithmetic Average Options." *The Journal of Financial Engineering*, Vol. 6, No. 3, pp. 179–191.

[2] Prasad Chalasani, Somesh Jha, and Isaac Saias. "Approximate Option Pricing." *Annual Symposium on Foundations of Computer Science*, 1996, pp. 244–253.

[3] Darrell Duffie. *Security Markets: Stochastic Models.* New York:Academic Press, 1988.

[4] Michael C. Fu, Dilip B. Madan, and Tong Wang. "Pricing Continuous Asian Options: a Comparsion of Monte Carlo and Laplace Transform Inversion Methods." *Journal of Computational Finance*, Winter 1998, pp. 49–74.

[5] Fischer Black, and Myron Scholes. "The Pricing of Options and Corporate Liabilities." *Journal of Political Economy* , 81, No. 3, pp. 637-654.

[6] John Hull. *Options, Futures, and Other Derivatives.* 3rd edition. Prentice-Hall, 1997.

[7] John Hull and Alan White. "Efficient Procedures for Valuing European and American Path-Dependent Options." *Journal of Derivatives*, Fall 1993, pp. 21–31.

[8] John Hull and Alan White. "Pricing Interest Rate Derivative Securities." *Review of Financial Studies,* , 3, No. 4, 1990, pp. 573-592.

[9] John Hull and Alan White. "One-Factor Interest Rate Models and the Valuation of Interest Rate Derivatives Securities." *Journal of Financial and Quantitative Analysis*, Vol. 28, 1993, pp. 235–254.

[10] John Hull and Alan White. "Numerical Procedures for Implementing Term Structure Models I: Single Factor Models." *Journal of Derivatives,* 2, No. 1, 1994, pp. 7–16.

[11] Jarrow and Turnbull. *Derivatives Securities.* International Thomson Publishing, 1996.

[12] A. Kemna and A. Vorst. "A Pricing Method Based on Average Asset Values," *Journal of Banking and Finance*, 14, pp. 113-129

[13] Fima C. Klebaner. *Introduction to Stochastic Calculus with Applications.* Imperial College Press,1998.

[14] E. Levy "Pricing European Average Rate Currency Options" *Journal of International Money and Finance,* 11 (1992), pp. 474–491.

[15] Yuh-Dauh Lyuu. *Introduction to Financial Computaion: Principles, Mathematics, Algorihms.* Not published yet. Link http://www.csie.ntu.edu.tw/̃lyuu for more information.

[16] Yuh-Dauh Lyuu. "Very Fast Algorithms for Barrier Option Pricing and the Ballot Problem." *The Journal of Derivatives,* 5, No. 3 (Spring 1998), pp. 68–79.

[17] Edward Omberg. "A Note on the Convergence of Binomial-Pricing and Compound-Option Models." *Journal of Finance,* June 1987, pp. 463–469.

[18] Oldrich Vasicek "An Equilbrium Characterization of the Term Structure." *Journal of Financial Economics*, 1977, pp. 177–188.