A Comparison of Different Option Pricing Methods

Bill Huang

May 2017

1 Introduction

An option is the financial derivative that gives people the right to buy or sell an asset by a certain date (known as the expiration date or maturity) for a pre-determined price (known as the strike price). A call option gives people the right to buy and a put option gives people the right to sell. The option only gives people the right, but not the obligation, to buy or sell the asset. The buyer can decide whether to make the transaction, and this is referred to as exercising the option. It is optimal to exercise the call option if the strike price is below the spot price of the underlying assets since one can gain immediate profit by selling the assets, and the reverse holds true for the put option: exercise if the strike price is above the spot price of the underlying assets. If the option is not exercised under the range of date specified in the contract, the option expires. An European option can only be exercised at the maturity, while an American option can be exercised at any time prior to the expiration date. If an American option is exercised prior to maturity, it is referred to as early exercise. American option is more valuable than the European option since it gives the buyer more flexibility on exercising his right.

Option has been largely used for hedging or even speculating purpose after its introduction to the exchange market. Option pricing theory is important for helping buyers and sellers to value contracts that the outcomes depend on quantifiable uncertain future event. It also provides symmetry information to both the buyers and sellers about what a reasonable price should be for a certain option. However, it was not until 1973 that Fischer Black and Myron Scholes presented the first satisfactory equilibrium option pricing model which led to a simple closed form solution for the valuation of the European option on non-dividend paying stock or stock that pays continuous dividend proportional to the stock price under some "ideal" conditions [2]. In the same year, Robert Mertons extends the model in many different ways [8], and this is later known as the Black-Scholes-Merton (BSM) model. The model provides the major framework for later research in the option pricing. However, the pricing formula is derived by utilizing the itô process in solving the partial differential equation, which is not intuitive to many people without strong mathematical background. Some approximation techniques, such as tree-based algorithm and Monte Carlo simulation, use more elementary mathematics to achieve the approximate result. Moreover, there is no closed form solution for American option in general and European option with discrete dividends from the BSM model. The tree-based approximation algorithm and Monte Carlo simulation also help address these problems.

In year 1979, John Cox, Stephen Ross, and Mark Rubinstein presented the binomial tree algorithm in valuing option [5]. The tree consists a set of nodes, representing the possible future stock prices. At each node, the stock price can either go up or down [1] by a fixed amount that depends on volatility of stock price and the time between connecting nodes. As shown in Cox et al.'s paper, if we let the time between connecting nodes to be infinitely small, one can achieve the same valuation from BSM model for European option on non-dividend stocks. In the same year, Richard Rendleman and Brit Bartter also presented binomial tree algorithm but with different set of parameters that resulted in the same conclusion [9]. In 1988, Phelim Boyle presented an extension to the binomial model by allowing the stock price to go upward, downward, or stay the same in the next state (node), which is known as the trinomial tree [3]. Here, we are including a new free parameter into the model, enhancing the convergent rate. It means that with a trinomial tree, we can approximate the valuation from the BSM model with fewer steps than the binomial tree. Both binomial tree and trinomial tree are efficient in valuing European and American options on stock with non-dividend payment or continuous-dividend payment. However, when the dividends are discrete, then the tree is no longer recombining. The number of nodes grow exponentially as the number of dividend increases. Nelson Areal and Artur Rodrigues addressed the issue by presenting an adaptive binomial non-recombining tree algorithm [1], and the adaptive step can be easily extended to trinomial tree.

Different from tree-based algorithm that discretize the continuous distribution of stock price in a appropriate way, the Monte Carlo simulation method estimates the risk natural valuation of the option, which is a expectation so can be expressed in terms of integral, by the Weak Law of Large Number. The risk neutral valuation is equivalent to the result from solving the BSM partial differential equation on European option. Moreover, in the BSM formula, the price of underlying asset need to follow the lognormal distribution, but in Monte Carlo estimation, the price of underlying asset can follow any random process, for example, Poisson distribution or mix of different distribution. In 1977, Phelim Boyle presented a Monte Carlo Approach on valuing a European options with discrete dividend payment [4]. However, American put [2] was difficult to approach by Monte Carlo since one needs a way to estimate the future expected payoff so one can compare it to the pay-off from immediate exercise. Fortunately, the issue got resolved by the introduction of least square Monte Carlo (LSM) approach by Francis Longstaff and Eduardo Schwartz in

2001 [7]. Longstaff and Schwartz used a least square regression approach to estimate the conditional expectation of the payoff from continuing to keep the option alive. Different from tree-based algorithm, Monte Carlo techniques is flexible in the situation when multiple factors affect the price of the underlying asset jointly. One can simply incorporate them into the stochastic process to model the price of the asset.

The remainder of this article is organized as follows. Section 2 presents the BSM option pricing formula derived from the risk neutral valuation approach. Sections 3-5 discuss the accuracy and computational efficiency of tree-based algorithm and Monte Carlo simulation on European call option on non-dividend stock, European call option on stock with discrete dividend, and American put option on non-dividend stock respectively. Last, Section 6 summarizes the results and draw conclusion.

2 Derivation of Black-Scholes-Merton formula using risk neutral valuation

Many approximation techniques in option pricing, such as the tree-based algorithm and Monte Carlo, are all based on the risk neutral valuation in the BSM formula. Here, I will present how we can derive the BSM formula by the risk neutral valuation approach. First, let me introduce the "ideal conditions" that Black and Scholes introduced when they derived the option pricing formula. Some key conditions are: (1) the stock price follows a generalized Wiener process which produces a log-normal distributions for stock price between any two points in time; (2) the stock pays no dividends during the life of the derivative; (3) the risk-free, r, and the volatility, σ, are known and constant through time; (4) the option is "European" so it can only be exercised at the end of the expiration date; (5) No riskless arbitrage opportunities [3]. Now let me define the key variables that will be used throughout the paper. Let

- f = price of the option
- S_t = the current stock price at time t
- K = the strike price of the option
- r = the risk-free rate per annum (compounded continuously)
- σ = the volatility per annum
- T = the expiration date of the option
- $N(\cdot)$ = the cumulative density function for standard normal distribution
- $\phi(\cdot)$ = the probability density function for standard normal distribution

Under the risk neutral assumption, the equilibrium rate of return on all assets is equal to the risk free rates. Recall that the stock follows a log-normal distributions between any two point in time, then it follows that at maturity, $\ln S_T \sim \mathcal{N}\big(\ln S_t + (r - \sigma^2/2)(T -$

$t), \sigma^2(T-t)\big)$, which we can express S_T as

$$S_T = S_t e^{(r-\sigma^2/2)(T-t)+x\sigma\sqrt{T-t}}$$

where $x \sim \mathcal{N}(0,1)$, the standard normal distribution. Here, I will focus on the European call option, where the payoff at maturity is $\max(S_T - K, 0)$, then the price of value should be $f = e^{-r(T-t)}\max(S_T - K, 0)$ under the no arbitrage condition. Thus, the expectation for option price is (the random variable here is x)

$$\begin{aligned}
\mathbb{E}[f] &= \int_{-\infty}^{\infty} f\phi(x)\,dx \\
&= e^{-r(T-t)}\int_{-\infty}^{\infty} \max(S_t e^{(r-\sigma^2/2)(T-t)+x\sigma\sqrt{T-t}} - K, 0)\phi(x)\,dx
\end{aligned}$$

We need find the range for x that we can get rid of the max function, which we need to solve the inequality $S_t e^{(r-\sigma^2/2)(T-t)+x\sigma\sqrt{T-t}} > K$, and the solution is

$$x > \frac{\ln(K/S_t) - (r-\sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} := -d2$$

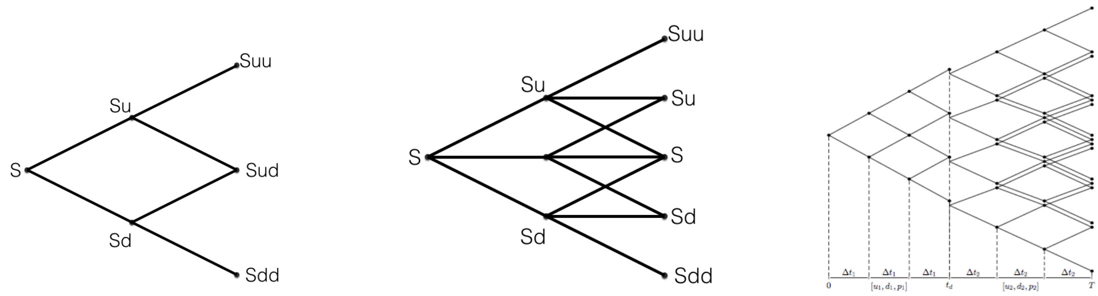
It follows that

$$\begin{aligned}
\mathbb{E}[f] &= e^{-r(T-t)}\int_{-d2}^{\infty} (S_t e^{(r-\sigma^2/2)(T-t)+x\sigma\sqrt{T-t}} - K)\phi(x)\,dx \\
&= e^{-r(T-t)}\left(\int_{-d2}^{\infty} S_t e^{(r-\sigma^2/2)(T-t)+x\sigma\sqrt{T-t}}\phi(x)\,dx - \int_{-d2}^{\infty} K\phi(x)\,dx\right) \\
&= e^{-r(T-t)}\left(\int_{-d1}^{\infty} e^{r(T-t)} S_t\phi(\widetilde{x})\,d\widetilde{x} - \int_{-d2}^{\infty} K\phi(x)\,dx\right) \\
&= S_t N(d_1) - e^{-rt} K N(d_2)
\end{aligned}$$

where $d1 = d2 + \sigma\sqrt{T-t}$ and $\widetilde{x} = x - \sigma\sqrt{T-t}$ [4], and thus we obtain the price of European call option that is the same as the one derived from solving the partial differential equation in the BSM model. In the case of American option, the option price f is no longer a simple function given that we need to consider early exercise. Thus, the analytical solution for the integral corresponding to the American option does not exist. We need to need to use tree-based algorithm and Monte Carlo simulation to approximate the solution.

3 Performance on European Call

Before going into discuss how the approximation algorithms are used on pricing options that do not have a closed form solution, let us first look into the performance of the tree-based algorithms and Monte Carlo simulation on valuing European call option, which we



(a) Two Steps Binomial Tree (b) Two steps Trinomial Tree (c) Adapted Tree from [1]

have a closed form solution for comparisons.

Binomial Tree

Figure 1a is a two step binomial recombining tree. Here, the continuous distribution of stock price is cut into discrete period. Suppose the price of stock is S at time t, then at time $t + \Delta t$ (Δt is length of time between connecting nodes), the price will be Su where $u > 1$ with probability p, or the price will be Sd where $0 < d < 1$ with probability $1 - p$. In the recombining tree, we need $d = 1/u$. If the tree is non-recombining, the nodes will grow exponentially as we increase the steps of the tree. For computational efficiency, recombining trees is more preferable. In the Cox, Ross, and Rubinstein's paper [5], the following parameters are used,

$$u = e^{\sigma\sqrt{\Delta t}} \qquad d = e^{-\sigma\sqrt{\Delta t}} \qquad p = \frac{e^{r\Delta t} - d}{u - d}$$

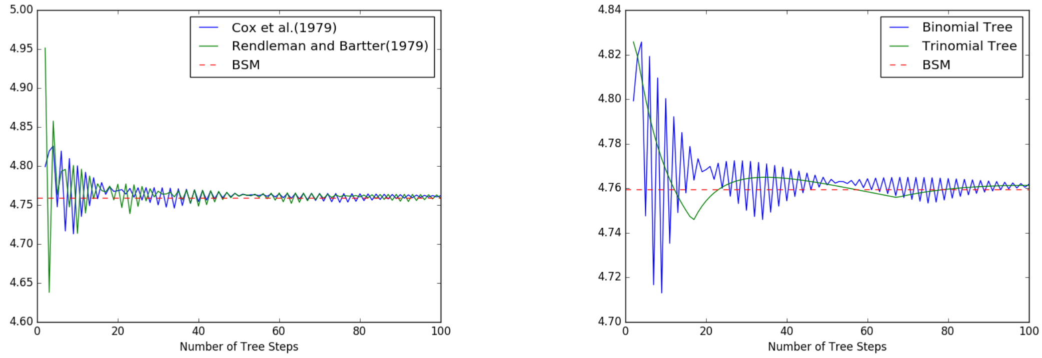
In the Rendleman and Bartter's paper [9], a different set of paramters are used,

$$u = e^{(r-0.5\sigma^2)\Delta t+\sigma\sqrt{\Delta t}} \qquad d = e^{(r-0.5\sigma^2)\Delta t-\sigma\sqrt{\Delta t}} \qquad p = 0.5$$

In either case, the parameters are chosen to match the log normal distribution of stock price in the assumption of the BSM model by Taylor Expansion with ignoring some higher order term for Δt. This approximation is robust when the number gets large, then the $\Delta t \to 0$, making the higher order term insignificant for approximation. To value an option using a binomial tree is straightforward. We will start at the end of the tree and work backward. For the nodes at the end of the tree, the valuation is

$$f_T = \max(S_T - K, 0) \text{ for call} \qquad f_T = \max(K - S_T, 0) \text{ for put}$$

To calculating f_t for a node at any time t, we use $f^u_{t+\Delta t}$ (the option price for the node where the stock price go up) and $f^d_{t+\Delta t}$ (the option price for the node where the stock



(a) Convergence comparisons of binomial tree with different parameters

(b) Convergence comparisons of binomial tree (Cox et al.) and trinomial tree

Figure 2: The convergence analysis is based on a European call option with $S_0 = 42,\ K = 40,\ r = 0.1,\ \sigma = 0.2, T = 0.5$

price go down), which

$$f_t = e^{-r\Delta t}\left(pf^u_{t+\Delta t} + (1-p)f^d_{t+\Delta t}\right)$$

By continuing the process recursively back through the tree, we can get the the pricing for the European option. Figure 2a shows the convergence of the valuations by the two binomial trees discussed above to the true valuation calculated by the BSM formula. One can see that the binomial tree with parameters introduced by Cox et al. converges much faster to the option valuation by BSM. Thus, in the rest of the paper, any binomial tree I refer to is the one with parameters introduced by Cox et al.

Trinomial Tree

Similar to the idea of binomial tree, in the trinomial tree, we are also allowing the stock to remain unchanged from one step to the next, as shown in Figure 1b. Now suppose that the price of stock is S at time t, then at time $t+\Delta t$, the price will be Su with probability p_u, or S with probability p_m, or Sd with probability p_d, and $p_m = 1 - p_u - p_d$. Again, in the case of recombining tree, we need $u = 1/d$. There are various set of parameters for the trinomial tree, and here I will use the one that is generally used, which is referred to as combining two steps of the binomial tree by Cox et al [6],

$$u = e^{\sigma\sqrt{2\Delta t}} \qquad d = e^{-\sigma\sqrt{2\Delta t}} \qquad p_u = \frac{e^{\sigma\Delta t/2} - d^{1/2}}{u^{1/2} - d^{1/2}} \qquad p_d = \frac{u^{1/2} - e^{\sigma\Delta t/2}}{u^{1/2} - d^{1/2}}$$

The steps of using trinomial tree in valuing the option is the same as those in binomial trees, and we again work backward through the tree. The calculation for f_t is

$$f_t = e^{-r\Delta t}\left(p_u f^u_{t+\Delta t} + p_m f^m_{t+\Delta t} + p_d f^d_{t+\Delta t}\right)$$

Figure 2b shows the convergence comparison between binomial and trinomial tree to the true valuation. One can see that the trinomial tree converge in a much faster and smoother rate than the binomial tree. With about 25 steps, the trinomial tree converges to region with ± 0.01 error to the valuation computed by the BSM. On the other hands, it takes the binomial tree more than 50 steps to converge to the ± 0.01 error region. From a theoretical perspective, the parameter I choose for trinomial tree is the parameter resulted in combining two steps of a binomial tree, and we should expect the tirnomial tree to converge should twice as fast as binomial tree.

Monte Carlo Simulation

The key of Monte Carlo simulation is based on the weak law of large numbers. Consider the following integral,

$$\int g(y)f(y)\,dy$$

where $g(y)$ is any arbitrary function and $f(y)$ is a probability density function. One can approximate the above integral by drawing n random samples of y_i from its probability density distribution, then by weak law of large number,

$$\frac{1}{n}\sum_{i=1}^{n} g(y_i) \to \int g(y)f(y)\,dy$$

In Section 2, I have shown that the price for European call option can be calculated by solving the integral below

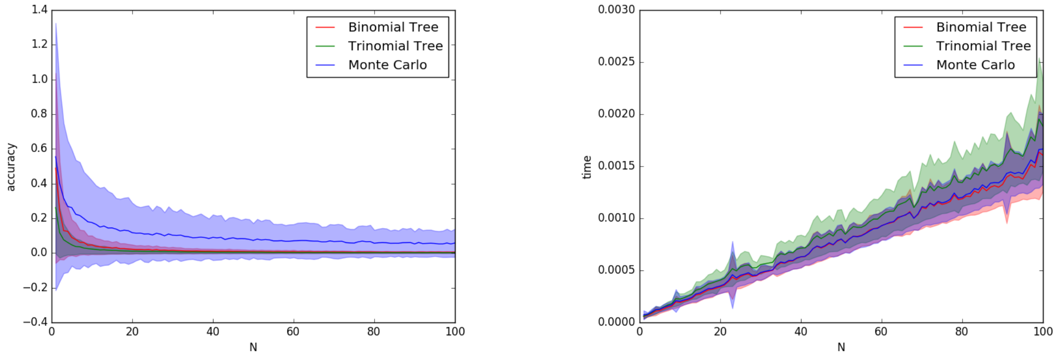
$$\mathbb{E}[f] = \int_{-\infty}^{\infty} f\phi(x)dx$$

where $x \sim \mathcal{N}(0,1)$, and $f = e^{-r(T-t)}\max(S_t e^{(r-\sigma^2/2)(T-t)+x\sigma\sqrt{T-t}} - K, 0)$. Thus, one can approximate the above integral by drawing n random samples of x_i from a standard normal distribution, then evaluate f for each x_i, and we obtain the following approximation

$$\frac{1}{n}\sum_{i=1}^{n} f(x_i) \to \mathbb{E}[f]$$

Accuracy and Computational Efficiency

Figure 3a shows the average accuracy of the three approximation algorithm over 1000 random European call option for different value of N (the number of tree steps for tree-based algorithm, the number of random draws in terms of $200N$ for Monte Carlo simulation). The shaded region is within one standard deviation away from the mean (the solid line). The accuracy is computed by $|f - f^*|$ where f is the option value computed by the approximation algorithm and f^* is the value computed by the BSM formula. We can see that as we increase N, we are getting a better accuracy, but this is in the cost of more



(a) Accuracy (b) Computational Time (seconds)

Figure 3: The comparison in accuracy and computational efficiency for binomial tree, trinomial tree, Monte Carlo on 1000 random sample European call options for non-dividend stock generated by the following set of parameters: $S_0 = 20, 30, 40, 50, 60, 70$; $K = 40, 45, 50$; $r = 0.02, 0.04, 0.06, 0.08, 0.1$; $\sigma = 0.1, 0.2, 0.3, 0.4$; $T = 0.5, 1, 1.5, 2$. For tree based algorithm, N stands for number of tree steps. For Monte Carlo, N means that the number of random samples is $200N$.

computational time as one can see in Figure 3b: as N increase, the computational time increase linearly. Accuracy comes with greater computational cost. Moreover, trinomial tree is the one with the highest accuracy but the computational cost is not so expensive as compared to others. Recall that the trinomial tree converge twice as fast as the binomial tree, but its computational cost is much less than twice the computation cost of binomial tree. Clearly, Monte Carlo simulation will not be optimal for approximating European call option on non-dividend stock comparatively to the tree-based algorithm, though its implementation is simple and striaght-forward.

4 Valuing European Call with Discrete Dividends

European call option on stock with discrete dividends does not have a closed form solution from the partial derivative in the Black-Scholes models. Different techniques have been developed to approximate the valuation. Here, I will look into the adaptive tree algorithm proposed by Areal and Rodrigues [1] and the Monte Carlo algorithm presented by Boyle [4].

Adapted Tree Algorithm

Figure 1c is the binomial tree for an option on stock with discrete dividend payments at time t_d that is presented in Areal and Rodrigues' paper [1]. We can see that the tree is no longer recombining after the dividend payment at time t_d, and the number of nodes grows exponentially after the dividend payment (from 4 to 16 in this case). The adaption is done by letting each dividend payment date as the nodes in the tree. In the case of single dividend stock in Figure 1c and suppose we want to divide the time between each dividend payment date by k time steps, we will let

$$\Delta t_1 = \frac{t_d}{k} \qquad \Delta t_2 = \frac{T - t_d}{k}$$

and the other parameters can be adjusted according to different length of time between connecting nodes,

$$u_j = e^{\sigma\sqrt{\Delta t_j}} \qquad d_j = e^{-\sigma\sqrt{\Delta t_j}} \qquad p_j = \frac{e^{r\Delta t_j} - d_j}{u_j - d_j}$$

Suppose the dividend is D, by making the above adaption, for each node at the first dividend date, it grows another binomial tree with stock price at the starting node be $S_{t_d} - D$ for each node. In the example of Figure 1c, at t_d, there are four nodes, so four binomial trees grow from the four nodes with the respective stock price less dividend D. Thus, in valuing the option, we can first compute the option price at the four nodes on time t_d by valuing the binomial trees that grow out from the four nodes. Then, we can value the binomial tree with that four nodes (at time t_d) as end points to get the valuation for the whole tree. This adaption makes the valuation and its implementation straight-forward. Moreover, the adaption can also be used in valuing American option on stock with discrete dividends. As Merton discussed in his paper, an American put option is never optimal to exercise immediately before the payment of any dividend [8]. Thus, when deciding whether to exercise at the dividend payment node, one should use ex-dividend value of the underlying asset (the valuation of American option on tree-based algorithm will be discussed in more detail in Section 5). On the other hand, an American call option is never optimal to exercise immediately after the payment of the dividend, so when deciding whether to exercise the option at the dividend payment node, one should use the price of asset before going ex-dividend. One limitation of this model is that the number of nodes grows exponentially as the number of discrete dividends increase. Suppose one use the model to value to option with n discrete dividends payment and we divide the interval between each dividend payment date into a k steps, then total number of nodes of the adapted tree will be $(k+1)^{n+1}$ [5], and this will result in great computational cost when n and k are both large as shown below.

Monte Carlo Simulation

The Monte Carlo estimation proposed by Boyle on valuing European call option is straightforward and easy to implement. Boyle utilized the assumption under BSM model that the stock price follows log normal distribution. Suppose we know the stock price at time t_0 is S_{t_0}, then stock price at time $t_1 > t_0$ can be expressed as

$$S_{t_1} = S_t e^{(r-\sigma^2/2)(t_1-t_0)+x\sigma\sqrt{t_1-t_0}}$$

where $x \sim \mathcal{N}(0,1)$, the standard normal distribution. Supposed the dividend payment date are $t_1, t_2, \ldots, t_n$, then the simulation process will go as the following. Moreover, let t_0 be today (the day that we value the option) and we still use T for expiration date.

- Simulate S_{t_1} from the above equation by sampling x randomly from the standard normal distribution.
- If $S_{t_1} > D_{t_1}$ (the dividend payment at t_1), then let $S_{t_1} = S_{t_1} - D_{t_1}$; otherwise, stop the simulation.
- Simulate $S_{t_2}, \ldots, S_{t_n}$ by the two steps above.
- After we get S_T, compute the option value of $\max(S_T - K, 0)$.
- Repeat the above simulation n times and compute the average of all the option values computed from each simulation to be the valuation of the European option.

Accuracy and Computational Efficiency

Table 1 presents the comparisons between the tree-based algorithm and Monte Carlo simulation on pricing European call on stock with discrete dividend that is used in Boyle's paper [4]. I chose that option since accurate option value computed by numerical integration is provided in the paper so I can make comparisons. For the option with spot price that is at least as large as the strike price, both tree-based algorithm and Monte Carlo simulation approximation is within less than 1% error from the true valuation. In the experiment I ran, tree algorithm is doing a better job than Monte Carlo simulation in general. However, the tree here is no longer recombining, which the number of nodes grows exponentially, so is the computational time, as we can see that the computational time for the tree with N and k both being large is enormous comparatively to other algorithms. For example, the trinomial tree algorithm with $k = 6$ on valuing the option with $N = 6$ and $S_0 = 50$ required 33.483 seconds while it only took it 0.231 seconds to finish valuing the option with same S_0 but with $N = 4$. If one needs to value an option on stock with 9 discrete payment, which $N = 10$ in this case, the computational time will be extremely expensive when k is slightly big (for example 6 for the Trinomial tree). On the other hand, in the case of Monte Carlo, as I increase N, the computational time only grows linearly, so it is much more efficient for using Monte Carlo simulation to value options on stocks with many discrete dividend payments in terms of computational power.

Table 1: Comparison between the adapted tree algorithm and Monte Carlo simulation on pricing European call on stock with discrete dividends.

S_0	N	Algorithm	Valuation	Accurate Values	Percent Difference	Time(s)
25	4	Binomial (4)	0.067	0.075	−10.67	0.009
		Binomial (10)	0.068		−9.33	0.187
		Trinomial (2)	0.067		−10.67	0.006
		Trinomial (6)	0.071		−5.33	0.189
		Monte Carlo	0.0794		5.9	0.270
25	6	Binomial (4)	0.258	0.266	−3.01	0.229
		Binomial (10)	0.260		−2.26	21.674
		Trinomial (2)	0.258		−3.01	0.121
		Trinomial (6)	0.256		−3.76	34.948
		Monte Carlo	0.282		6.02	0.409
50	4	Binomial (4)	7.219	7.251	−0.44	0.011
		Binomial (10)	7.251		0	0.183
		Trinomial (2)	7.219		−0.44	0.005
		Trinomial (6)	7.253		0.03	0.231
		Monte Carlo	7.295		0.61	0.278
50	6	Binomial (4)	9.002	9.001	0.01	0.236
		Binomial (10)	9.014		0.14	22.406
		Trinomial (2)	9.002		0.01	0.139
		Trinomial (6)	9.014		0.14	33.483
		Monte Carlo	8.921		−0.89	0.407
75	4	Binomial (4)	27.843	27.819	0.09	0.008
		Binomial (10)	27.806		−0.05	0.189
		Trinomial (2)	27.843		0.09	0.005
		Trinomial (6)	27.822		0.01	0.214
		Monte Carlo	27.841		0.08	0.270
75	6	Binomial (4)	29.329	29.300	0.10	0.224
		Binomial (10)	29.297		−0.01	23.225
		Trinomial (2)	29.329		0.10	0.149
		Trinomial (6)	29.300		0	33.263
		Monte Carlo	29.741		1.51	0.403

Note. S_0 is the stock price today, N is the number of period to maturity where the length of each period is one quarter. The other parameters for the options are $K = 50,\ r = 0.06\,\sigma^2 = 0.1$. Each dividend payment D is 0.25 and it is paid at the end of each quarter except for maturity. The number in bracket after “binomial” or “trinomial” is the k in adapted tree algorithm. Accurate values is obtained by numerical integration provided on Boyle’s paper [4]. Time is the computational time associated with the valuation in unit of seconds. Monte Carlo is done by drawing 10,000 samples.

5 Valuing American Put with No Dividend

American call on non-dividend stock is never optimal to exercise early, and thus the valuation of the American call is the same to the valuation of the European call with the same set of parameters. However, for American put, it is possible to exercise early, but no closed form solution exist from solving the partial differential equation in the Black-Scholes model. Here, I will present two approximation algorithm: the tree-based algorithm and the Least Square Monte Carlo.

Tree-Based Algorithm

Both binomial tree and trinomial tree are efficient in handling the American option. We only consider whether to exercise early at each node we have in the tree, though in reality, we can exercise at any time we want. For example, at a given node A with option value f_A and the associated stock price of S_A, we will exercise the option at A if $f_A < K - S_A$ for American put, meaning that one should exercise if immediate payoff is greater than the payoff for keeping the option alive. In another word, the option value at node A will be $\max(f_A, K - S_A)$ rather than simply f_A, and we need to do the same thing for all interior nodes (nodes that are not at the end of the tree) when working backward through the tree.

Least Square Monte Carlo

As discussed in the tree-based algorithm, it is optimal to exercise the American option early if the immediate exercise value if greater than the expected cash flow from continuing. However, it is not obvious how to compute the conditional expected value of keeping the option alive by using Monte Carlo until the introduction of the Least Square Monte Carlo. As suggested by Longstaff and Schwartz, it is better to explain the algorithm by using a simple numerical example. Let us consider an American put option on non-dividend stock. Suppose $K = 50$ and it is exercisable at 3 month, 6 month, and 9 month where 9 month is also the maturity. Consider $r = 0.06,\ \sigma = 0.1,\ S_0 = 48$. I will first generate the stock price paths again based on the log normality of stock price, which is the same stock price sampling step in the Monte Carlo simulation algorithm presented in Section 4. Here, I will only use 8 paths for illustrations, and the stock price paths matrix is the following.

Path	t = 0	t = 0.25	t = 0.5	t = 0.75
1	48	48.849	50.763	50.965
2	48	45.931	46.582	47.608
3	48	51.611	53.498	57.585
4	48	47.635	44.005	44.784
5	48	49.907	55.113	52.928
6	48	45.085	46.791	45.434
7	48	48.109	49.236	47.848
8	48	48.857	52.557	52.590

Similar to the tree-based algorithm, we also work backward in this algorithm. Let us first focus on the payoff at $t = 0.75$ (t is expressed in unit of year). Recall that it is only optimal to exercise the put if the strike price is above the price of the underlying asset. The payoff matrix at $t = 0.75$ is the following.

Path	t = 0.25	t = 0.5	t = 0.75
1	–	–	0.000
2	–	–	2.392
3	–	–	0.000
4	–	–	5.216
5	–	–	0.000
6	–	–	4.566
7	–	–	2.152
8	–	–	0.000

Now we need to look at the $t = 0.5$. If the put is in the money at $t = 0.5$, it is possibly optimal for the option holder to exercise early, so we need a way to compute the conditional payoff of continuing the option to $t = 0.75$. From the stock price paths matrix, we can see that path 2, 4, 6, 7 are in the money at $t = 0.5$. Let X denotes the stock price at $t = 0.5$ for those 4 paths. Now let Y denote the discounted payoff received at $t = 0.75$ if the put option is not exercised at time $t = 0.5$. The vector of X and Y is summarized in the following table.

Path	Y	X
1	–	–
2	$2.392 \times e^{-0.06\times 0.25}$	46.582
3	–	–
4	$5.216 \times e^{-0.06\times 0.25}$	44.005
5	–	–
6	$4.566 \times e^{-0.06\times 0.25}$	46.791
7	$2.152 \times e^{-0.06\times 0.25}$	49.236
8	–	–

Here is where the Least Square involved in the process. We will regress Y on a constant, X and X^2, and the resulting regression is the conditional expectation function for the payoff of continuing the option, which is $\mathbb{E}[Y|X] = 83.48 - 2.87X + 0.0248X^2$ in this case. Now we can compute the expected value of the option if keeping alive and compare it to the payoff of immediate exercise at $t = 0.5$ (which is $50 - X$). The table below shows the comparison.

Path	Exercise	Continuation
1	–	–
2	3.418	3.483
3	–	–
4	5.995	5.095
5	–	–
6	3.209	3.367
7	0.764	2.167
8	–	–

The comparison implies that path 4 is worth immediate exercise, and the payoff matrix at time $t = 0.5$ conditional on not exercising the option prior to $t = 0.5$ is the following.

Path	t = 0.25	t = 0.5	t = 0.75
1	–	0.000	0.000
2	–	0.000	2.392
3	–	0.000	0.000
4	–	5.995	0.000
5	–	0.000	0.000
6	–	0.000	4.566
7	–	0.000	2.152
8	–	0.000	0.000

Since the option can only be exercised once, if one decides to exercise early, then no payoff will be received after exercising, as the payoff of path 4 at $t = 0.75$ becomes 0 if it is exercised early at $t = 0.5$. Now we proceed recursively and check whether it is optimal to exercise at $t = 0.25$. Again, from the stock price paths matrix, all paths except 3 are in the money, and I will let X denotes stock price at $t = 0.25$ for the 7 paths and find the corresponding Y, which

Path	Y	X
1	$0.000 \times e^{-0.06\times0.25} + 0.000 \times e^{-0.06\times0.5}$	48.849
2	$0.000 \times e^{-0.06\times0.25} + 2.392 \times e^{-0.06\times0.5}$	45.931
3	–	–
4	$5.995 \times e^{-0.06\times0.25} + 0.000 \times e^{-0.06\times0.5}$	47.635
5	$0.000 \times e^{-0.06\times0.25} + 0.000 \times e^{-0.06\times0.5}$	49.907
6	$0.000 \times e^{-0.06\times0.25} + 4.566 \times e^{-0.06\times0.5}$	45.085
7	$0.000 \times e^{-0.06\times0.25} + 2.152 \times e^{-0.06\times0.5}$	48.109
8	$0.000 \times e^{-0.06\times0.25} + 0.000 \times e^{-0.06\times0.5}$	48.857

Regressing Y on X, and we get the conditional expectation for the payoff of continuing the option, which is $\mathbb{E}[Y|X] = -581.40 + 25.59X - 0.280X^2$. Now we can construct the matrix for comparing the two payoff (immediate exercise vs. continue) at $t = 0.25$ to decide whether to exercise early.

Path	Exercise	Continuation
1	1.151	1.265
2	4.069	3.939
3	–	–
4	2.365	2.956
5	0.093	-0.881
6	4.915	3.823
7	1.891	2.393
8	1.143	1.252

The above comparison implies that it is optimal to exercise at $t = 0.25$ for path 2, 5, 6. and this yields the following payoff matrix for the full life of the put option.

Path	t = 0.25	t = 0.5	t = 0.75
1	0.000	0.000	0.000
2	4.069	0.000	0.000
3	0.000	0.000	0.000
4	0.000	5.995	0.000
5	0.093	0.000	0.000
6	4.915	0.000	0.000
7	0.000	0.000	2.152
8	0.000	0.000	0.000

The matrix above shows that it is optimal to exercise at $t = 0.25$ for path 2, 5, and 6, optimal to exercise at $t = 0.5$ for path 4, and optimal to exercise at $t = 0.75$ for path 7. For the other path, it is not optimal to exercise at all. Thus, we can discount each value back to $t = 0$ and take the average over the 8 paths, which is 2.102, and this is the approximating option value for the American by Least Square Monte Carlo.

Table 2: Comparison between the Tree-based algorithm and Least Square Monte Carlo (LSM) on pricing American put option on non-dividend stock.

S_0	N	Binomial	Trinomial	LSM
25	52	25.000 (2.821)	25.000 (2.821)	24.970 (2.791)
	100	25.000 (2.821)	25.000 (2.821)	24.979 (2.800)
	500	25.000 (2.821)	25.000 (2.821)	24.993 (2.814)
50	52	5.057 (0.306)	5.058 (0.307)	5.114 (0.363)
	100	5.064 (0.313)	5.064 (0.313)	5.012 (0.261)
	500	5.069 (0.318)	5.069 (0.318)	4.943 (0.192)
75	52	0.609 (0.027)	0.605 (0.023)	0.612 (0.030)
	100	0.605 (0.023)	0.600 (0.018)	0.598 (0.016)
	500	0.603 (0.021)	0.602 (0.020)	0.603 (0.021)

Note. S_0 is the stock price today, N is the number of possible exercisable date for the option, including maturity, then $N-1$ chance of exercising the option early before maturity. The other parameters for the options are $K = 50,\ r = 0.06\,\sigma^2 = 0.1,\ T = 1$. The number in the bracket is the dollar value by having right of early exercise, which is the American put value less the corresponding European put value calculated from the BSM formula.

Accuracy and Computational Efficiency

Table 2 presents the comparisons between the tree based-algorithm and Least Square Monte Carlo on pricing an American put option on non-dividend stock. Since American put does not have a closed form solution, there is no true value that I can compare to. From Table 2, one can see that the approximation of all algorithms converge to similar value. Moreover, for American put option with stock price much lower than the strike price, the premium of having the right to exercise early (American vs European) is also more expensive. Since all three algorithms perform similarly on valuation, we might care about the computational efficiency when deciding which method to use. Figure 4 shows that the computational cost for Least Square Monte Carlo is weight greater than the binomial tree and trinomial tree in valuing the American put options. Thus, it is optimal in valuing the American option on non-dividend stock by using the tree-based algorithm.

Conclusion

In this paper I present how tree-based algorithm and Monte Carlo simulation can be used to approximate the value of different option, either with closed form solution (European call option on non-dividend stock) or not (European call option on stock with discrete dividend and American put option on non-dividend stock) . In comparison, the tree-based algorithm often yields a more accurate approximation than the Monte Carlo simulation method even with lower computational cost in most of the example I provided

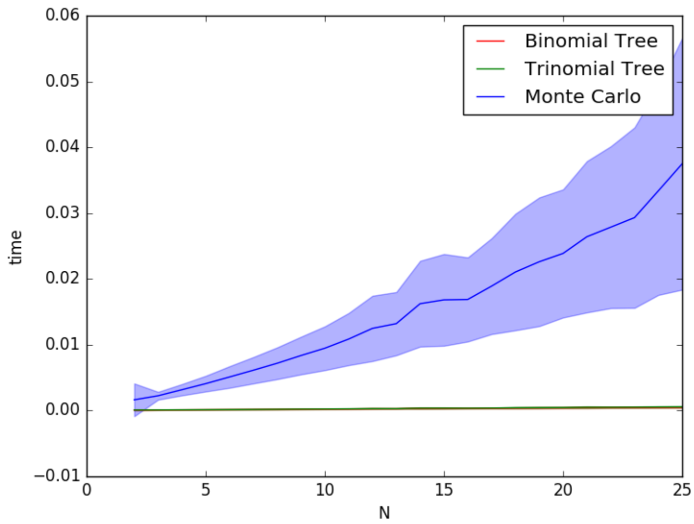


Figure 4: The comparison in computational efficiency (presented by computational time in unit of seconds) for binomial tree, trinomial tree, and Least Square Monte Carlo on different value of N (the number of possible exercisable date for the option, including maturity) on 1000 random sample American put options on non-dividend stock generated by the following set of paramters: $S_0 = 20, 30, 40, 50, 60, 70$; $K = 40, 45, 50$; $r = 0.02, 0.04, 0.06, 0.08, 0.1$; $\sigma = 0.1, 0.2, 0.3, 0.4$; $T = 0.5, 1, 1.5, 2$.

in this paper. However, when the underlying asset follows a mix of different distribution, the tree-based algorithm might be computational expensive as I have discussed in valuing European call on stock with discrete dividend payment. Moreover, Monte Carlo simulation also allows for more flexibility in the approximation process since one can also take other factors that might potentially affect the price of the underlying asset into the simulation process.

Notes

[1]only two possible next states, so it is called binomial

[2]American call with non-dividend is not optimal to exercise early, so the valuation is the same as the European call, which has a closed solution under BSM formulation.

[3]This is not full set of the assumptions, but some key ones that are used in the process of showing the BSM formula is risk neutral valuation. Other assumptions can be refered to Black and Scholes' paper [2]

[4]The skipping steps are provided in the Appendix

[5]Each binomial tree between connecting dividend payment nodes has $k+1$ nodes. At each node at the dividend payment date, it grows out another binomial tree with $k+1$ nodes. Thus, with n dividend payment dates, the number of nodes for the whole tree will be $(k+1)^{n+1}$.

References

[1] Nelson Areal and Artur Rodrigues. "Fast trees for options with discrete dividends". In: *The Journal of Derivatives* 21.1 (2013), pp. 49–63.

[2] Fischer Black and Myron Scholes. "The pricing of options and corporate liabilities". In: *Journal of political economy* 81.3 (1973), pp. 637–654.

[3] Phelim P Boyle. "A lattice framework for option pricing with two state variables". In: *Journal of Financial and Quantitative Analysis* 23.01 (1988), pp. 1–12.

[4] Phelim P Boyle. "Options: A monte carlo approach". In: *Journal of financial economics* 4.3 (1977), pp. 323–338.

[5] John C Cox, Stephen A Ross, and Mark Rubinstein. "Option pricing: A simplified approach". In: *Journal of financial Economics* 7.3 (1979), pp. 229–263.

[6] Emanuel Derman, Iraj Kani, and Neil Chriss. "Implied trinomial tress of the volatility smile". In: *The Journal of Derivatives* 3.4 (1996), pp. 7–22.

[7] Francis A Longstaff and Eduardo S Schwartz. "Valuing American options by simulation: a simple least-squares approach". In: *Review of Financial studies* 14.1 (2001), pp. 113–147.

[8] Robert C Merton. "Theory of rational option pricing". In: *The Bell Journal of economics and management science* (1973), pp. 141–183.

[9] Richard J Rendleman and Brit J Bartter. "Two-State Option Pricing". In: *The Journal of Finance* 34.5 (1979), pp. 1093–1110.

Appendix A The "missing" step in solving the Integral

The missing step is the u-substitution from x to $\widetilde{x} = x - \sigma\sqrt{T-t}$ for the integral, the steps are the following.

$$\begin{aligned}
&\int_{-d2}^{-\infty} S_t e^{(r-\sigma^2/2)(T-t)+x\sigma\sqrt{T-t}}\phi(x)\,dx \\
=&\int_{-d1}^{\infty} S_t e^{(r-\sigma^2/2)(T-t)+(\widetilde{x}+\sigma\sqrt{T-t})\sigma\sqrt{T-t}}\phi(\widetilde{x}+\sigma\sqrt{T-t})\,d\widetilde{x} \\
=&\int_{-d1}^{\infty} S_t e^{(r-\sigma^2/2)(T-t)+(\widetilde{x}+\sigma\sqrt{T-t})\sigma\sqrt{T-t}}\frac{1}{\sqrt{2\pi}}e^{-0.5(\widetilde{x}+\sigma\sqrt{T-t})^2}\,d\widetilde{x} \\
=&\int_{-d1}^{\infty} S_t e^{(r-\sigma^2/2)(T-t)+(\widetilde{x}+\sigma\sqrt{T-t})\sigma\sqrt{T-t}-\widetilde{x}\sigma\sqrt{T-t}-0.5\sigma^2(T-t)}\frac{1}{\sqrt{2\pi}}e^{-0.5\widetilde{x}^2}\,d\widetilde{x} \\
=&\int_{-d1}^{\infty} S_t e^{r(T-t)}\phi(\widetilde{x})\,d\widetilde{x}
\end{aligned}$$

where $d1 = d2 + \sigma\sqrt{T-t}$.