American Option Valuation Methods

Jinsha Zhao[1]

[1] Kingston Business School, Kingston University, London, UK

Correspondence: Jinsha Zhao, Kingston Business School, Kingston University, London, KT2 7LB UK. Tel: 044-208-417-5138. E-mail: j.zhao@kingstson.ac.uk

Abstract

This paper implements and compares eight American option valuation methods: binomial, trinomial, explicit finite difference, implicit finite difference and quadratic approximation methods. And three Monte Carlo methods: bundling technique of Tilley (1993), simulated tree (ST) of Broadie, Glasserman, and Jain (1997), and least square regression method (LSM) of Longstaff and Schwartz (2001). Methods are compared in terms of computation efficiency and price accuracy. The findings suggest that binomial is the best performing numerical method in terms of accuracy and efficiency. LSM beats the other two simulation methods in terms of efficiency, accuracy and number of discrete exercise opportunities.

Keywords: American options, numerical methods, binomial tree, simulation method, least square regression method

1. Introduction

American option pricing problems have been extensively studied over past two decades. Many numerical and analytical methods have been developed and most standard problems in American option pricing literature have been solved. However, research in this area is far from end. Recent development in financial engineering has introduced a variety of new American option whose payoff contingent on multiple source of uncertainty. Pricing these options, which have sophisticated payoff structures, are computationally costly. Existing pricing methods have not provided a satisfactory answer. Some numerical methods are readily capable of pricing these options, but their efficiency and accuracy may need further improvements.

This paper evaluates five popular numerical methods and three simulation methods which are widely used for pricing American option. Each method is compared based on its computational efficiency and price accuracy. Results in the paper show that each method has its advantage and disadvantage, depending on the actual application. As higher efficiency usually comes with a cost – lower accuracy, a method could be very efficient but not very accurate. For example, to value a large number of short-term options, quadratic method is the best fit as the method is very efficient and accuracy for short-term option is also good. Overall, the most flexible method is binomial tree, where users can pre-specific the number of tree steps based on available hardware and desirable accuracy. So any level of accuracy can be achieved. Simulation methods are generally not very efficient, as early exercise has to be calculated forward in time but option values can only be evaluated backwards in time. As the result shows, the most efficient simulation method, LSM, is about 40 times slower than binomial method with comparable accuracy. However, this is expected as simulation methods are most suited to value American options with multiple uncertainties.

The rest of the paper is outlined as follows: section 2 review American option pricing literature. Section 3 provides descriptions of each method. Section 4 compares efficiency and accuracy of each method. Section 5 concludes and summaries findings.

2. Literature Review

Development of American option pricing is more dramatic than its European counterpart. Unlike European options where closed form pricing formula are available, pricing of American options are complicated by its exercise contingency. Closed form solutions are usually rare. The only case where a closed-form solution to pricing an American option exists is an American option with no dividend. Numerical methods hence needed to be employed for all other American options. Early attempts made to price American options are the binomial lattice model of Cox, Ross, and Rubinstein (1979). And the finite difference methods of Schwartz (1977) and

Brennan and Schwartz (1977). The methods are simple and easy to implement, they still remained the best ways for pricing American styled options. Binomial model has been extended by Boyle (1986) in which a middle price jump was incorporated in the price tree. The result trinomial model converges to true option values quicker than that of binomial model. As later discussed in Broadie and Detemple (1996) that trinomial model dominate binomial model in terms of both speed and accuracy. Horasanlı (2007) also reached the same conclusion that trinomial model have a faster convergence rate.

The two alternative finite difference methods have been compared in Geske and Shastri (1985) and they concluded that explicit finite difference methods with log-transformation was the most efficient approach when large numbers of stock options are being evaluated. To compensate explicit finite difference's instable problem (not always converges), Hull and White (1990) proposed modified version of explicit finite difference which always converge. They also discussed the possibility that efficiency of explicit finite difference could be largely improved by imposing a fixed mesh ratio between time and price jumps. Its model had also applied to higher dimensional interest rate options.

Geske and Johnson (1984) presented an exact analytic solution to the American put problem. However their formula is an infinite series that can only be evaluated approximately by numerical methods. Based on this exact formula, MacMillian (1986) proposed a quadratic approximation of American put option. Few months later, Barone-Adesi and Whaley (1987) generalised this method to American call options and proposed quadratic approximation. It is still the fastest way for American option pricing. However, quadratic method suffered a shortcoming that it is not convergent. There is no parameter in the model that can be increased to give arbitrarily high accuracy. And quadratic approximation results for long maturity American option are less satisfactory, this will be demonstrated in the next section.

Also based on Geske and Johnson (1984), Carr and Faguet (1994) proposed improvements to methods of lines, which they call analytic method of lines. As shown in Broadie and Detemple (1996), analytic method of lines of Carr and Faguet (1994) has a quicker convergent rate than binomial method. Kim (1990) developed another analytical solution for American option. The American option value is represented by an integral equation where the exercise boundary is implicitly defined. So a computationally intensive recursive numerical procedure can be performed to solve for the exercise boundary and option price.

Binomial method is very simple and has huge success in single state context. However, its success in higher dimensions where American option price depends on more than one underlying assets is limited. First theoretical extension of such model have been carried out by Boyle (1988), he proposed a procedure for valuing options when there are two correlated state variables. In a three dimensional space, the original binomial tree expanded to a pyramid where its top is the initial price. A year later, Boyle, Evnine, and Gibbs (1989) generalised Boyle (1988) work to account for more than two correlated underlying state variables. However, their model has not been applied to American options. Kamard and Ritchken (1991) subsequently showed that the convergence rate of Boyle, Evnine and Gibbs (1989) can be improved by incorporating horizontal jumps in the price movements. Although their work can easily apply to American pricing problem, their focus is on European options. And most other works carried out in 80s also concentrated on the price of European option.

Although theoretical multinomial model have been developed, its practical use in higher dimensional problem proved to be very difficult. This difficulty has to do with data storage requirements. Increasing dimensions make the lattice methods and finite differences computationally prohibitive.

The only known solution to the dimensional problem is Monte Carlo method since its simulation convergence law only depends on the variance of the price population, not on the number of stochastic dimensions of the population. The first attempt made to apply simulation for American option is by Tilley (1993) where he proposed a bundling technique to replicate the backward induction algorithm. However, Tilley method had not provided a satisfactory solution to the dimensionality problem. As Broadie, Glasserman, and Jain (1997) had pointed out that Tilley's method was not easy to be generalized to higher dimensions. Broadie, Glasserman and Jain (1997) also offered a non-recombining binomial simulation approach. But their method suffered computational problems as simulated tree does not recombine. Number of nodes in the tree increase exponentially with time steps. It quickly becomes unmanageable as the number of exercise opportunities grows. This method can only effectively handle American options that have four or less discrete exercise opportunities. Broadie, Glasserman and Jain (1997) thus provided some enhancement to this method. They proposed a pruning technique to reduce computation burden and other variance reduction technique to increase precision. They also demonstrated the results with some higher dimensional problems. After this extension their method was a very promising technique for American option with finite exercise opportunities. Broadie and Glasserman (1998)

designed stochastic mesh method for high-dimensional problems. The primary advantage of this method is that computational effort increases quadratically with the number of mesh points and linearly with the number of exercise opportunities. However, this method as demonstrated in their paper is not generally useful for extremely accurate results. And they suggested that problem specific variance reduction technique would improve applicability of the method.

The first definite breakthrough in pricing early exercise derivatives by Monte Carlo was done by Longstaff and Schwartz (2001). Their least square Monte Carlo (LSM) method is computationally efficient and converges to the true value and can be readily extended to high dimensions. Its superior performance have attracted much attentions in the academic, many improvements have been proposed. Stentoft (2003) has explored theoretical foundation of LSM and property of its estimator. He also proved that the LSM approximations converge to the true expectation functions under general assumptions. Stentoft (2004) provided another detailed analysis of LSM and he shows LSM is computationally more efficient than existing numerical methods. He also demonstrated that the LSM method can be implemented easily for dimensions as high as ten or more.

3. American Option Valuation Techniques

In this section, five numerical techniques and three Monte Carlo techniques for American option pricing are discussed. The five numerical techniques are binomial method of Cox, Ross, and Rubinstein (1979), trinomial method of Boyle (1986), quadratic approximation of Barone-Adesi and Whaley (1987), implicit and explicit finite difference methods of Schwartz (1977). The three Monte Carlo methods are bundling method of Tilly (1993), simulated random tree of Broadie, Glasserman and Jain (1997), and least square method of Longstaff and Schwartz (2001). These methods are chosen because they have huge success in American option pricing literature and they are relatively simple to implement.

3.1 General Framework

This section defines the option pricing framework that would be used throughout the paper. The assumption used here are standard in the literature and there are consistent with those introduced by Black-Scholes (1973). The underlying price *S* follows geometric Brownian motion. The price process is:

$$\frac{dS}{S} = (r - q)dt + \sigma dZ$$

Z is a standard wiener process. Interest rate *r* and dividend *q* are continuous and constant throughout the maturity unless stated otherwise. Variance σ^2 of the underlying process is also constant for whole maturity.

3.2 Binomial Tree Method

Binomial method is one of the most successful numerical techniques. The method's major advantage is its simplicity. Anyone with excel installed can easily implement this method.

Introduction about this method are kept minimum since it has already been covered in many books and papers. The basic intuition behind binomial tree model is simple – by constructing a dense tree that capture all possible future prices, American option value can be evaluated via a backward induction from the end of the tree to the initial node. Both Hull (2004) and Global Derivatives (Note 1) provide detailed description of such methods.

Implementation is fairly straightforward. Firstly, computing the jump parameter *u* and its associated probability *p* then the downward jump parameter is 1/*u* with probability 1 – *p*. Secondly, building the tree based on the four parameters and the convergent parameter *n* – number of time steps. Finally, the terminal option value are defined by their payoff function, then option values in all *n* – 1 nodes are evaluated as maximum of its intrinsic value and its corresponding discounted expected payoff.

3.3 Trinomial Tree Method

This is a major modification of CRR's binomial tree method in previous section. This method is also simple and efficient. Its only difference with binomial method is that a horizontal jump was incorporated and the horizontal jump always has a probability of 2/3. Its main improvement over binomial method is that it converges quicker given the same number of time steps. This result should be intuitively expected since for every time step trinomial tree has more price nodes.

Details of the method is not discussed either since this is also a commonly known technique. Implementation is very similar to that of BT. The only difference is in the construction of tree to include the middle jump.

3.4 Quadratic Approximation

It is widely accepted that American option price is equal to sum of an equivalent European option and the early

exercise premium:

$$C(S,T) = c(S,T) + \varepsilon(S,T) \tag{1}$$

$\varepsilon(S,T)$ is the early exercise premium. BAW assumed that $\varepsilon(S,T) = K(T)f(S,T)$. So that the premium will satisfy the differential equation:

$$S^2 f_{ss} + \frac{2(r-q)}{\sigma^2} S f_s - \frac{2r}{\sigma^2 K} f - 2r \frac{2r}{\sigma^2}(1-K)f_K = 0 \tag{2}$$

The last term in above equation is assumed to be approximately zero for short maturity options. Then the new approximated second order differential equation can be solved for $f(S,T)$. The resulting function $f(S,T)$ are then bring back into equation (1) so that we have:

$$C(S,T) = c(S,T) + K(T)f(S,T) \tag{3}$$

Under the smooth pasting condition of Merton (1973) American option is optimally exercised when price S^* is reached such:

$$C(S^*,T) = S^* - X \tag{4}$$

With the known early exercise premium function the equation can be write as:

$$S^* - X = c(S^*,T) + K(T)f(S^*,T) \tag{5}$$

With this equation, the optimal exercise price can be solved iteratively.

The major advantage of quadratic approximation is speed, BAW have even proposed a very quick approximation method to solve early exercise price. Implementation of this method is fairly easy, the boundary price can be solved using Newton's method which involves differentiate right hand right side of equation (5) with respect to S^*.

3.5 Finite Difference Methods

Finite difference methods were first introduced by Schwartz (1977) and Brennan and Schwartz (1977) for American option pricing. They approximated the Black-Scholes partial differential equations by using discrete estimates of changes in the options values for small changes in time and the underlying stock price. The resulted approximation difference equation then can be evaluated to solve American option value. There are two major ways of estimating change's in option value with respect to time and the stock price: forward and backward, respectively they represent implicit and explicit finite difference methods.

Implicit finite difference has the advantage that its convergence is stable, but has the disadvantage of inversing matrices. Explicit finite difference is more computational friendly as it is only a replication of multiple trinomial trees. Its only disadvantage is that convergence is not always ensured. Hull and White (1990) mentioned that a fix ratio between time increment and price increment could ensure convergence. Hull (2006) then suggested it is numerically most efficient to set $\Delta Z = \sigma\sqrt{3\Delta t}$, where ΔZ is the log-transformed price increment.

Both methods are specified as follow: *firstly*, the higher boundary for underlying price is determined to be 3 times of exercise price (e.g. $\log(3X)$, where X is the exercise price). Although 2 times of exercise price is fairly large to cover all possible initial prices, there are some extreme cases where underlying price climbed to a very high level (as demonstrated in later sections when early exercise boundary is examined under extreme parameters). *Secondly*, the lower boundary for underlying is 1 (e.g. $\log 0$), because both methods are implemented under log-transformed specification. *Thirdly*, for explicit finite difference methods the ratio between ΔZ^2 and Δt are fixed at $\Delta Z^2 = 3\Delta t\sigma^2$ to ensure convergence. In computation Δt will be given as the convergence parameter, ΔZ is then computed according to the expression. After implementation variety of input parameters was tested, as expected the explicit finite difference always converge (however, accuracy is not tested in these cases).

Option prices that line between two nodes will be approximated using linear extrapolation by assuming option value function between the two nodes is a straight line. Such approximation will not significantly compromise accuracy if the grid is very condensed.

3.6 Bundling Method of Tilley

It is the first attempt made to apply Monte Carlo method for American option pricing problem. Although later

literature have pointed out that bundling estimator are biased, next section demonstrates that this algorithm still approximate the option price with fairly good accuracy. The intuition behind bundling is that it groups paths whose stock price are similar to obtain an estimate of the one period ahead option value, then this procedure are carried out recursively until time 0.

Implementation of algorithm is as follow:

1) Reorder the stock price paths by stock price, from lowest price to highest price for a call option or from highest price to lowest price for a put option. Reindex the paths from 1 to R according to the reordering.

2) For each path k, compute the intrinsic value $I(k, t)$ of the option.

3) Partition the set of R ordered paths into Q distinct bundles of P paths each.

4) For each path k, the option's "holding value" $H(k, t)$ is computed as the following mathematical expectation taken over all paths in the bundle containing the path k:

$$H(k,t) = Exp(-r\Delta t)P^{-1} \sum_{\forall j \in P(k,t)} V(j, t+1)$$

$V(k, t)$ is fully defined in step 8 below. At time N (the terminal step), $V(k, N) = I(k, N)$ for all k.

5) For each path, compare the holding value $H(k, t)$ to the intrinsic value $I(k, t)$ and decide "tentatively" whether to exercise or hold. Define an indicator variable $x(b, t)$ as follows:

$$x(b,t) = \begin{cases} 1 & \text{if } I(k,t) < H(k,t) \\ 0 & \text{if } H(k,t) \geq I(k,t) \end{cases}$$

6) Examine the sequence of 0's and 1's. Determine a "sharp" boundary between the hold decision and the exercise decision as the start of the first string of 1's the length of which exceeds the length of every subsequent string of 0's. Let $k_*(t)$ denote the path index (in the sample as ordered in step 1 above) of the leading 1 in such a string. The "transition zone" between hold and exercise is defined as the sequence of 0's and 1's that begins with the first 1 and ends with the last 0.

7) Define a new exercise or hold indicator variable $y(k, t)$ that incorporates the sharp boundary as follows:

$$y(b,t) = \begin{cases} 1 & \text{if } k \geq k_*(t) \\ 0 & \text{if } k < k_*(t) \end{cases}$$

8) For each path k, define the current value $V(k, t)$ of the option as follows:

$$V(k,t) = \begin{cases} I(k,t) & \text{if } y(k,t) = 1 \\ H(k,t) & \text{if } y(k,t) = 0 \end{cases}$$

After the algorithm has been processed backward from time N to time 0, the indicator variable $z(k, t)$ for $t < N$ is estimated as follows:

$$z(k,t) = \begin{cases} 1 & \text{if } y(k,t) = 1 \text{ and } y(k,s) = 0 \text{ for all } s < t \\ 0 & \text{otherwise} \end{cases}$$

The American option value is then defined as:

$$Value = R^{-1} \sum_{k=1}^{R} \sum_{t=1}^{N} z(k,t)D(k,t)I(k,t)$$

Where $D(k,t)$ is path specific discount factor. In the paper it is assumed that $D(k,t)$ is equal to $Exp(-r\Delta t)$ for all path k and for all t.

3.7 Simulated Random Tree

In contrast to the Tilley's approach, in the simulated random tree algorithm the evolution of stock prices is simulated using random trees rather than just sample paths. In the tree each node generates b number of branches, where b is called the branching parameter. As price jump is completely random and in any directions, the tree will not recombine. Each node in the tree is generated from its prior node using the equation:

$$S_i = S_{i-1} Exp\left[(r - q - \sigma^2/2)(t_i - t_{i-1}) + \sigma Z \sqrt{t_i - t_{i-1}}\right]$$

Z is a standard normal random variable. For simplicity, the time increment t_i-t_{i-1} is assumed to be equal to Δt over the whole tree for every price jump.

The simulated tree then can be used to derive two estimators, one biased high and one biased low. The high estimator at time t as defined in Broadie, Glasserman and Jain (1997) is the maximum of its intrinsic value and the average of the discounted high estimators from its successor nodes at $t+1$. The high estimator at T (maturity) is defined as option's intrinsic value. The low estimator at t for any node A is defined as the average of "decision value" from its successor nodes at $t+1$. The "decision value" in each $t+1$ successor node is defined as its discounted option intrinsic value if the average of the rest b-1 intrinsic values is greater than intrinsic value of node A; or the decision value is intrinsic value of A if the average is smaller. Confidence interval of true option value and a point estimate can then be computed based on the two estimators. The low estimator at T (maturity) is defined as its intrinsic value.

Formal definition of both estimators is outlined below,

$$\Psi_T^{i_1 i_2 \dots i_T} = L_T(S_T^{i_1 i_2 \dots i_T})$$

$$\Psi_t^{i_1 i_2 \dots i_t} = \max\left\{ L_t(S_t^{i_1 i_2 \dots i_t}), \frac{1}{b}\sum_{j=1}^{b} d_t \Psi_{t+1}^{i_1 i_2 \dots i_{tj}} \right\}, t = 0, \dots T-1$$

$$\psi_T^{i_1 i_2 \dots i_T} = L_T(S_T^{i_1 i_2 \dots i_T})$$

$$\psi_t^{i_1 i_2 \dots i_t} = \frac{1}{b}\sum_{j=1}^{b} \eta_t^{i_1 i_2 \dots i_{tj}}, t = 0, \dots T-1$$

where Ψ and ψ denote the high estimator and the low estimator respectively. L_t is the option value if immediately exercised. b is the branching parameter, each node at time t will generate b new nodes at $t+1$. d_t is the discount factor, for simplicity, it is always equal to $Exp(-r\Delta t)$. And $\eta_t^{i_1 i_2 \dots i_{tj}}$ is the 'decision value' as

$$\eta_t^{i_1 i_2 \dots i_{tj}} = \begin{cases} L_t(S_t^{i_1 i_2 \dots i_t}) & \text{if } L_t(S_t^{i_1 i_2 \dots i_t}) > \frac{1}{b-1}\sum_{i=1, i\neq j}^{b} d_t \psi_{t+1}^{i_1 i_2 \dots i_{ti}} \\ d_t \psi_{t+1}^{i_1 i_2 \dots i_{tj}} & \text{otherwise} \end{cases}$$

3.8 Least Squares Simulation Method

This method signals a definite breakthrough in American option pricing using Monte Carlo simulation. Its general intuition is that using cross sectional data to estimate options continual value. For every discrete exercise point, linear regression was performed to estimate option's continual value function. So the 'decision' exercises or continues can be made, this procedure is then carried out backward to time 0. Detailed methodology is not discussed and it can be found in Longstaff and Schwartz (2001).

The specifications used in this paper are: for single underlying asset, the basis functions are underlying price and squared underlying price. For two underlying assets, there are four basis functions which are underlying prices and squared underlying prices of both assets. For options with five underlying assets, there are two specifications: firstly, LSM is valued using five basis functions – first five Weighted Laguerre polynomials. Secondly: LSM is rerun using 11 basis functions – first five weighted laguerre, the product of the highest and the second highest value, second highest and third highest, etc and finally the product of all five values. In addition, LSM basis function specifications are further tested in section 6.2.

4. Comparisons of Valuation Techniques

In this section detailed comparisons will be performed for each methods outlined in last section. Since all Monte Carlo methods implemented in last section only have discrete exercise opportunities, comparisons of valuation techniques are divide into two subsections: first subsection compares numerical methods and second subsections compare simulation methods.

4.1 Comparisons of Numerical Methods

In this section we compare all the numerical methods described in section 3. Comparisons are based on two criteria: computation speed and results accuracy. Speed of computation is measured using average computation time per option. Accuracy of results is measured using Roots Mean Square (RMS) relative error. It is defined as,

$$\text{RMS} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} e_i^2}, \qquad \text{where} \quad e_i = \frac{\hat{C}_i - C_i}{C_i}$$

Where C_i is option true value. This measure is also used in Broadie and Detemple (1996). Options true values are computed using binomial method with 5000 time steps. This is the maximum number of time steps the PC (Note

2) under Visual Basic environment can effectively handle. It is worth noting that different hardware has different processing power. Such physical limitation is thus case specific.

Table 1 below compares speed and accuracy of the five methods for five call-option values. These parameters are identical to column 4, table 1 of BAW (1987). The results are discussed as follow: *Firstly*, the fastest method is quadratic approximation. Its average computation time per option is effectively 0. The internal clock under Windows XP environment is accurate to every millisecond (Note 3). However such accuracy is not sufficient to measure computation time for quadratic approximation, it is simply *too fast*. But it also has the largest error. (RMS reported are based on more significant digits than are shown in the tables) This is well expected. Secondly, binomial method is the second fastest with only 0.308 seconds to compute every option. Trinomial method has the longest computation time of 1.047 seconds per option. But it is also the most accurate method with the smallest error. This is also well expected since trinomial method has more nodes than that of binomial method for a given time steps. With the extra nodes generated, trinomial method is computationally more expensive and also more accurate. *Thirdly*, explicit finite difference dominates implicit finite difference in terms of both speed and accuracy.

Table 1. American call option values T=0.5, X=100, r=0.08, q=0.12, σ =0.2

S	Binomial	Trinomial	Explicit	Implicit	Quadratic	True
80	0.215	0.215	0.215	0.216	0.229	0.215
90	1.361	1.360	1.361	1.361	1.387	1.360
100	4.709	4.707	4.713	4.709	4.724	4.710
110	11.000	10.997	11.002	10.994	10.955	10.998
120	20	20	20	20	20	20
Time	0.328	1.047	0.541	0.572	0.000	
Error	4.5E-04	3.3E-04	1.8E-03	2.0E-03	2.9E-02	

1) Binomial and Trinomial methods are based on *n*=500 time steps.

2) Explicit Finite Difference method is based on *n*=500 time steps. *m*, number of steps in stock price, is then computed using $\Delta Z^2 = 3\Delta t\sigma^2$. In this case *m*=520 price steps.

3) Implicit Finite Difference method is based on *n*=500 time steps and *m*= 520 price steps.

4) Time is in seconds. It is average time spend in calculating above 5 option values.

5) Error is RMS as defined in the above paragraph.

6) The true value column is based on the binomial methods with *n*=5,000 time steps.

To make comparison meaningful, convergent parameters for implicit finite difference method is specifically chosen. Explicit finite method in table 1 has 500 time steps and according to the fixed ratio between ΔZ^2 and Δt, it has 520 price steps. Convergent parameters for implicit method are then set to 500 time steps and 520 price steps. As shown in the table, explicit method has both less computational time and lower relative mean error. Such result is consistent with Geske and Shastri (1985) which observed that explicit method is more efficient than implicit method.

Table 2 shows values of equivalent American put options. Results in table 2 are very similar to that of table 1. But some very interesting results are observed. All methods are less efficient in computing put option values except implicit method. As the computation time has shown, all three methods Binomial, Trinomial and Explicit need longer time to compute each option value. However small the time difference is, it will be very substantial if a large number of options are being valued. This is again consistent with previous literature. Merton (1973) showed that exercise boundary for put options must be checked at every instant, but call options may be exercised only at the ex-dividend dates. Geske and Shastri (1985) also showed that approximation techniques are more efficient for call options than for put options.

Table 2. American put option values T=0.5, X=100, r=0.08, q=0.12, σ =0.2

S	Binomial	Trinomial	Explicit	Implicit	Quadratic	True
80	20.957	20.957	20.958	20.958	20.982	20.957
90	12.634	12.634	12.634	12.635	12.645	12.633
100	6.365	6.364	6.371	6.368	6.372	6.367
110	2.650	2.649	2.652	2.648	2.650	2.648
120	0.918	0.918	0.920	0.922	0.919	0.918
Time	0.341	1.068	0.553	0.531	0.000	
Error	4.0E-04	2.9E-04	1.3E-03	1.9E-03	9.8E-04	

Note. All methods specification are identical to that of Table 1.

Surprisingly, implicit method shows shorter computational time for valuing put options. Such shorter computation time may be due to other factors (e.g. other operations that are running in the PC when options are valued) rather than the algorithm itself. Interestingly, running the programmed algorithm in other PCs (with variety of input parameters) showed that this relative time difference tends to disappear when large number of options are valued. When less than three options are valued, the method is still more efficient for put options. But checking computation time for each Visual Basic command line reveals that the timing difference is due to procedure that involves solving systems of equation. And the procedure for both call and put are exactly identical. This leaves with the only conclusion that internal clock of PC has small random errors every millisecond and it only accurate up to a certain point. Visual Basic Timer function may also have small random error every millisecond. When valuing only five options or less, these errors are quite substantial (since high level of accuracy is needed). But when many options are valued together, such random errors tend to be offset and disappear.

Another interesting finding in table 2 is that quadratic approximation shows accuracy that is even higher than that of finite difference methods. But this is somewhat expected, as already discussed in BAW (1987) that quadratic approximation has the highest accuracy for half year or less maturity.

Given results presented in table 1 and 2, it is quite clear that binomial method has relative advantage. It is computationally efficient and has relatively low error. But does these results stands for other input parameters?! Table 3 shows option values for many varieties of input parameters. Table 4 shows long maturity option values. Generally, results from both tables are consistent with Table 1 and 2. All techniques are less efficient for put options. The advantage of fixing the ratio between ΔZ^2 and Δt for explicit method is strongly demonstrated from Table 3. Explicit method has the second shortest computing time per option (see Table 3 second last row) with relatively good accuracy. However, this relative computation efficiency is not sustained for longer maturity option as shown in Table 4.

Secondly, results from both Table 3 and 4 shows binomial dominate trinomial method in terms of both accuracy and speed. All relative errors and time are in favor of binomial method. This result is inconsistent with many previous literatures which state that trinomial converges quicker. This result may largely due to the accuracy of the true option value. In fact, the true option value based on n=5,000 may simply not sufficiently accurate. It is well known that convergence of a binomial option price to the true price is not monotonic, but oscillatory in the step size. The "true value" reported therefore tends to favourably biased toward binomial method.

Table 3. American option values T=1.5, X=100

		Call Options						Put Options					
Param	S	Binomial	Trinomial	Explicit	Implicit	Quadratic	TRUE	Binomial	Trinomial	Explicit	Implicit	Quadratic	TRUE
	80	1.500	1.499	1.503	1.498	1.548	1.500	24.260	24.260	24.262	24.259	24.268	24.260
r=0.02	90	3.834	3.832	3.834	3.830	3.886	3.833	17.170	17.170	17.172	17.168	17.174	17.169
q=0.05	100	7.809	7.806	7.818	7.806	7.835	7.811	11.537	11.536	11.541	11.540	11.545	11.541
σ=0.2	110	13.573	13.565	13.577	13.562	13.531	13.570	7.406	7.398	7.403	7.401	7.405	7.403
	120	21.037	21.032	21.035	21.030	20.939	21.035	4.560	4.559	4.561	4.559	4.559	4.558
	80	10.237	10.235	10.248	10.230	10.455	10.239	33.824	33.827	33.830	33.820	34.078	33.826
r=0.05	90	14.614	14.606	14.618	14.603	14.838	14.614	29.155	29.157	29.168	29.150	29.396	29.158
q=0.1	100	19.723	19.711	19.740	19.714	19.941	19.727	25.131	25.128	25.143	25.131	25.362	25.139
σ=0.5	110	25.528	25.504	25.534	25.506	25.707	25.523	21.699	21.680	21.692	21.680	21.897	21.693
	120	31.945	31.923	31.944	31.920	32.078	31.937	18.745	18.733	18.739	18.722	18.925	18.738
	80	4.201	4.196	4.203	4.195	4.308	4.199	26.012	26.008	26.018	26.007	26.163	26.010
r=0.05	90	7.538	7.534	7.537	7.530	7.656	7.536	20.174	20.172	20.181	20.168	20.308	20.171
q=0.08	100	12.052	12.047	12.064	12.047	12.162	12.055	15.406	15.405	15.424	15.409	15.528	15.412
σ=0.3	110	17.723	17.720	17.720	17.712	17.790	17.723	11.630	11.633	11.642	11.628	11.727	11.632
	120	24.462	24.453	24.455	24.447	24.463	24.456	8.699	8.695	8.701	8.690	8.771	8.693
Time		0.330	1.051	0.203	0.526	0.000		0.349	1.070	0.222	0.526	0.000	
Error		2.2E-04	5.0E-04	7.4E-04	7.0E-04	1.4E-02		3.0E-04	3.2E-04	4.4E-04	3.6E-04	6.6E-03	

Note. All methods specification are identical to that of Table 1.

Table 4. Long maturity american option values X=100

Param	S	Call Options						Put Options					
		Binomial	Trinomial	Explicit	Implicit	Quadratic	True*	Binomial	Trinomial	Explicit	Implicit	Quadratic	True*
T=3	80	14.730	14.712	14.727	14.711	15.378	14.728	38.154	38.147	38.123	38.096	39.107	38.150
r=0.08	90	19.391	19.366	19.363	19.363	20.077	19.383	34.464	34.450	34.383	34.367	35.449	34.452
q=0.12	100	24.581	24.555	24.569	24.561	25.296	24.584	31.223	31.214	31.134	31.101	32.244	31.228
σ=0.5	110	30.304	30.274	30.250	30.270	31.000	30.297	28.411	28.402	28.234	28.216	29.420	28.404
	120	36.498	36.458	36.443	36.459	37.160	36.487	25.921	25.906	25.681	25.647	26.917	25.910
T=5	80	23.201	23.138	22.258	22.980	24.135	23.200	51.522	51.509	50.448	50.408	53.051	51.519
r=0.05	90	28.517	28.442	27.295	28.226	29.489	28.517	48.900	48.880	47.448	47.403	50.486	48.899
q=0.1	100	34.192	34.117	32.652	33.827	35.188	34.197	46.535	46.525	44.649	44.602	48.173	46.543
σ=0.6	110	40.220	40.136	38.304	39.755	41.206	40.218	44.422	44.400	42.008	41.962	46.069	44.416
	120	46.562	46.470	44.220	45.989	47.521	46.553	42.490	42.468	39.488	39.449	44.145	42.475
T=5	80	0.040	0.039	0.039	0.040	0.037	0.040	41.977	41.981	41.982	41.962	42.068	41.974
r=0.02	90	0.340	0.333	0.335	0.338	0.304	0.342	35.963	35.968	35.969	35.948	36.034	35.961
q=0.1	100	2.178	2.151	2.154	2.165	2.053	2.182	30.073	30.078	30.080	30.059	30.131	30.073
σ=0.1	110	10.000	10.000	10.000	10.000	10.000	10.000	24.451	24.455	24.456	24.439	24.501	24.454
	120	20.000	20.000	20.000	20.000	20.000	20.000	19.275	19.279	19.280	19.273	19.323	19.283
Time		0.329	1.054	0.153	0.522	0.000		0.347	1.070	0.170	0.527	0.000	
Error		2.9E-03	1.3E-02	3.0E-02	9.5E-03	5.1E-02		2.2E-04	2.5E-04	2.9E-02	2.9E-02	2.7E-02	

Note. All methods specification are identical to that of Table 1.

*True values are computed based on binomial method with n=4,000, this is due to PC memory limitation.

To achieve a better estimation of true option value, extrapolation technique may need to be employed. So that physical limitation of PC memory problem can be handled and option true values are estimated with greater accuracy.

Thirdly, by comparing results in table 3 and 4, it is very clear that when pricing long maturity option using finite difference methods more time steps and price steps are needed to ensure convergence. Since both methods have errors that only marginally smaller than that of quadratic approximation.

Fourthly, for explicit method it is very intriguing that the longer the maturity, the shorter the computation time. Closer look into the algorithm shows that fixed ratio between ΔZ^2 and Δt is no longer efficient for long maturity options. Given 500 time steps and 5 year maturity then price steps is computed to be only 50. This certainly will not result convergence. In fact, fixing ratio will not result satisfactory convergence if there are extreme parameters. (E.g. volatility greater than 1, interest rate near 0 etc.) Purely increasing time steps in this case will not solve the problem. Both time steps and price steps have to be separately specified to ensure convergence.

To sum up, all methods have their advantages. Quadratic approximation is superbly fast. Its computation time is effectively indifferent from zero. While it also shows good precision when maturity is short. Binomial method is the simplest to implement, it provides good accuracy and speed tradeoff. Advantage of trinomial method, however, is not so well demonstrated from the results because of the problems with the "true value". Explicit finite difference show very attractive efficiency when grid ratio is fixed, it also solved its inherent instability problem and always converge. However fixing grid ratio has difficulty when option parameters are extreme. Implicit finite difference can easily solve the extreme parameter problem by increasing grid density. But it is relatively slow because of inverting matrices.

4.2 Comparisons of Monte Carlo Methods

In this section, three simulation methods will be compared for a variety of parameters. Before proceeding to comparison, first let's look at the random number generator. Throughout this paper standard normal random numbers are generated using Box-Muller transformation. This choice is largely due to its simplicity. Although Box-Muller is computationally expensive, it will not significantly influence speed since valuing any single American option usually needs no more than 500,000 random numbers. Implementation of the generator follows directions from Steele and Douglas (2005). Testing of generator is not performed since it is comprehensively tested in Steele and Douglas (2005).

Table 5 shows values of seven American call options based on the three simulation methods. option parameters used here are identical to those in table 1 of Broadie, Glasserman, and Jain (1997). Quick glance at Table 5 reveals that all three methods can approximate option value quite tightly. And standard errors are fairly small for

most initial asset prices. Standard errors are highest for options with initial price 110. But computation time for bundling method is extensively long.

Table 5. American call option values using simulation   *T*=1, *X*=100, *r*=0.05, *q*=0.1, *σ* =0.2

	Bundling		LSM		Simulated Tree (ST)					
S	Value	s.e.	Value	s.e.	Value	High	s.e.	Low	s.e.	True
70	0.121	0.013	0.124	0.006	0.123	0.124	0.005	0.122	0.005	0.121
80	0.669	0.031	0.677	0.014	0.677	0.683	0.018	0.671	0.018	0.670
90	2.296	0.058	2.299	0.026	2.324	2.360	0.042	2.288	0.041	2.303
100	5.732	0.096	5.749	0.040	5.792	5.882	0.096	5.703	0.094	5.731
110	11.203	0.126	11.341	0.051	11.372	11.708	0.125	11.036	0.177	11.341
120	20.000	0.000	20.000	0.000	20.213	20.426	0.085	19.883	0.142	20.000
130	30.000	0.000	30.000	0.000	30.061	30.185	0.064	29.916	0.102	30.000
Time	2274.112		66.560		47.302					

1) Options only have four exercise opportunities at times 0, *T*/3, 2*T*/3 and *T*.
2) Bundling method is computed with 100 independent samples of 4,900 paths each using a partition of 70 bundles by 70 paths per bundle.
3) LSM is computed with 100 independent samples of 50,000 paths.
4) Simulated tree is computed using 100 independent estimation of both high and low estimators, *b*=50.
5) Time is averaged computation time per option, it is in seconds.
6) True values are based on binomial methods with *n*=1,200. Exercise opportunities are adjusted accordingly.

It takes about 2274 seconds to compute a single option value. That's roughly 40 minutes. Computing Bundling column of Table 5 took about five hours. Checking computation time for each command line shows that the problem lies mainly on paths reordering. Bundling method reorder price paths every time when there is an exercise opportunities. Reorder a single string of prices is simple task that can be done within millions fraction of a second, however the method requires reorder of all the paths that are related to the price. This is a two dimensional sorting procedure. Unless a quick algorithm is introduced or accuracy is not an important issue, using bundling method for American options valuation is practically impossible or at least cumbersome.

On the other hand, both LSM and Simulated Tree show splendid speed. With roughly one minute to compute each option. It is worth noting that ST will become computationally impossible if exercise opportunities are too large. If exercise opportunities increase to five, ST will not be able to price American option. This is largely due to data storage limit in the computer. Furthermore, computation time of ST grows exponentially with exercise opportunities. On the other hand, data storage requirement and computation time of LSM is not exponentially related to exercise opportunities. It can readily handle options with more than 100 exercise opportunities. LSM's advantage over the other two methods is obvious. It also has the lowest standard error for options valued.

Some more comparisons are reported in Table 6. To make table easy to read, standard errors are not reported. Error measure *RMS* is reported again for all options valued in the table. Average computation time per option is also reported. The first serial of option parameters in the table is identical to that of Tilley (1993), but option values are different from his because there are only four exercise opportunities. The results can be generally summarised as follow: *Firstly*, simulated tree is the fastest method with average computation time of only 48 seconds per option. However, its accuracy is not so satisfactory (*RMS* is greater than 1%).This is very strange result since branch parameters and numbers of independent samples used for ST computation are identical to that of Broadie, Glasserman and Jain (1997). Using 200 independent samples and compute option values (not reported) in the table again, the estimation error persists.

Table 6. American option values using simulation   *X*=40

		Call				Put			
	S	Bundling	ST	LSM	True	Bundling	ST	LSM	True
T=3	30	5.295	5.227	5.258	5.256*	10.000	10.088	10.000	10.000
r=0.07	35	8.244	8.312	8.272	8.27*	6.890	6.969	6.890	6.889
q=0	40	11.748	11.655	11.798	11.79*	4.912	4.851	4.946	4.947
σ=0.3	45	15.665	15.516	15.704	15.70*	3.558	3.478	3.569	3.568
	50	19.953	19.798	19.895	19.90*	2.606	2.616	2.592	2.593

T=3	30	4.132	4.156	4.191	4.196	16.927	16.968	16.970	16.968
r=0.07	35	6.106	6.173	6.142	6.152	15.096	15.239	15.087	15.091
q=0.14	40	8.350	8.546	8.437	8.441	13.485	13.480	13.482	13.482
σ=0.5	45	10.950	10.895	11.028	11.041	12.012	12.109	12.097	12.101
	50	13.513	13.980	13.895	13.904	10.904	10.926	10.897	10.900
T=1.5	30	0.175	0.178	0.177	0.177	11.888	11.882	11.868	11.870
r=0.07	35	0.775	0.746	0.768	0.769	8.350	8.350	8.324	8.327
q=0.14	40	2.220	2.283	2.227	2.227	5.436	5.440	5.431	5.430
σ=0.2	45	5.000	5.081	5.000	5.000	3.285	3.309	3.308	3.307
	50	10.000	10.011	10.000	10.000	1.891	1.893	1.895	1.895
T=1.5	30	1.839	1.835	1.814	1.814	14.758	14.886	14.728	14.727
r=0.02	35	3.357	3.337	3.387	3.389	11.884	11.982	11.867	11.868
q=0.1	40	5.515	5.634	5.538	5.544	9.521	9.529	9.487	9.487
σ=0.4	45	8.168	8.133	8.251	8.260	7.582	7.521	7.545	7.546
	50	11.252	11.423	11.464	11.473	5.996	5.887	5.980	5.981
Time		904.8	48.3	71.1					
Error		0.008	0.012	0.001					

Note. All specification are the same with Table 5 except Bundling method is computed with 10 independent samples of 10,000 paths each using a partition of 100 bundles by 100 paths per bundle. Error is measured in *RMS* as defined previously.

*True values are computed using Black-Scholes.

It seems that ST has relatively low precision when valuing option with maturity longer than one year. Other variance reduction techniques may be implemented with the method in order to achieve better estimates.

Secondly, Tilley's bundling method is still the slowest with roughly 15 minutes for each option. Though the results are not conclusive as discussed previously, the problem lies in the implementation. Speed could be improved if a more efficient reordering algorithm is introduced. (Using a more advanced programming language could also significantly increase speed, but here only relative speed matters.)

Finally, least square method has stunning accuracy. For most option valued, it is accurate up to 2 decimal places. Not only it has superior accuracy, it is also speedy. Its average computation time is 71 seconds per option, a little longer than ST, but with much greater precision. Application of LSM to higher dimensional problem is also straightforward, as discussed in later section.

To sum up, bundling method is a little complex to implement and its efficiency can only be improved with a quicker reorder mechanism. ST is fast but can only handle options with no more than four exercise opportunities. And its accuracy over long term options is relatively poor. LSM is the most promising technique with decent speed and superb accuracy, its speed and accuracy can be even better with some simple modification.

5. Conclusion

Most results and findings presented in this paper are consistent with previous literature. There are a few technical implementation problems where results contradict to previous findings. For example, advantage of trinomial method is not so strongly presented because possible biases in the option true values, which needs to be estimated with greater accuracy. And also the low efficiency of bundling method could be significantly improved if an alternative optimization procedure is implemented.

It is also worth noting that implementations can be further improved. For finite difference methods, price grid is recomputed every time a new option is valued. But *not* stored in the memory so that next option price can be simply retrieved from the grid. Efficiency of tree methods can also be improved by utilizing trees previously computed rather than re-computing whole tree again. If such improvements are implemented, conclusions reached may be different. But given the resources and time constraint, the results are most likely to be right for small investors who do not have to price large number of options every few hours. It may also be right for small companies which do not have a full department of computer programmers working on every aspects of computation optimization.

Overall the findings can be summarized as follows: Binomial are simple and efficient. Trinomial compensate efficiency with accuracy (Through not in second test). Explicit finite differences methods are very attractive when the grid ratio is fixed. Quadratic approximation is very fast but for long maturity option accuracy becomes a huge problem. Simulations are relatively slow even when dimensions are low. Among the three Monte Carlo

methods, LSM shows the best performance. Simulated tree is also very efficient, but its accuracy for long maturity options is disappointing. It is very difficult for simulated tree to price options with more than five exercise opportunities. Bundling methods can approximate option value with very good accuracy but not in an efficient way. Its computation time is about ten times that of LSM.

Although this paper examined some of the most popular American option pricing methods, there are still many research questions that are not explored or not explored in sufficient details. Many areas are worth further research: Firstly, code implemented may not be the most efficient. There are many areas that optimization is still possible, which could substantially change conclusions in the paper. For example, bundling method could be implemented with more efficient sorting procedure which can significantly improve efficiency. Finite difference methods could be implemented in a way that price grids are pre-stored, computation of option price only need to choose the right node from the stored grid. Tree methods can also be improved by utilizing trees previously computed rather than re-computing whole tree again. LSM can be implemented using matrix algebra for the cross sectional regressions rather than rely on Excel function "Linest".

Secondly, variance reduction techniques could be employed for simulation methods. Techniques that can increase precision of estimates are quite important. Some possible variance reduction techniques of ST method have already been introduced in Broadie, Glasserman, and Jain (1997).

References

Barone-Adesi, G., & Whaley, R. (1987). Efficient Analytic Approximation of American Option Values. *Journal of Finance, 42*, 301-320. https://doi.org/10.1111/j.1540-6261.1987.tb02569.x

Boyle, P. (1986). Option Valuation Using a Three-Jump Process. *International Options Journal, 3*, 7-12.

Boyle, P. (1988). A Lattice Framework for Option Pricing with Two State Variables. *Journal of Financial and Quantitative Analysis, 23*(1), 1-12. https://doi.org/10.2307/2331019

Boyle, P., Broadie, M., & Glasserman, P. (1997). Simulation Methods for Security Pricing. *Journal of Economic Dynamics and Control, 21*, 1267-1321. https://doi.org/10.1016/S0165-1889(97)00028-6

Boyle, P., Evnine, J., & Gibbs, S. (1989). Numerical Evaluations of Multivariate Contingent Claims. *Review of Financial Studies, 2*, 241-250. https://doi.org/10.1093/rfs/2.2.241

Brennan, M. J., & Schwartz, E. S. (1978). Finite Difference Method and Jump Processes Arising in the Pricing of Contingent Claims. *Journal of Financial and Quantitative Analysis, 13*, 461-474. https://doi.org/10.2307/2330152

Brennan, M., & Schwartz, E. (1977). The Valuation of American Put Options. *Journal of Finance, 32*, 449-462. https://doi.org/10.2307/2326779

Broadie, M., & Detemple, J. (1996). American Option Valuation, New Bounds, Approximations, and a Comparison of Existing Methods. *Review of Financial Studies, 9*(4), 1211-1250. https://doi.org/10.1093/rfs/9.4.1211

Broadie, M., & Glasserman, P. (1997). Pricing American-Style Securities Using Simulation. *Journal of Economic Dynamics and Control, 21*(8/9), 1323-1352. https://doi.org/10.1016/S0165-1889(97)00029-8

Broadie, M., Glasserman, P., & Jain, G. (1997). Enhanced Monte Carlo Estimates for American Option Prices. *Journal of Derivatives, 5*, 25-44,. https://doi.org/10.3905/jod.1997.407983

Broadie, M.,& Glasserman, P. (1998). *A Sotchastic Mesh Method for Pricing High-Dimensional American Options.* Working paper.

Carr, P., & Faguet, D. (1994). *Fast Accurate Valuation of American Options.* Working paper, Cornell University.

Cox, J., Ross, S., & Rubinstein, M. (1979). Option Pricing: A Simplified Approach. *Journal of Financial Economics, 7*, 44-50. https://doi.org/10.1016/0304-405X(79)90015-1

Geske, R., & Johnson, H. E. (1984). The American Put Option Valued Analytically. *Journal of Finance, 39*, 1511-1524. https://doi.org/10.1111/j.1540-6261.1984.tb04921.x

Geske, R., & Shastri, K. (1985). Valuation of Approximation: a Comparison of Alternative Approaches. *Journal of Financial and Quantitative Analysis, 20*, 45-72. https://doi.org/10.2307/2330677

Horasanlı, M. (2007). A Comparison of Lattice Based Option Pricing Models on the Rate of Convergence. *Journal of Applied Mathematics and Computation, 184*(2), 649-658. https://doi.org/10.1016/j.amc.2006.06.064

Hull, J. C. (2006). *Options, Futures and Other Derivative Securities.* Upper Saddle River, New Jersey, Prentice-Hall.

Hull, J. C., & White, A. (1990). Valuing Derivative Securities Using the Explicit Finite Difference Method. *Journal of Financial and Quantitative Analysis, 25*(1), 87-100. https://doi.org/10.2307/2330889

Kamrad, B., & Ritchken, P. (1991). Multinomial Approximating Models for Options with k State Variables. *Journal of Management Science, 37*(12), 1641-1652. https://doi.org/10.1287/mnsc.37.12.1640

Kim, I. J. (1990). The Analytic Valuation of American Options. *Review of Financial Studies, 3*, 547-572. https://doi.org/10.1093/rfs/3.4.547

Longstaff, F., & Schwartz, E. (2001). Valuing American Options by Simulation: A Least-Squares Approach. *Review of Financial Studies, 14*, 113-147. https://doi.org/10.1093/rfs/14.1.113

MacMillan, L. W. (1986). An Analytic Approximation for the American Put Price. *Advances in Futures and Options Research, 1*, 119-139.

Merton, R. C. (1973). Theory of Rational Option Pricing. *Bell Journal of Economics and Management Science, 4*, 141-183. https://doi.org/10.2307/3003143

Schwartz, E. (1977). The Valuation of Warrants: Implementing a New Approach. *Journal of Financial Economics, 4*, 79-93.

Steele, A., & Douglas, J. R. (2005). Simplifications from Simulations Monte Carlo Methods for Uncertainties. *NCSLI Measure*.

Stentoft, L. (2003). *Convergence of the Least Squares Monte-Carlo Approach to American Option Valuation.* University of Aarhus working paper.

Stentoft, L. (2004). Assessing the Least Squares Monte-Carlo Approach to American Option Valuation. *Review of Derivatives Research, 7*(2), 129-168. https://doi.org/10.1023/B:REDR.0000031176.24759.e6

Tilley, J. (1993). Valuing American Options in a Path Simulation Model. *Transactions of the Society of Actuaries*, 45.

Notes

Note 1. Available at: http://www.global-derivatives.com/index.php?option=com_content&task=view&id=14

Note 2. With 2.80-GHz Pentium 4 CPU.

Note 3. See 'About PC Clocks'. Available at
http://www.greyware.com/software/domaintime/technical/accuracy/pcclocks.asp

Copyrights