

A Depth-First Search Technique For The Valuation Of American Path-Dependent Derivatives*

Patrick Dennis, (Email: pjd9v@virginia.edu), University of Virginia

Abstract

Pricing path-dependent American options is difficult since the number of paths through a binomial tree grow exponentially with the number of binomial periods. In practice even moderately sized trees of 20 to 25 periods can quickly exhaust available memory on most computer systems. This paper describes a method that can be used to price path dependent American style derivatives where the amount of memory grows linearly, not exponentially, in the number of binomial periods. The method is applied to pricing Asian options, fixed income derivatives based on the Heath-Jarrow-Morton model, and corporate bonds.

1. Introduction

Pricing American options on path-dependent derivatives is a difficult computational problem. The model for the underlying variable on which the option is written is typically based on a binomial tree, which means that the number of possible paths through the tree grows exponentially with the number of time steps. Due to this exponential growth in the number of paths through the tree, computer resources can quickly be exhausted when pricing American path-dependent derivatives. However, the ability to price these securities is critical for investment professionals. For example, as of the fourth quarter of 1999 commercial banks had a notional \$35 trillion dollars of derivatives in their portfolios, and 80%, or \$28 trillion, were interest rate derivatives¹. Most of these interest rate derivatives have American features, such as callability or prepayment options, and one of the most widely used models to price and hedge in-

terest rate derivatives is the Heath, Jarrow and Morton (1992) model, which exhibits path dependence.

This paper describes an algorithm, based on the depth-first search technique from graph theory², which can be used to value any path-dependent American derivative security using only one memory location for each time step in the binomial tree³. While the computational time still grows exponentially, the algorithm presented here makes it possible to value American path dependent derivatives using binomial trees of arbitrary size with minimal amounts of computer memory. This method provides a feasible method to price these securities for both academics and practitioners. The algorithm is illustrated using two common examples of path-dependent American style derivatives, namely American Asian options written on the arithmetic average of the stock price, and American style fixed income derivatives where the Heath-Jarrow-Morton model is used to model the evolution of

Readers with comments or questions are encouraged to contact the author via email.

Morton model is used to model the evolution of the forward rate curve. Applications of the algorithm to pricing corporate bonds are also discussed.

If a derivative is path dependent, but European in nature, there often exist techniques that can be used to obtain the exact or approximate price of the derivative. Take, for example, average price Asian options, where the option is written on either the arithmetic or geometric average asset value. While a closed form solution is available for European Asian options using a geometric average (Turnbull & Wakeman (1991)), this is not the case for the arithmetic average value. One can approximate the value for an arithmetic average European Asian option using an Edgeworth expansion to approximate the probability density function for the arithmetic average (Turnbull & Wakeman (1991)). When the price of a path-dependent European option cannot be obtained in closed-form or be approximated analytically, Monte-Carlo simulation can be used to estimate the price of the option (Boyle (1977), Kemna & Vorst (1990)).

While techniques such as Monte Carlo simulation are often used by investment professionals to price path-dependent European options, pricing path-dependent American options is an even more complex problem due to the early exercise decision. Although there has been some recent progress in using Monte-Carlo simulation to value American options (Brodie, Glasserman & Jain (1997)), and using certain approximation techniques for valuing American Asian options (Hull & White (1993), Cho & Lee (1997)), pricing American path-dependent derivatives is still a difficult problem facing academics and practitioners alike.

The most common approach to pricing American path-dependent options is to use a discrete-time approximation to the continuous-time process followed by the underlying asset, and employ standard dynamic programming techniques to value the American option. To price path dependent equity derivatives the binomial

approximation developed by Cox, Ross, & Rubinstein (1979) and Rendleman and Bartter (1979) is typically used. Using these approximations usually results in the number of paths through the tree growing exponentially in the number of binomial periods used⁴. To price interest-rate derivatives based on a no-arbitrage term structure model, some form of the Heath-Jarrow-Morton (HJM) (1990, 1992) model is usually employed. Since the forward rate volatility in the HJM model can take very general forms, the model usually results in path-dependent discrete-time approximations⁵.

It is a common assumption that it is not feasible to value American style path dependent derivatives using binomial trees with many periods due to the fact that there is not enough memory to keep track of all possible paths through the tree. For example, a T period American Asian option would have 2^T possible paths through the tree. This means that a 30 period tree results in $2^{30} = 1,073,741,824$ paths.

Using the method presented in this paper, it is possible to value T period path dependent American option using only kT memory locations, where k is some small integer⁶. This results in a substantial savings in computer resources, and allows many more time-steps to be used when computing the price of path-dependent derivatives. As a result of more time steps in the binomial tree, derivative prices that are closer to the continuous-time limit can be obtained, enabling practitioners to get more accurate prices and hedge parameters. The paper is organized as follows. Section 2 describes the algorithm in general terms. Section 3 applies the results to the pricing of American Asian options, interest rate derivatives based on the HJM model, and corporate bonds. Section 4 concludes the paper, and Section 5 provides suggestions for future research.

2. Algorithm

2.1 Intuition

The key to the algorithm is that at each time

only one floating point memory location is needed to store all information relevant to computing the price of the derivative. Because of this, one only needs a $T \times 1$ array to compute the price of a T period path-dependent derivative. To understand why only one memory location is needed at each time, consider figure 1. Let $p(t, s)$ denote the value of the underlying state variable on which the option is written and $d(t, s)$ denote the value of the derivative security at time t and state s . We want to use information at time $t+1$ and beyond to compute the value of the derivative security at time t , state s , $d(t, s)$. To do this we must first determine the value of the derivative security at time $t+1$, state s (the "down" state), and time $t+1$, state $s+1$ (the "up" state). Given the derivative security values at time $t+1$, the value of the derivative security at time t is simply the expected value under the risk-neutral measure, discounted at the risk free rate.

After the value of the derivative is computed at time $t+1$, state s , it is stored in the memory location allocated to time $t+1$, denoted by $m(t+1)$. Next, the value of the derivative security at time $t+1$, state $s+1$, $d(t+1, s+1)$, is computed. Once $d(t+1, s+1)$ is computed, the value of the derivative at time t can be found by discounting the expected value of the derivative at time $t+1$ at the risk free rate of interest:

$$d(\tau, s) = \left(\frac{\pi d(\tau+1, s+1) + (1-\pi)d(\tau+1, s)}{e^{r\Delta\tau}} \right),$$

where r is the annual risk free rate, Δt is the length of one binomial period, and π is the risk-adjusted probability of an up movement in the underlying state variable.

The key to the algorithm is that after $d(t, s)$ is computed, it is stored in memory location $m(t)$, freeing up $m(t+1)$ to be used when computing the next $t+1$ derivative security value. In this fashion, only one memory location is needed at each time step to compute the value of the derivative security. As soon as enough information is available at time $t+1$ to compute $d(t, s)$, $d(t,$

$s)$ is computed and stored in $m(t)$, which frees $m(t+1)$ for the next iteration. Because of this, at most one memory location is occupied at each time, and a T period path-dependent derivative can be calculated using only T floating point memory locations.

2.2 Details

To understand the details of the method, consider the three-period path-dependent binomial tree shown in figure 2. We want to value a three-period American put option with strike price K on the underlying path-dependent state variable $p(t, s)$. Two of the more common state variables, which lead to path-dependent pricing problems, are the average stock price in the case of Asian options and the price of a bond in the case of the Heath-Jarrow-Morton stochastic term-structure framework. Both of these cases will be discussed later.

Beginning at time 3, the boundary condition is imposed to value the put option in the uppermost states: $d(3, 7) = \max(0, K - p(3, 7))$, $d(3, 6) = \max(0, K - p(3, 6))$. Using these values, the price of the put option at time 2, state 3 is determined by discounting the risk-adjusted expected payoff at time 3 at the risk free rate, and then imposing the early-exercise condition:

$$d(2,3) = \max \left(K - p(2,3), \frac{\pi d(3,7) + (1-\pi)d(3,6)}{e^{r\Delta\tau}} \right) \quad (1)$$

The value of the put option at time 2, $d(2, 3)$, is then stored in memory location $m(2)$ for future use. Next, the value of the put option at time 2, state 2, $d(2, 2)$, is computed from the discounted expected value at time 3 in a manner identical to equation (1). Having just computed $d(2, 2)$, the value of the put option at time 1, state 1, $d(1, 1)$, is the larger of the exercise value $K - p(1, 1)$, or the expected payoff at time 2 using the risk-adjusted probabilities discounted at the risk free rate.

The value of the put $d(1, 1)$ is stored in memory location $m(1)$ for later use. It is impor-

tant to note that at this point memory location $m(1)$ is in use, but memory location $m(2)$, which used to contain the value $d(2,3)$, is not needed anymore and is free to be used for future computations.

To obtain the final price of the put, $d(0, 0)$, we have to find $d(1, 0)$. To accomplish this, we begin, again, at the maturity of the option. The value of the put at time 2, state 1, $d(2, 1)$, is computed from the boundary values in a manner identical to equation (1), and is temporarily stored in memory location $m(2)$. The value of the put at time 2, state 0, $d(2, 0)$ is similarly determined from the boundary values, and the value of the put at time 1, state 0, $d(1,0)$, is the larger of the exercise value, $K-p(1,0)$, or the expected value at time 2 discounted at the risk free rate.

Finally, the present price of the put option is the expected value at time 1 discounted at the risk free rate, $d(0, 0) =$

$$\left(\frac{\pi d(1,1) + (1 - \pi) d(1,0)}{e^{r\Delta\tau}} \right),$$

where the value $d(1, 1)$ was previously stored in $m(1)$.

The key to this algorithm is the fact that the computations are carried out in such a way that only one memory location is needed for each time step. This means that only a 100 x 1 floating point array is needed to value a 100 period American Asian option or other path dependent derivative, even though $2 + 2^2 + \dots + 2^{99} + 2^{100} = 2^{101} - 2$ nodes occur in the binomial tree.

3. Applicability of Results

While this algorithm can be used to price any path-dependent derivative, the three broadest classes of path-dependent derivatives that it may be applied to are Asian options, interest rate options, and corporate debt options. Section 3.1 applies the method to arithmetic Asian options,

one of the simplest classes of derivatives that result in path-dependent lattices. Section 3.2 applies the method to interest rate options that are priced using sophisticated Heath-Jarrow-Morton type term-structure models. Last, Section 3.3 discusses how the method can be used to price corporate bond options.

3.1 Application to American Asian Options

In this section we apply the method to price American Asian options based on an arithmetic average stock price. Let ω index each complete path through the tree from time 0 to time T . For a T period tree, $\omega \in (0, 1, 2, \dots, 2^T - 1)$. To use notation consistent with figure 2 and the previous section, let the state at time t along path ω ,

$s(t, \omega)$, be given by $s(t, \omega) = \text{int} \left(\frac{\omega}{2^{T-t}} \right)$ and let

$v(t, s(t, \omega))$ denote the value of the stock at time t along path ω . Then the average stock price at time t in state s , $p(t, s(t, \omega))$, is the underlying state variable on which the option is written and is defined as:

$$p(t, s(t, \omega)) = \frac{1}{t} \sum_{j=1}^t v(t, s(t, \omega)). \quad (2)$$

With the value of the underlying defined in (2), we can value the American Asian option as described in section 2.

As an example, assume that the underlying stock currently has a value of \$100, an annual volatility of $\sigma = 20\%$, and that the annual continuously compounded riskless rate is $r = 10\%$. Say that we want to value a three-year American Asian put option with a strike price of \$100 using a three-period binomial model. In this case, the up and down factors for the stock price movement are $u = e^{\sqrt{\sigma\Delta t}} = 1.2214$ and $d = e^{-\sigma\sqrt{\sigma\Delta t}} = 0.8187$, and the risk-adjusted probabilities are $\pi = \frac{e^{-r\Delta t} - d}{u - d} = 0.7114$ and $1 - \pi = 0.2886$.

Figure 3 shows the evolution of the arithmetic average stock price (top number) and Asian option price (bottom number) for the three-period model. There are a total of 8 paths through the tree, $\omega \in 0, 1, \dots, 7$, and the numbers in parentheses represent the state indices, $s(t, \omega)$, along each path ω . Beginning with the boundary condition at time 3 in the lowest two states (state 0 and 1), it is optimal to exercise the option when the average stock price is 67.93 (exercise value = $100 - 67.93 = 32.07$), and when the average stock price is 76.93 (exercise value = $100 - 76.93 = 23.07$). Using these values, the option value at time 2, state 0, is $e^{-0.1} ((\pi)(23.07) + (1-\pi)(32.07)) = 23.23$. The value of the option if exercised, however, is $100 - 74.45 = 25.55$. This value (25.55), which represents the value of the option at time 2 state 0, is saved in memory location $m(2)$ for later use. Using a similar calculation, it can be shown that it is also optimal to exercise the option at time 2, state 1, so the option is worth 9.06 in this state. The value for the option at time 1, state 0 can now be determined from the value of the option at time 2, state 0 (25.55), which was previously stored in memory location $m(2)$, and the value of the option at time 2, state 1 (9.06). Specifically, the value for the option at time 1, state 0 is $\$e^{-0.1} ((\pi)(9.06) + (1-\pi)(25.55)) = 12.51$. The value of the option if exercised, however, is $100 - 81.87 = 18.13$. The value for the option at time 1, state 0, 18.13, is saved in memory location $m(1)$ for later use. Note that memory location $m(2)$ is now available for future computations. Again, the way that memory locations are recycled means that the memory requirements grow linearly, not exponentially, in the tree size.

Using a similar algorithm, the option price at time 1, state 1, is determined to be 0. This value, along with the value of the option at time 1, state 0, which was previously saved in memory location $m(1)$, can be used to determine the option value at time 0: $e^{-0.1} ((\pi)(0) + (1-\pi)(18.13)) = 4.73$.

Table 1 shows the option values, number of paths, and computational times for option pricing problems from 3 to 30 periods. There are several interesting points to note. First, as can be seen from the table, while the number of paths increases exponentially with the number of periods, the memory required only increases linearly with the number of periods. Second, in this case, it appears that models with 25 to 30 periods are needed to get an option price with an error of less than one cent. Implementing American-Asian pricing models of this scale using regular techniques would not be possible on any computer with less than 100 Million bytes of memory. Last, and unfortunately, the computational time still increases exponentially with the number of paths. Hence, while the algorithm overcomes the memory constraint, processing speed is still a limiting factor in implementing very large models.

3.2 Application to Heath-Jarrow-Morton Term Structure Models

Path-dependent option pricing problems not only result from valuing Asian options, but also from valuing interest rate derivatives using certain forms of the Heath-Jarrow-Morton (1990, 1992) model. In particular, HJM models where the volatility of the forward rate is not constant can lead to path-dependent options. While European interest rate derivative securities can be valued using Monte Carlo simulation, American style derivatives must be valued using dynamic programming and a discrete time approximation to the continuous time process.

Let the present time be $t = 0$. Say we want to value a call option, expiring at time T_c , on a coupon bond maturing at time T_b , where $0 < T_c < T_b$. Let t be some time between the present time, $t = 0$, and when the call option matures, T_c , or $0 \leq t < T_c$, and let $B(t, T, s)$ be the price at time t , state s of a \$1.00 par zero coupon bond maturing at time T . The Heath-Jarrow-Morton model gives us the arbitrage free evolution of the term structure between 0 and T_b . At any time t

Table 1
American Asian Option Valuation

This table shows the American Asian option values, based on an arithmetic average stock price, for binomial tree sizes from 3 to 30 periods. The memory required is in terms of the number of double-precision floating-point locations.

Number of Periods	Number of Paths	Memory Required	Option Value
3	8	3	4.730
5	32	5	4.802
10	1,024	10	4.385
15	32,768	15	4.077
20	1,048,576	20	3.858
25	33,554,432	25	3.742
30	1,073,741,824	30	3.753

and state $s(t, \omega)$ there are two possible outcomes for the state of the world at time $t + 1$ - bond prices can increase, the "up" state, or bond prices can decrease, the "down" state. Given the structure of the forward rate volatilities, $\sigma(t, T, s)$, one can use the HJM model to find the shape of the term-structure in these "up" and "down states". These results are given in appendix 1. Since the forward rate volatilities can be state and time dependent, the sum of the forward volatilities along a path consisting of an up move followed by a down move will, in general, be different than a down move followed by an up move, and the tree will not recombine.

To illustrate the method, we will use the HJM model to value an American option on a coupon bond. Given the initial term structure, $B(0, T, 0)$, and the structure of the forward rate volatilities, $\sigma(t, T, s)$ for $0 < t < T < T_b$, one can work forward through the tree and compute the arbitrage-free evolution of the term structure. Knowing the term structure at any time t and state $s(t, \omega)$, the coupon bond can be valued. This is necessary since the coupon bond, whose price is denoted by $p(t, s(t, \omega))$, is the asset on which the option is written. To value the coupon bond at time t along path ω , we need to know the amount and timing of the cash flows that the bondholder receives. Let CF_j denote the cash flow that the bondholder receives at time j , which may consist of coupon interest or coupon

interest plus principal. The value of the coupon bond on which the option is written can then be expressed as the present value of all future cash flows,

$$p(t, s(t, \omega)) = \sum_{j=t+1}^{T_b} CF_j B(t, j, s(t, \omega))$$

To see how the algorithm works with the HJM model, consider an American call option with a strike price of \$100.00 expiring in 1.5 years written on a \$100 par bond maturing in 2 years. The current term structure is flat, with an annual spot rate of 2% per year, quoted as an annual effective rate. To make the bond sell at par, the semi-annual coupon is set to $\$1.02^{0.5} - 1 = 0.99505\%$ of par. The volatility of the one-period forward rate, $\sigma(t, T, s)$, is proportional to the level of the forward rate, $\sigma(t, T, s) = \eta(f(t, T, s) - 1)$ where $f(t, T, s) \equiv \frac{B(t, T, s)}{B(t, T + 1, s)}$, and η is set to 0.1.

Table 2 contains the price of the American call option for tree sizes ranging from 3 to 21 periods. Though the number of paths through the tree grow exponentially, the amount of memory required to price the option only grows linearly with the number of periods. As with the Asian option, the computational time still grows

Table 2
Bond Option Values Using the Heath-Jarrow-Morton Model

This table shows the price, number of paths, and amount of memory for pricing a 1.5 year call option with a strike of \$100 on a 2 year \$100 par bond paying a semi-annual coupon of \$0.99505. The HJM model is used to determine the arbitrage free term structure evolution. The memory required is in terms of the number of double-precision floating point locations. For a T period model, one $1 \times T$ array is needed to hold the forward rate curve, one $1 \times T$ array is needed to hold the one-period riskless rate, and one $1 \times T$ array is needed to hold the option value.

Number of Periods	Number of Paths	Memory Required	Option Value
3	8	$2 \times 3 = 6$	0.11254
6	64	$2 \times 6 = 12$	0.54487
9	512	$2 \times 9 = 18$	0.70554
12	4,096	$2 \times 12 = 24$	0.79121
15	32,768	$2 \times 15 = 30$	0.83866
18	262,144	$2 \times 18 = 36$	0.87325
21	2,097,152	$2 \times 21 = 42$	0.89560

exponentially with the tree size. Pricing fixed income derivatives using the HJM model is slightly more complicated than pricing Asian options since at time t along path ω a $1 \times T$ array is needed to store the term structure. As the term structure evolution is computed, this same array can be used at each point along path ω . Because of the need to keep track of the entire term structure, two $1 \times T$ arrays are needed - one for the term structure, and one to provide temporary storage for the price of the derivative at time t . This is why the number of floating point locations required to compute the option value is twice the number of time steps.

3.3 Application to Corporate Bond Options

Pricing corporate bonds options can be very difficult, since one has to account for both interest rate risk and risk to the firm's assets not due to interest rate risk (Amin and Jarrow (1992)). In the interest of space, an a detailed example applying this to corporate debt has been omitted, but the general procedure is straightforward and is outlined here. In the case of corporate debt options, such as a convertible bond, there are two state variables, one accounting for interest rate risk and one for non-interest rate risk. This can result in very large lattices, since each state

would have four successors - (assets up, interest rates up), (assets up, interest rates down), (assets down, interest rates up), (assets down, interest rates down). In this case, one could price the derivative using three temporary floating point memory locations for each time step. Once the fourth value of the derivative is calculated at time $t+1$, the discounted expected value can be taken, and the result stored in a memory location at time t , freeing up the three temporary memory locations at time $t+1$ for future use. Generally, if s state variables exists, each modeled by a binomial process, then $T(2^s-1)$ memory locations would be needed to price a derivative based on a T period tree.

4. Conclusion

Pricing path-dependent American options using discrete-time trees is difficult due to the amount of computational resources required. The algorithm presented in this paper allows the prices of these options to be computed using very small amounts of memory, overcoming one of the difficulties in obtaining accurate prices for path-dependent derivatives. Specifically, the amount of memory required grows only linearly, not exponentially, with the number of time-steps in the binomial tree. Given that the notional

value of path dependent derivatives with American features held by banks is roughly twenty trillion dollars, this is an important problem to practitioners. The method is applied to pricing American style Asian options, pricing options on a coupon bond where the Heath-Jarrow-Morton model is used to model the stochastic evolution of the forward rate curve, and pricing corporate bond options.

5. Suggestions for Future Research

While the algorithm presented in this paper overcomes the memory constraint imposed by pricing path-dependent derivatives with many binomial periods, the computational time required still grows exponentially as the number of periods in the tree increases. Future pricing algorithms may be able to significantly reduce the computational time required by trying to identify the early exercise boundary, within the context of the discrete time approximation of the continuous time process. Once the early exercise boundary is identified, pricing the option would be a simple matter of using the risk-adjusted probability measure to find expected payoff along the early exercise boundary, and then discount the expected payoff at the riskless rate.

The paper has benefited from the comments of Don Chance, Wake Epps, Gayle Erwin, and Larry Kochard. The author thanks the McIntire School of Commerce for financial support. Address all correspondence to Monroe Hall, University of Virginia, Charlottesville, VA 22901. Phone: 804-924-4050, FAX: 804-924-7074, E-Mail: pjd9v@virginia.edu.

6. Endnotes

1. See the Office of the Comptroller, Currency Bank Derivatives Report, Fourth Quarter 1999.
2. See Tarjan (1972) for an analysis of the depth-first search technique. Baase (1988, p. 172) provides a somewhat less technical description.
3. For a derivative written on a single state

variable, this algorithm requires locations for a T period tree. The method can be easily extended to price derivatives written on s state variables using only $(T)(2^s - 1)$ memory location.

4. Hull and White (1993) show that to value path-dependent options, one only has to value the option for all possible values of the path function at a given node in the tree. For some path-dependent options, such as lookback options, the number of possible values for the path function does not grow too fast (exponentially) with the size of the tree. This results in significant gains in computational efficiency.
5. In the special case of the HJM model where the forward rate volatilities are constant, the HJM model reduces to the HO-Lee model (1986). In this case, the discrete-time approximation is not path-dependent.
6. For the applications illustrated in this paper, $k \leq 2$.
7. Jarrow (1996) provides a very accessible derivation of the HJM model.

7. References

1. Amin, Kaushik and Robert A. Jarrow, "Pricing Options on Risky Assets in a Stochastic Interest Rate Economy," *Mathematical Finance* Vol. 4, pp. 217-237, 1992.
2. Baase, Sara, *Computer Algorithms*, Second Edition, 1988, Addison-Wesley.
3. Boyle, Phelim, "Options: A Monte Carlo Approach," *Journal of Financial Economics* Vol. 4, pp. 323-338, 1977.
4. Brodie, Mark, Paul Glasserman, and Gautam Jain, "Enhanced Monte Carlo Estimates for American Option Prices," *Journal of Derivatives*, pp. 25-44, Fall 1997.
5. Cox, John, Stephen Ross, and Mark Rubinstein, "Option Pricing: A Simplified Approach," *Journal of Financial Economics* Vol. 7, pp. 229-263, 1979.
6. Cho, He Youn, and Hee Yong Lee, "A Lattice Model for Pricing Geometric and Arithmetic Average Options," *Journal of*

- Financial Engineering* Vol. 6, pp. 179-191, 1997.
7. Heath, David, Robert Jarrow, and Andrew Morton, "Bond Pricing and the Term Structure of Interest Rates: A Discrete Time Approximation," *Journal of Financial and Quantitative Analysis* Vol. 25, pp. 419-440, 1990.
 8. Heath, David, Robert Jarrow, and Andrew Morton, "Bond Pricing and the Term Structure of Interest Rates: A New Methodology for Contingent Claims Valuation," *Econometrica* Vol. 60, pp. 77-105, 1992.
 9. Ho, Thomas S. Y. and Sang-bin Lee, "Term Structure Movements and Pricing Interest Rate Contingent Claims," *Journal of Finance* Vol. 41, pp. 1011-1029, 1986.
 10. Hull, John, and Alan White, "Efficient Procedures for Valuing European and American Path-Dependent Options," *Journal of Derivatives*, pp. 21-31, Fall 1993.
 11. Jarrow, Robert A., *Modeling Fixed Income Securities and Interest Rate Options*, First Edition, 1996, McGraw-Hill.
 12. Kemna, A.G.Z. and A.C.F. Vorst, "A Pricing Method for Options Based on Average Asset Values," *Journal of Banking and Finance* Vol. 14, pp. 113-129, 1990.
 13. Rendleman, Richard J. and Brit Bartter, "Two-State Option Pricing," *Journal of Finance* Vol. 34, pp. 1093-1110, 1979.
 14. Tarjan, Robert. "Depth-First Search and Linear Graph Algorithms," *SIAM Journal on Computing* Vol. 1, pp. 146-160, 1972.
 15. Turnbull, Stuart and Lee Wakeman, "A Quick Algorithm for Pricing European Average Options," *Journal of Financial and Quantitative Analysis* Vol. 26, pp. 377-389, 1991.

Appendix 1

Let Δt be the length of one binomial period, and $\sigma(t, T, s)$ be the volatility, at time t , of the forward rate realized at time $t+1$ that is in effect between T and $T+1$. The HJM model shows that the shape of the term structure at time $t+1$ if bond prices increase is given by:

$$B(t + 1, T, s(t+1, \omega)) =$$

$$\frac{B(t, T, s(t, \omega))}{B(t, t + 1, s(t, \omega))} \left[\cosh \left(\sum_{j=t+1}^{T-1} \sigma(t, j, s(t, \omega)) \sqrt{\Delta t} \right) \right]^{-1} \times \exp \left[\sum_{j=t+1}^{T-1} \sigma(t, j, s(t, \omega)) \sqrt{\Delta t} \right],$$

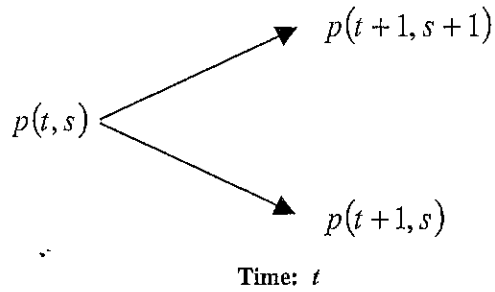
and the shape of the term structure at time $t+1$ if bond prices decrease is given by:

$$B(t+1, T, s(t+1, \omega)) =$$

$$\frac{B(t, T, s(t, \omega))}{B(t, t + 1, s(t, \omega))} \left[\cosh \left(\sum_{j=t+1}^{T-1} \sigma(t, j, s(t, \omega)) \sqrt{\Delta t} \right) \right]^{-1} \times \exp \left[\sum_{j=t+1}^{T-1} \sigma(t, j, s(t, \omega)) \sqrt{\Delta t} \right].$$

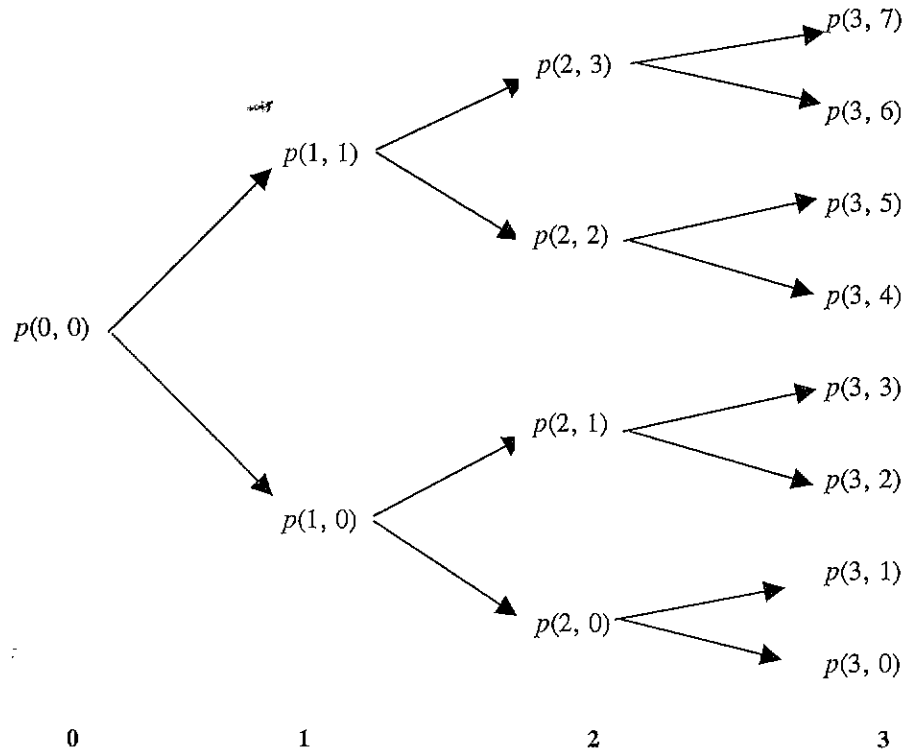
Since $\sigma(t, T, s)$ can be time and state dependent, the sum of the forward volatilities along a path consisting of an up move followed by a down move will, in general, be different than a down move followed by an up move, and the tree will not recombine⁷.

Figure 1
Notation Used for One-Period Binomial Model



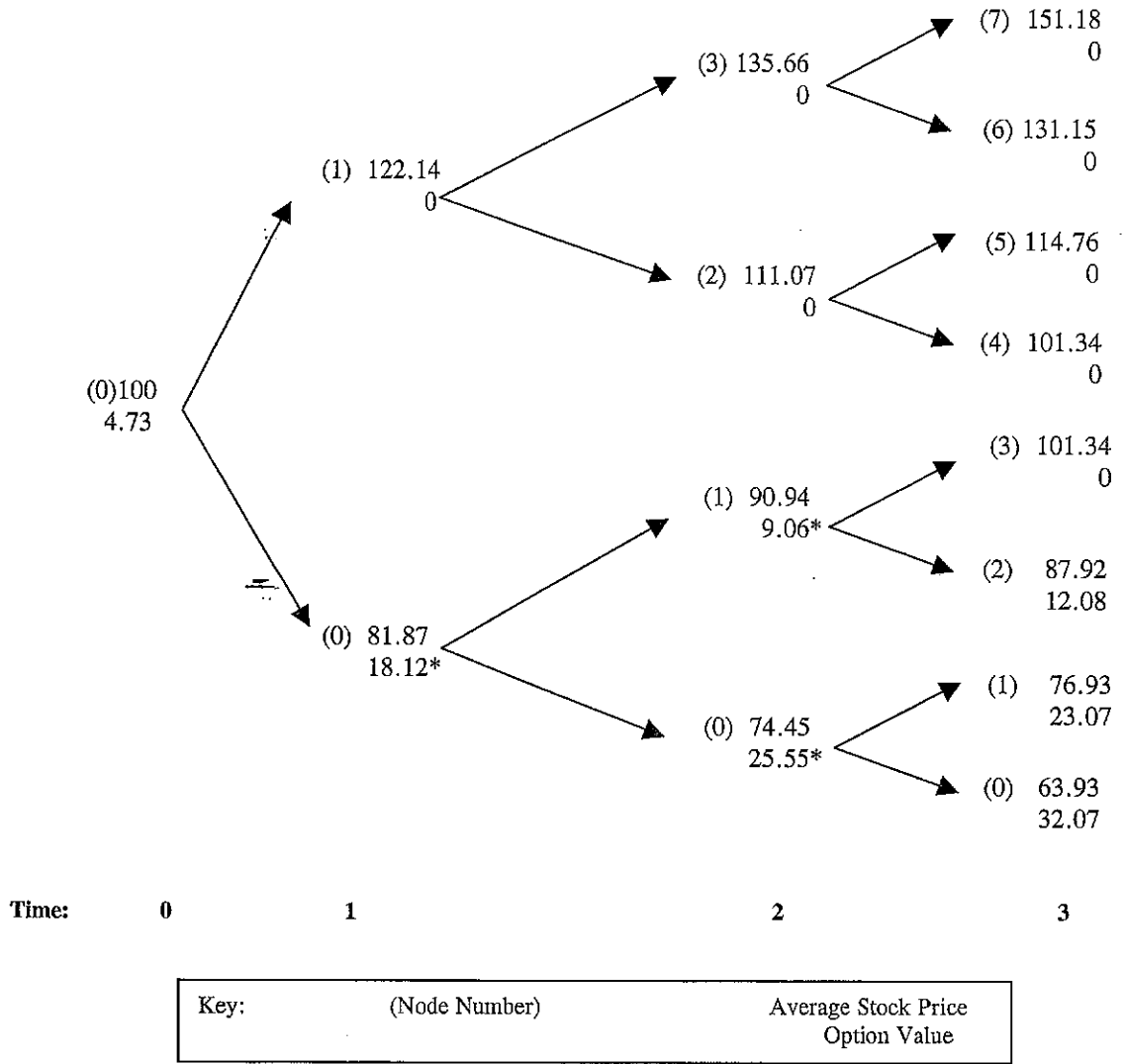
One period binomial model. $p(t, s)$ denotes the price of the state variable on which the option is written at time t and state s .

Figure 2
Three-Period Non-Recombining Binomial Tree



A three period non-recombining binomial tree. There are 2^t states at each time t . At each time t and state s , the value of the underlying state variable on which the option is written is denoted by $p(t, s)$. Time is labeled at the bottom of the figure.

Figure 3
Evolution of Arithmetic-Average Stock Price



This figure shows the evolution of the arithmetic average stock price for a three-period model. At each node, the number in parentheses is the node number, the number in the first row is the arithmetic average stock price as of that node, and the number in the bottom row is the value of the American Asian option at that node. A '*' indicates that early exercise of the option is optimal.

