Note on the Efficiency of the Binomial Option Pricing Model

Dr Les Clewlow

and

Dr Andrew Carverhill

October 1995

Financial Options Research Centre
Warwick Business School
University of Warwick
Coventry
CV4 7AL
Phone: (01203) 524118

FORC Preprint: 96/66
(Revision of FORC Preprint 91/28)

A Note on the Efficiency of the Binomial Option Pricing Model

Les Clewlow and Andrew Carverhill

October 1995

Author for correspondence:

Dr. Les Clewlow (Senior Research Fellow)
Financial Options Research Centre
Warwick Business School
University of Warwick
Coventry
England
(UK) 0203 524118

Dr. Andrew Carverhill (Lecturer)
Department of Finance and Economics
The Hong Kong University of Science and Technology
Clear Water Bay
Kowloon
Hong Kong
(Hong Kong) 358 7672

We would like to acknowledge helpful discussions with Stewart Hodges and Archie Pitts and the comments of an anonymous referee. Any errors remain our own.

Funding for this work was provided by corporate members of the Financial Options Research Centre: HSBC, Mitsubishi Finance International, Morgan Grenfell, Swiss Bank Corporation, UBS.

A Note on the Efficiency of the Binomial Option Pricing Model

ABSTRACT

We discuss the efficiency of the binomial option pricing model for single and multivariate American style options. We demonstrate how the efficiency of lattice techniques such as the binomial model can be analysed in terms of their computational cost. For the case of a single underlying asset the most efficient implementation is the extrapolated jump-back method. That is to value a series of options with nested discrete sets of early exercise opportunities by jumping across the lattice between the early exercise times and then extrapolating from these values to the limit of a continuous exercise opportunity set. For the multivariate case, the most efficient method depends on the computational cost of the early exercise test. But, for typical problems, the most efficient method is the standard step-back method. That is performing the early exercise test at each time step.

1 Introduction

The binomial option valuation procedure was introduced by Sharpe (1978), Cox, Ross and Rubinstein (1979) and Rendleman and Bartter (1979). It can be used to value options where no analytical solution exists, in particular American put options. It can also be used where the underlying asset pays continuous or discrete dividends. Geske and Johnson (1984) introduced a method for pricing American put options based on the compound option model of Geske (1979). Breen (1991) has described an alternative implementation of the binomial procedure for American options based on the Geske and Johnson (1984) model. Previous authors have compared the efficiency of lattice methods for option pricing problems (see Smith (1976), Geske and Shastri (1985)) by timing the various methods for a range of inputs. This approach has the disadvantage that the results depend on the implementation of the method and on the characteristics of the computer used.

In this paper we demonstrate how the efficiency of lattice techniques such as the binomial procedure can be analysed and quantified independently of the implementation software and hardware. A similar technique was used by Kamrad and Ritchken (1991) to compare the efficiency of binomial and trinomial models. In this paper we generalise and formalise the procedure. Firstly, we consider the efficiency of the binomial procedure for American options on a single underlying asset. In this case we show that the most efficient method is the extrapolated jump-back method. This involves valuing a series of options with nested discrete sets of early exercise opportunities by jumping across the lattice between the early exercise times and then extrapolating from these values to the limit of a continuous exercise opportunity set. We then consider the efficiency of the binomial method for American options on multiple underlying assets. The most efficient method depends on the computational cost of the early exercise test. But, for typical problems, the most efficient method is the standard step-back method in which the early exercise test is performed at each time step to obtain the American option value directly.

In section 2 we briefly review the binomial procedure. Section 3 shows how the efficiency of lattice methods can be analysed using the standard binomial procedure as an example. In section 4 we analyse the efficiency of the binomial method for American options on a single underlying asset. Section 5 analyses the efficiency of the binomial method in the multivariate case. The summary and conclusions are in section 6.

2 The Binomial Option Pricing Model

In order to help our exposition, we briefly review the binomial procedure in its basic form. The idea is to replace the random process followed by the underlying asset by a binomial random walk in which the steps have the same mean and variance as the original process. We take the usual approach (Cox and Ross (1976)) and work with risk-neutral probabilities that is the price S_t of the underlying asset obeys the stochastic equation

$$dS_t = rS_t dt + \sigma S_t dB_t \tag{1}$$

in which r is the riskless interest rate and σ is the volatility of the asset (both assumed to be constant), and dB_t is the increment of a Standard Brownian Motion. The underlying process is replaced with a binomial random walk on the lattice, in which the time steps have length dt. Equation (1) implies that the proportional increments of the underlying over each time step are iid, and so it is possible either to think of the lattice as referring to proportional changes in the underlying, or to work with logarithms of the underlying; we prefer the latter. So, put $\log(S_t) = s_t$. Then the stochastic equation for s_t is

$$ds_t = (r - \tfrac{1}{2}\sigma^2)dt + \sigma dB_t \qquad (2)$$

From each node on the lattice, the random walk can either step up by a distance Δs_u with probability p_u, or it can step down by a distance Δs_d with probability p_d. These parameters must be chosen so that the mean and variance of the random walk over each time step match those of the process in equation (2). There is a degree of freedom in making this choice, because this subjects these 4 parameters to 2 constraints, together with the constraint $p_u + p_s = 1$. Two convenient choices are those of Cox, Ross, and Rubinstein (CRR) (1979), which imposes $\Delta s_u = -\Delta s_d$, and Jarrow and Rudd (JR) (1983), which imposes $p_u = p_s = \tfrac{1}{2}$.

At the lattice nodes corresponding to the maturity date of the option its value is known. From these values, one can then compute the option values backwards through the lattice by using the rule that at each node, the option value is the discounted expectation of its value at the next time step, i.e.

$$\phi_{i-1,j} = \exp(-rdt)(p^{-}\phi_{i,j} + p^{+}\phi_{i,j+1}) \tag{3}$$

where $\phi_{i,j}$ represents the option value after i time steps, if there have been j up-steps by time i .

To value American options, we test for early exercise at each node by comparing the value given by equation (3) with the value the option would yield if it were exercised at that time. If exercising makes the option worth more, we replace the result of equation (3) with the exercise value. We call this the *step-back* method.

For European options, it is possible to jump in one action over all the steps of the binomial lattice. Suppose the random walk has N steps in all up to the maturity date of the option. Then note that the lattice has $N+1$ nodes at the option maturity date, and that all paths leading to say the kth node (which corresponds to k up-steps) have the same probability, namely $(p_u)^k(p_d)^{N-k}$. Thus the probability of the binomial walk ending up at say the kth node at maturity is $\binom{N}{k}(p_u)^k(p_d)^{N-k}$. The present value of the option is therefore,

$$\phi_{0,0} = e^{-rT}\sum_{k=0}^{N}\binom{N}{k}(p_u)^k(p_d)^{N-k}\phi_{N,k} \tag{4}$$

where T is the time to maturity of the option[1]. We call this the *jump-back* method.

3 The Computational Efficiency of the Binomial Technique

We can calculate the relative efficiency of the jump-back and step-back methods by computing the number of basic computational operations involved. For both methods we must compute the $N+1$ node values of the option at maturity (terminal nodes), so we may ignore this in the comparison.

For the jump-back method we must sum over the $N+1$ terminal nodes the product of the probability of that node and the value of the option at that node (equation (4)). Now this can be reduced to three multiplications and an addition for each term in the summation[2] So the computation time is given by

$$(N+1)(3\tau_m+\tau_a) \tag{5}$$

where τ_m is the time required for a floating point multiplication and τ_a is the time required for a floating point addition.

The step-back method requires two multiplications and an addition for each non-terminal node (see equation (3), the discount factor can be combined with the up and down probabilities). At each step $i, 1 \le i \le N$, there are i nodes to evaluate, and therefore the computation time is

$$\sum_{i=1}^{N} i(2\tau_m+\tau_a) = \tfrac{1}{2}N(N+1)(2\tau_m+\tau_a) \tag{6}$$

Thus for $N > 3$ the jump-back method becomes progressively more efficient than the step-back method as N increases.

4 The Univariate Case for American Options

The standard implementation of the binomial option pricing model for American options is the step-back method since the early exercise test must be performed at each node. However, an alternative is to value options which are exercisable at a small number of dates (exercise opportunities) and to extrapolate from these to the American option value. This is similar to the approach adopted by Geske and Johnson (1984) and is also used by Breen (1991). As Omberg (1987) points out the sequences of exercise opportunities should be nested so that the option values are monotonically increasing. An example of a suitable nested sequence is $\{T\}$, $\{T, {}^{2T}\!/_{3}\}$, $\{T, {}^{2T}\!/_{3}, {}^{T}\!/_{3}\}$. Therefore, we first calculate the European option value ϕ_1, i.e. assume there is just one exercise opportunity, at the maturity date T of the option, by jumping over all the steps, as described in the previous section. Next, we assume that there are two exercise opportunities, at times ${}^{2T}\!/_{3}$ and T, and calculate the value ϕ_2 of the option by jumping first over the time interval $[{}^{2T}\!/_{3}, T]$, and then over $[0, {}^{2T}\!/_{3}]$. Last, we calculate the value ϕ_3, using three exercise opportunities, at times $T, {}^{2T}\!/_{3}, {}^{T}\!/_{3}$. Now, the sequence of option values $\{\phi_1, \phi_2, \phi_3, \ldots\}$. obtained by adding further exercise opportunities at the midpoints of the current intervals for example, converges to the American option value ϕ. We can extrapolate from $\{\phi_1, \phi_2, \phi_3\}$ to ϕ by using Richardson Extrapolation (see Geske and Johnson (1984)) to obtain

$$\phi = \phi_3 + \tfrac{7}{2}(\phi_3 - \phi_2) - \tfrac{1}{2}(\phi_2 - \phi_1) \tag{7}$$

We will call this the extrapolated jump-back method. A third approach is to use the step-back method to obtain the values ϕ_1, ϕ_2, and ϕ_3 instead of the jump-back method. We will deal with this extrapolated step-back method in section 5. We can calculate the computation time required to value an American option by both the step-back and extrapolated jump-back methods in the way we described in section 3. For the step-back method we simply need to add the time taken to perform the early exercise test at each node. We assume here that the computational cost of the early exercise test (CCEET) can be reduced to simply a floating point comparison and assignment and is therefore at least as fast as a floating point multiplication. If the CCEET is more computationally costly then the extrapolated jump-back method will be even more efficient relative to the step-back method. The computation time under our assumption is therefore

$$\tfrac{1}{2}N(N+1)(3\tau_m + \tau_a) \tag{8}$$

For the extrapolated jump-back method we must jump-back firstly over N nodes to reach the final node. Secondly over $N/2$ nodes for each of the $N/2$ mid-point nodes and the final node to reach the final node. Finally we must jump over $N/3$ nodes for the $2N/3$ two-third-point nodes, the $N/3$ one third-point nodes and the final node. The most efficient way to perform these multiple jump-backs is to pre-compute the one-dimensional array of probabilities which multiply the option values. This requires two multiplications (one for the binomial coefficient and one for the probability factor) for each probability. The computational time for a size $(M+1)$ array is therefore

$$2(M+1)\tau_m \tag{9}$$

The total computation time for the extrapolated jump-back method is therefore

$$\begin{aligned}&(N+1)(\tau_m+\tau_a)+2(N+1)\tau_m+\\&((N/2+1)(\tau_m+\tau_a)+\tau_m)((N/2+1)+1)+2(N/2+1)\tau_m+\\&((N/3+1)(\tau_m+\tau_a)+\tau_m)((2N/3+1)+(N/3+1)+1)+2(N/3+1)\tau_m\end{aligned} \tag{10}$$

which simplifies to

$$\left(\frac{7}{12}N^2+\frac{29}{3}N+17\right)\tau_m+\left(\frac{7}{12}N^2+\frac{9}{2}N+6\right)\tau_a \tag{11}$$

In this case N must be greater than approximately 10 before the extrapolated jump-back method becomes more efficient than the standard method. However N will normally be much greater than 10 in order to give reasonable accuracy. Asymptotically, as $N \to \infty$ the extrapolated jump-back method is 55% faster than the standard method for a CCEET of one floating point multiplication, and becomes increasingly more efficient as the CCEET increases (see Table 1).

5 The Multivariate Case

Boyle, Evnine and Gibbs (1989) (BEG) show how the binomial technique can be generalised to the case of n underlying assets involved in the option valuation. Examples of options to which this technique will apply include multi-currency options such as the option to convert the ECU into any one of its constituents at predetermined rates as well as spread options, that is options on the difference in price between two related assets.

Richardson extrapolation together with the step-back method can be applied in a straightforward way to the multivariate case. But jumping across nodes is not feasible. Firstly, if the assets are correlated all paths which reach the same end-point from a given starting point do not have the same probability. Secondly, even if the assets are uncorrelated (or if we transform to an orthogonal set of variables) the cost of computing the multinomial coefficients makes jumping back too inefficient. We can analyse the efficiency of the standard (step-back) BEG and extrapolated step-back BEG methods in a similar way to the one dimensional case. For the standard method each node depends on 2^n previous nodes and at step i there are i^n nodes. The computation time is therefore

$$\sum_{i=1}^{N} i^n \left(2^n (\tau_m + \tau_a) - \tau_a + \tau_m\right) \tag{12}$$

For the extrapolation method we must step-back over N steps, $N/2$ steps and $2N/3$ steps and perform the early exercise test at each exercise opportunity. The total computation time is therefore

$$\sum_{i=1}^{N} i^n \left(2^n(\tau_m+\tau_a)-\tau_a\right)+$$
$$\sum_{i=1}^{N/2} i^n \left(2^n(\tau_m+\tau_a)-\tau_a\right)+(N/2)^n\tau_m+$$
$$\sum_{i=1}^{2N/3} i^n \left(2^n(\tau_m+\tau_a)-\tau_a\right)+\left((2N/3)^n+(N/3)^n\right)\tau_m \quad (13)$$

The efficiency of the extrapolated method depends on the trade-off between the reduction in the number of early exercise tests and the increase in the number of nodes which must be computed.

We can obtain a quantitative comparison of the computational times in units of τ_m if we ignore the computation time for floating point additions (floating point additions are typically much faster than floating point multiplications). Table 1 gives the ratio of the computational times for the extrapolated and standard methods for one, two and three underlying assets, for a typical range of values of N and for a CCEET of one and five floating point multiplications.

As we can see from Table 1 the extrapolated method is in fact less efficient than the step-back method for a CCEET of one floating point multiplication. This is because, although the extrapolated method considerably reduces the number of early exercise tests we must perform, their computational cost is small compared with the computational cost of the extra nodes that must be computed in more than one dimension. The extrapolated method only becomes more efficient if the computational cost of the early exercise test is very high (for the two dimensional case around 20 floating point multiplications).

6 Conclusions

In this paper we have considered the efficiency, in terms of the computational cost, of the binomial pricing model for single and multiple underlying assets. We have shown how the efficiency of lattice methods such as the binomial can be analysed and quantified independently of the implementation software and hardware. For a single underlying asset, we have demonstrated that the jump-back method is the most efficient method. In the multi-variate case, the most efficient method depends on the computational cost of the early exercise test. However, for typical problems, the step-back method is the most efficient. In our analysis we have not considered the relative accuracy of the various methods for a number of reasons. Our analysis is independent not only of the software and hardware but also of the instrument we are pricing. However, the accuracy of the methods will certainly depend on the nature of the instrument being priced. Using our analysis it would be possible to compare the relative computational cost versus accuracy of these and other methods. This is left for further work.

Acknowledgements

We would like to acknowledge helpful discussions with Stewart Hodges and Archie Pitts and the comments of an anonymous referee. Any errors remain our own.

Funding for this work was provided by corporate members of the Financial Options Research Centre: HSBC, Mitsubishi Finance International, Morgan Grenfell, Swiss Bank Corporation, UBS.

Footnotes

[1]This can be written as $\phi_{0,0} = \sum_{k=0} Q_{N,k}\phi_{N,k}$ where $Q_{N,k}$ is the state price of node N,k (i.e. the price at time zero of a security that pays one unit at state N,k and zero in all other states).

[2]Assume the binomial coefficients are pre-computed and stored, pre-compute $P_k = (p_u)^k (p_d)^{N-k}, k = 0$, then for each subsequent term simply multiply P_k by p_u / p_d. In the case in which the probabilities vary through out the tree we can go back to the state price representation (see footnote 1). That is we assume that the state prices are pre-computed in which case we have one multiplication and one addition for each term in the summation.

References

Boyle, P., J. Evnine and S. Gibbs, "Numerical Evaluation of Multivariate Contingent Claims", Review of Financial Studies 2(2), pp 241 - 250, 1989.

Breen, R., "The Accelerated Binomial Option Pricing Model", Journal of Financial and Quantitative Analysis 26, pp 153 - 164, 1991.

Cox, J. C., and S. A. Ross, "The Valuation of Options for Alternative Stochastic Processes", Journal of Financial Economics 3, pp 145 - 166, 1976.

Cox, J. C., S. A. Ross and M. Rubinstein, "Option Pricing: a Simplified Approach", Journal of Financial Economics 7, pp 229 - 263, 1979.

Geske, R., "The Valuation of Compound Options", Journal of Financial Economics 7, pp 63 - 81, 1979.

Geske, R. and H. E. Johnson, "The American Put Valued Analytically", Journal of Finance 39, pp 1511 - 1524, 1984.

Geske, R. and K. Shastri, "Valuation by Approximation: A Comparison of Alternative Option Valuation Techniques", Journal of Financial and Quantitative Analysis 20, pp 45 - 72, 1985.

Jarrow, R. and A. Rudd, "Option Pricing", Homewood, IL: Dow Jones-Irwin, 1983.

Kamrad, B. and P. Ritchken, "Multinomial Approximating Models for Options with k State Variables", Management Science 37(12), pp 1640 - 1652, 1991.

Omberg, E., "A Note on the Convergence of Binomial Pricing and Compound Option Models", Journal of Finance 42, pp 463 - 469, 1987.

Rendleman, R. J. and B. J. Bartter, "Two-State Option Pricing", Journal of Finance 34(5), pp 1093 - 1110, 1979.

Sharpe, W. F., "Investments", Englewood Cliffs, NJ: Prentice-Hall, 1978.

Smith, C.W. Jr., "Option Pricing: A Review", Journal of Financial Economics 3, pp 3 - 51, 1976.

Table 1

Ratio of Computational Times for Extrapolated Jump-back and Extrapolated Step-back Binomial Methods

	Computational Cost of the Early Exercise Test							
	1 floating point multiplication				5 floating point multiplications			
N	*n*=1#	*n*=1*	*n*=2*	*n*=3*	*n*=1#	*n*=1*	*n*=2*	*n*=3*
10	1.048	1.279	1.223	1.168	0.668	0.756	0.801	0.868
20	0.706	1.208	1.182	1.146	0.400	0.627	0.722	0.826
30	0.597	1.183	1.168	1.138	0.319	0.581	0.693	0.810
40	0.544	1.170	1.160	1.133	0.279	0.557	0.679	0.802
50	0.512	1.162	1.156	1.131	0.256	0.543	0.669	0.797
60	0.491	1.157	1.153	1.129	0.241	0.533	0.663	0.793
70	0.477	1.153	1.151	1.128	0.230	0.526	0.659	0.791
80	0.465	1.150	1.149	1.127	0.222	0.521	0.656	0.789
90	0.457	1.148	1.148	1.126	0.215	0.517	0.653	0.787
100	0.450	1.146	1.147	1.125	0.211	0.514	0.651	0.786

N is the number of time steps

n is the dimensionality of the lattice

\# Extrapolated jump-back method

* Extrapolated step-back method