Efficient Algorithms for Average-Rate Option Pricing

Tian-Shyr Dai
Department of Computer Science
and Information Engineering
National Taiwan University
Taipei, Taiwan
r6526008@csie.ntu.edu.tw

Yuh-Dauh Lyuu
Department of Computer Science
and Information Engineering
National Taiwan University
Taipei, Taiwan
lyuu@csie.ntu.edu.tw

Abstract

Average-rate options are sophisticated exotic derivatives whose payoff depends on the average value of the underlying asset. Pricing the geometric average-rate options by the lattice model and the combinatorial approach is presented in this paper. The lattice model can also price arithmetic average interest rate options under the Hull-White model. For the harder arithmetic average-rate equity option pricing, a sophisticated method for constructing the lattice is proposed. Comprehensive experimental results show that this novel approach offers more accurate results than existing methods.
Keywords: Asian-option, lattice, pricing, derivatives, algorithm.

1 Introduction

Average-rate options are options whose payoff depends on the average value of the underlying asset. These options are strongly path-dependent, which in turn makes the valuation problem difficult. In the over-the-counter market, many option contracts are written on such average options. An efficient and correct pricing approach is therefore needed and important.

If the value of the underlying asset follows the log-normal distribution, the geometric average of the asset value is also log-normally distributed. An analytic formula for valuing European-style geometric average-rate options is provided by Kemna and Vorst [7]. In our paper, the $O(n^4)$ algorithm the lattice method of Cho and Lee [1] is investigated for their convergence and the characteristics of early exercise. A much faster $O(n^3)$ combinatorial approach is then introduced for European-style options. As argued in [9], this type of algorithm is useful for pricing European-style geometric average-rate options with non-standard payoff functions.

Arithmetic average-rate interest rate options under normal interest rate models such as the Vasieck model [11] or the Hull-White model [6] can be easily solved by modifying the previous lattice method. Experimental results will be used to illustrate the convergence and early exercise behaviors.

When the underlying asset value follows the log-normal distribution, pricing arithmetic average-rate options is a well-known hard problem. To solve this problem, a sophisticated trinomial lattice method is proposed. Experimental data show good performance of the proposed method and superior convergence than existing methods.

2 Pricing Geometric Average-Rate Options

Assume the maturity equals the averaging periods. The geometric average $G(n)$ is defined as

$$G(n) = [S(0)S(1)S(2)\dots S(n-1)S(n)]^{\frac{1}{n+1}}$$

where $S(i)$ denotes the underlying asset value at time i and n is the number of periods. The payoff at maturity for the geometric average-rate call option is

$$\max(G(n) - X, 0)$$

where X is the strike price. The payoff at time i if the option holder exercises the option is defined as $G(i) - X$.

This section presents a lattice method and a combinatorial method for pricing geometric average-rate options. The combinatorial method is much faster and uses a novel generating function. The lattice method, on the other hand, can price American-style options.

2.1 The Lattice Method

Define u and d as the magnitudes for the upward and downward movements, $N(i, j)$ as the node on the lattice for which i is the time and j is the number of down movements needed to reach this node. Under the CRR model, $ud = 1$. The maximum and the minimum geometric sums for node $N(i,j)$ are

$$N_{\max}(i,j) \equiv S(0)^{i+1} u^{i(i+1)/2 - j(j+1)}$$

and

$$N_{\min}(i,j) \equiv S(0)^{i+1} u^{-i(i+1)/2 + (i-j)(i-j+1)}$$

respectively. The set of possible geometric sums at $N(i,j)$, is

$$G_{N(i,j)} \equiv \{A : A = S(0)^{i+1} u^k; k = \frac{-i(i+1)}{2}, \frac{-i(i+1)}{2} + 2, \frac{-i(i+1)}{2} + 4, \dots, \frac{i(i+1)}{2}; N_{\min}(i,j) \le A \le N_{\max}(i,j)\}. \quad (1)$$

Node $N(i,j)$ will keep these $|G_{N(i,j)}|$ $(=2j(i-j))$ states.

The option value for the state $S(0)^{i+1}u^k$ at $N(i,j)$ is

$$D \equiv (P_u \times V_{N(i+1,j)}(S(0)^{i+1}u^{k+i-2j+1}) + P_d \times V_{N(i+1,j+1)}(S(0)^{i+1}u^{k+i-2j-1})) \times B \quad (2)$$

where $V_n(S)$ represents the option value for state S at node n, B is the discount factor for that period, and P_u and P_d are the up and down probabilities, respectively. This backward induction implies an $O(n^4)$ algorithm because the total number of states of all the nodes can be seen to be $O(n^4)$. For American-style calls, say, the option value for that state becomes

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

2.2 The Combinatorial Method

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

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

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

in which the ith up move contributes u^{x_i} to the u^m term. It can be shown that,

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

The probability for each path is 2^{-n} in this model. So the option value is the present value of

$$\sum_{i=0}^{n(n+1)/2} 2^{-n} q(i) \max(S(0)M^{\frac{1}{n+1}} - X, 0)$$

for the call. Since the $q(i)$ can be computed in $O(n^3)$ time, pricing European-style options can be solved in time proportional to n^3.

2.3 Experimental Data

Assume the underlying asset value is 100, the strike price is equal to 100, the volatility is 20%, the risk-free rate is 10%, and the time from the issuing day to maturity is 1 year. The analytical value then is 6.769955. The experimental data are in Figure 1. Both approaches converge quickly for European-style options. Figure 2 compares the performance of these

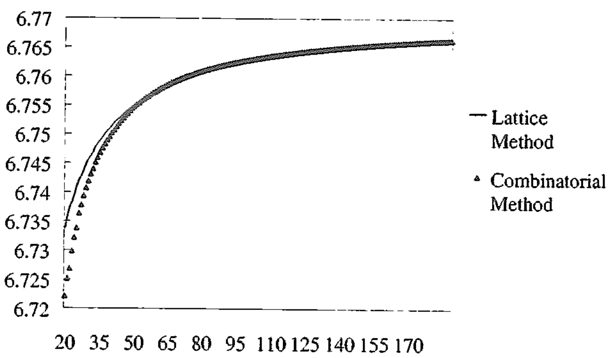


Figure 1: CONVERGENCE. The x-axis is the number of periods, and the y-axis is the price.

two methods. Clearly, the combinatorial method is much more efficient than the lattice approach.

Pricing American-style options using the lattice method is illustrated in Figure 3. The option value converges *monotonically* for n large enough. This should be contrasted the case of standard American-style options [10]. The early exercise behavior is illustrated in Figure 4. It implies the call may not be exercised merely because the underlying price is high.

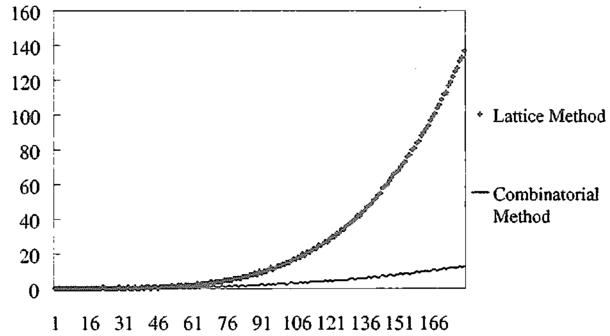


Figure 2: COMPUTATION TIME. The x-axis is the number of periods, and the y-axis is the computation time in seconds.

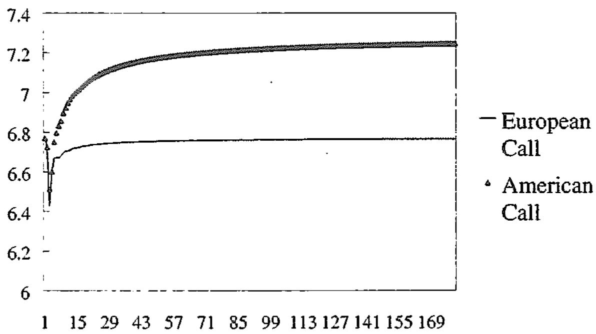


Figure 3: AMERICAN OPTION. The x-axis is the number of periods, and the y-axis is the option value.

3 Pricing Arithmetic Average-Rate Interest Rate Options

With the lattice method described above, we can price arithmetic average-rate interest rate options under the Hull-White model [5]. This is because the difference of the short rate between adjacent nodes at the same period is equal. In Figure 5, the difference of the short rate between adjacent nodes at the same time, ΔR, are all equal to 1.73. So the possible arithmetic sums for each node at time i must be from the set

$$A_i \equiv \{S + k\Delta R; \tfrac{-i(i+1)}{2} \leq k \leq \tfrac{i(i+1)}{2}\}$$

where S is the sum of the short rates from time 0 to time i. For example, $S = 9.02\%$ for node C in Figure 5.

The geometric average-rate options can be priced by modifying our lattice algorithm for their equity counterpart. The time complexity is $O(n^4)$ and memory usage is $O(n^3)$. In Figure 6, the parameters are: $a = 0.1$, $\sigma = 0.01$, the t-year continuous compounded zero coupon rate is $0.08 - 0.05e^{-0.18t}$,

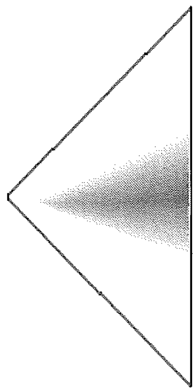
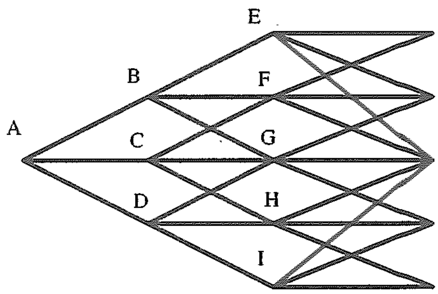


Figure 4: EARLY EXERCISE BEHAVIORS. The triangle denotes the lattice. The darker the point is, the more likely the option will be exercised there.



Node	A	B	C	D	E	F	G	H	I
r(%)	3.82	6.93	5.20	3.47	9.71	7.98	6.25	4.52	2.79
Pu	0.167	0.122	0.167	0.222	0.887	0.122	0.167	0.222	0.087
Pm	0.666	0.656	0.666	0.656	0.026	0.656	0.666	0.656	0.026
Pd	0.167	0.167	0.122	0.122	0.087	0.222	0.167	0.122	0.887

Figure 5: HULL-WHITE INTEREST RATE LATTICE MODEL.

and the time to maturity is one year. (See [5] for the definitions of these terms.) The payoff at maturity for this call option is defined as

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

where X is the strike value, and $A(i)$ is the arithmetic average of the short rate time 0 to time i. The payoff at time j if the option holder exercises the option is $100 \times (A(j) - X)$.

The convergence is monotonic but slow. The computation time also grows dramatically when n is large. It took almost nine minutes on a Pentium-Pro processor given a 150-period lattice. The early exercise behavior for the arithmetic average-rate interest-rate option is shown in Figure 7, which is different from Figure 4.

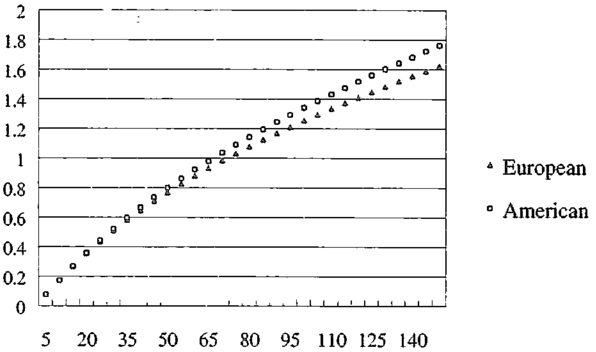


Figure 6: EUROPEAN AND AMERICAN INTEREST RATE AVERAGE-RATE OPTIONS UNDER THE HULL-WHITE MODEL.

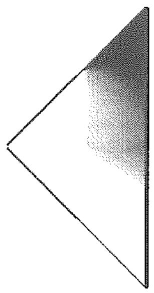


Figure 7: THE EARLY EXERCISE BEHAVIOR. The triangle denotes the lattice. The darker the point is, the more likely the option will be exercised there.

4 Pricing Arithmetic Average-Rate Options

Pricing arithmetic average-rate options when the underlying asset value follows the log-normal distribution is a well-known hard problem. There are at least two problems with most existing approaches. One is that they may not be used for pricing American-style options, and the other is that most approaches fail to get acceptable results under some cases [2].

The novel lattice method by Hull and White [4] is one approach that can be used to solve American-style options. But it seems that this approach fails on some data. In Figure 8 we replicate the algorithm that Hull described in [3].[1] The initial stock price is 50, the risk-free rate is 10% per year, the volatility is 0.3, and the life of the option is 0.5 year. We found that the series increases monotonically, but it's not clear if it converges when n is large.

The original Hull and White's algorithm in [4] using a different interpolation scheme (The parameter h, following the suggestions in [4], is defined as 0.005

[1]The interpolated values between the maximum and the minimum arithmetic sums at each node are equally spaced.

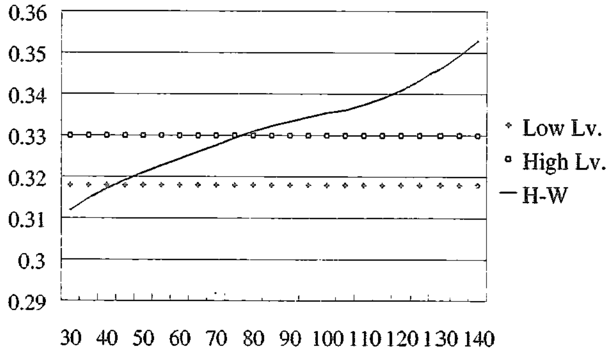


Figure 8: HULL AND WHITE'S METHOD USING LINEAR INTERPOLATION. The interval between *Low Lv.* and *High Lv.* is the 95% confidential interval that we get from [1]. The x-axis represents the number of periods of the lattice model.

in our experiments) also seems to perform poorly at some extreme cases. See Figure 9. It seems that

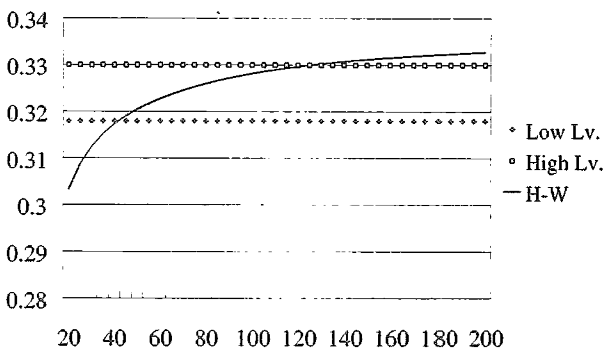


Figure 9: HULL AND WHITE'S METHOD USING EXPONENTIAL INTERPOLATION SCHEMES Parameters are the same as in Figure 8.

the series will converge when n becomes large, but the value is out of the range of the 95% confidential interval.

4.1 A New Lattice Model

Consider a lattice with the following property: the underlying asset value for each node is a positive integer.[2] Since the natural numbers are closed under addition, the possible arithmetic sums for any node $N(i,j)$ must also be positive integers. Assume the maximum arithmetic sum from time 0 to i for node $N(i,j)$ is $N_{\max}(i,j)$ and the minimum arithmetic

[2]The initial value of the underlying assets is set to be an integer for ease of discussion. This algorithm still works if this is not the case.

sum is $N_{\min}(i,j)$. The set of all possible arithmetic sums at $N(i,j)$ is

$$N_{sum}(i,j) \equiv \{A : A \in \aleph, N_{\min}(i,j) \leq A \leq N_{\max}(i,j)\}.$$

Obviously, $|N_{sum}(i,j)|$ is a finite number, which implies that we can solve the problem efficiently if there is a way to build the lattice such that the growth rate of $|N_{sum}(i,j)|$ is acceptable.

Out algorithm is described here. The lattice is trinomial. Redefine $N(i,j)$ as the node that has the j-th biggest value at time i. The symbols $\mu(s), \nu(s), \omega(s)$ denote the up, flat ,and down branches from node s,

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

Define $\Delta t = T/n$. $V(N)$ is the underlying asset value at node N, $M(N,\Delta t)$ and $Var(N,\Delta t)$ are the mean and variance at the next time if the current state is N, respectively, and $P_u(N)$, $P_m(N)$, $P_d(N)$ denote the up, flat, and down probabilities from node N. See Figure 10. Since we will calibrate the first and second moments of the underlying asset value, the variables at node N satisfy the following equations:

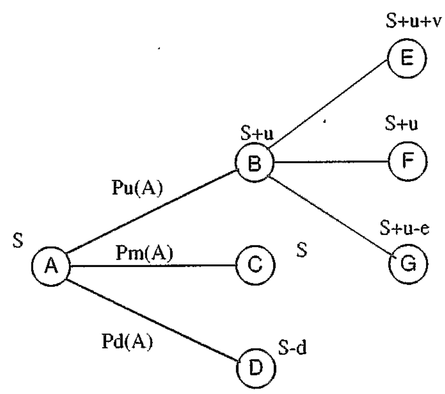


Figure 10: A NEW TRINOMIAL MODEL

$$M(N,\Delta t) = V(\mu(N)) \times P_u(N) + V(\nu(N)) \times P_m(N) + V(\omega(N)) \times P_d(N), \quad (3)$$

$$Var(N,\Delta t) = (V(\mu(N)) - M(N,\Delta t))^2 \times P_u(N) + (V(\nu(N)) - M(N,\Delta t))^2 \times P_m(N) + (V(\omega(N)) - M(N,\Delta t))^2 \times P_d(N), \quad (4)$$

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

Note that $V(N) - V(\mu(N))$ and $V(N) - V(\omega(N))$ are integers. Impose $V(N) = V(\nu(N))$ for the flat branch. Since $P_u(N)$, $P_m(N)$, and $P_d(N)$ are probabilities, the following inequalities must hold,

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

So the problem is, how to select proper up and down displacements to fit the conditions above? This can be done by specifying

$$\lceil \sqrt{Var} \rceil \quad (5)$$

for both the up and the down displacement. We finally impose the condition that the tree combines.

For example, assume the initial node is A in Figure 10. Then u and d can be solved by applying (5) and imposing $u = d$. Note that u and d here are additive instead of multiplicative. For the nodes whose underlying asset value is larger then S, like node B, the equation $e = u$ must hold to make the $C - G$ branch flat. The value of v can then be determined by applying (5) again .[3] For the nodes whose underlying asset value is smaller than S, similar steps must be taken.

A problem with this method is that the magnitudes of the upward and downward displacements may not be natural numbers when the variance of the underlying asset is small. This is because no solution can satisfy all the constraints then. One more idea is needed. Define A_j as

$$A_j \equiv \{\ldots, -\frac{2}{2^j}, -\frac{1}{2^j}, 0, \frac{1}{2^j}, \frac{1}{2^j} \cdots\}.$$

if j is a positive integer, then A_j is closed under addition. So the problem can be solved by setting the upward displacement for node N, call it u_N, and the downward displacement, call it d_N. In such a way that if N, u_N, $d_N \in A_j$, then all possible arithmetic sums under the algorithm will also be in A_j.

4.2 Reducing the Memory Requirements

In order to save computer memory, the lattice should be divided into partitions. Refer to Figure 11 in the following. Assume nodes A and B are on line $L3$. Lines $L1$ and $L2$ are composed of the nodes that can be reached by taking the up-moves from A and B, respectively. Define $Sa(N)$ to be the function that returns the smallest integer j such that $u_N, d_N \in A_j$, where N is a node. Let $F(N) \equiv \max(0, Sa(N))$. Assume for each node α,

$$V(\alpha) \leq V(B) \Rightarrow F(\alpha) = 2$$
$$V(B) < V(\alpha) \leq V(A) \Rightarrow F(\alpha) = 1$$
$$V(\alpha) > V(A) \Rightarrow Sa(\alpha) = 0$$

[3] We need modify the up and down displacements slightly if one of the probabilities becomes negative.

Table 1: **Compare the Required working Space.**

	Before Reduction		After Reduction			
Periods	100	160	100	135	150	160
Required Space	16,106,074	118,524,029	2,969,062	9,065,895	14,030,903	18,280,584

"Before Reducing" means the technique described in this section is not applied, whereas "After Reducing" means the technique is used. The required space denote the magnitude of array. (Eight bytes are needed to store an element in a array for C programs.)

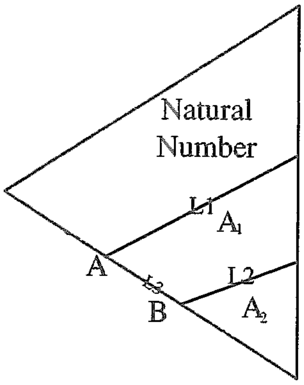


Figure 11: THE CLOSURE PROPERTY.

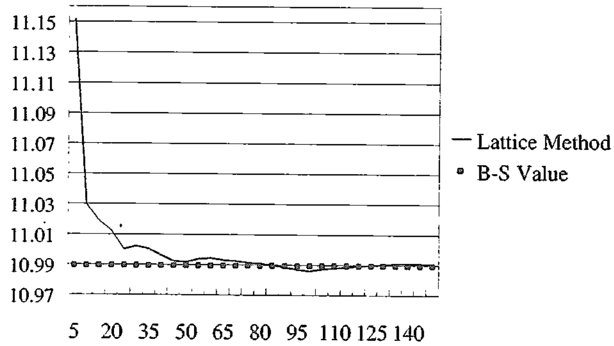


Figure 12: NEW LATTICE FOR STANDARD CALL OPTIONS. The initial stock value is 100, the strike price is equal to 100, the volatility is 0.2, the risk-free rate is 0.06, and the time to maturity is one year. The benchmark value derived from the Black-Scholes formula is 10.989547.

Then all possible arithmetic sums for the nodes in the lower part of the lattice are closed in A_2, all possible arithmetic sums for the nodes in the central part of the lattice are closed in A_1, and all possible arithmetic sums for the rest of the nodes are closed for natural number. Applying this technique can help save the working space dramatically. A numerical calculation shows why this is necessary. Assume the value of the underlying asset is 100, the strike price is also 100, the volatility is 0.2, the risk-free rate is 0.1, and the time to the maturity is one year. The required space computed by the programs are listed in Table 1. The saving is significant. Notice that the required working space still grows dramatically when the number of periods becomes large. So this algorithm may be inappropriate for cases that convergence is slow. Finally, we need a recurrence formula similar to (2) to compute the option value for each state.

4.3 Experimental Results

First, we price a standard option with our method to check if it works in the simplest case. Figure 12 shows that the performance is excellent. The convergence is quick, implying that the lattice model approximates the distribution of the underlying asset value well.

We next test the settings identical to the ones in Figure 8 and 9 (see Figure 13). It can be seen that our new method converges extremely well. This method does not over-price the options much, which

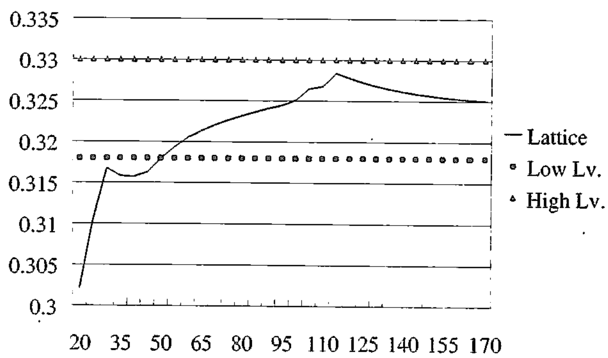


Figure 13: OUR LATTICE METHOD FOR ARITHMETIC AVERAGE-RATE OPTIONS. The assumptions are the same as Figure 8.

Table 2: **The Value of Arithmetic Average-Rate Options Derivated by Various Algorithms.**

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

The initial underlying asset value is 50; the risk free rate is 10% per year; the volatility is 0.3 per year; averaging is between the beginning of the life of the options to maturity. H-W denotes the Hull and White algorithm based on 40 time steps and $h = 0.005$. Monte Carlo simulations are based on 40 time steps and 100,000 trials. A.(30) is our lattice method with the number of periods equal to 30. Levy denotes Levy's approach described in [8].

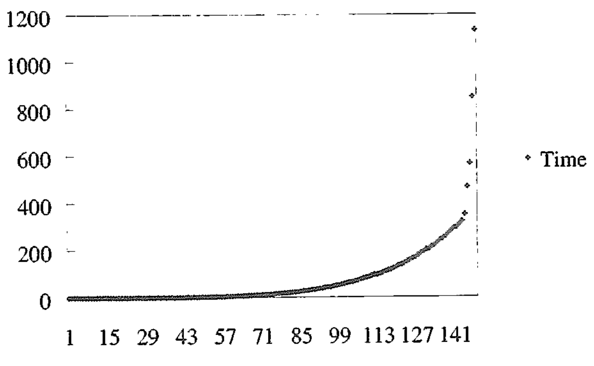


Figure 14: COMPUTATION TIME USED BY THE NEW METHOD. The numbers are in seconds.

happens to the Hull-White method when the the number of periods is large. The computation time however grows dramatically when the number of periods is beyond a certain threshold. See Figure 14 for a plot. Fortunately, it converges much earlier.

Some numerical results are illustrated in Table 2. The numerical data about the Hull-White method and Monte Carlo simulations are taken from [1]. The number of periods used for our new method is 30. It took about 2 seconds on a Pentium-Pro 300MHz computer. Most of the values computed by our algorithm are close to the value computed by Monte Carlo simulation. Only two value are out of the range of 95% confidence interval. They are marked with an "*".

Another set of experiments is focused on extreme cases mentioned in [2]. In that paper, the authors compare many proposed algorithms and conclude that some algorithms may fail in extreme cases. We test their extreme cases in Table 3 and show that our lattice model performs well in each one of them.

5 Conclusions

This paper develops several methods for pricing average-rate options. The combinatorial method was founded to improve upon the computational speed of the standard lattice method. Such a method has been found useful in single and double barrier options pricing [9]. Combining the idea with the interest rate model of Hull and White, we can price arithmetic average-rate interest rate options for normal models.

Table 3: **Testing the Lattice Model under Some Extreme Cases**

r	σ	T	$S(0)$	GE	$Shaw$	$Euler$	PW	TW	$MC10$	$MC100$	$S.E.$	$A.(30)$
0.05	0.5	1	1.9	0.195	0.193	0.194	0.194	0.195	0.192	0.196	0.004	0.193
0.05	0.5	1	2.0	0.248	0.246	0.247	0.247	0.250	0.245	0.249	0.004	0.246
0.05	0.5	1	2.1	0.308	0.306	0.307	0.307	0.311	0.305	0.309	0.005	0.306
0.02	0.1	1	2.0	0.058	0.520	0.056	.0624	.0568	.0559	.0565	.0008	0.0558
0.18	0.3	1	2.0	0.227	0.217	0.219	0.219	0.220	0.219	0.220	0.003	0.219
0.125	0.25	2	2.0	0.172	0.172	0.172	0.172	0.173	0.173	0.172	0.003	0.172
0.05	0.5	2	2.0	0.351	0.350	0.352	0.352	0.359	0.351	0.348	0.007	0.351

The exercise price is 2.0, r is the risk-free rate, T is the life of the options from the issuing day to maturity, σ is the volatility, $S(0)$ is the initial price of the underlying asset, and $A.(30)$ denotes our method. The other approximation methods for comparison are: Geman-Eydeland (GE), Shaw, Euler, Post-Widder(PW) and Turnbull-Wakeman (TW). The benchmark values ($MC10$ and $MC100$) and the approximation values are copied from [2]. S.E. is the standard error, also from [2].

A new lattice method is designed for solving arithmetic average-rate options. We use the closure property of natural numbers to reduce the number of states needed to an acceptable level. Experiments show that this new method compares favorably with the Hull-White method and a host of other methods.

Acknowledgement

Special thanks go to Ming-Chen Sun for giving us the results on the Hull and White method, and Guang-Shieng Huang for checking the mathematical statements in this paper.

References

[1] Choa, H. Y. and H. Y. Lee, "A Lattice Model for Pricing Geometric and Arithmetic Average Options," The Journal of Financial Engineering, Vol. 6, No. 3, pp. 179–191, Sep. 1997.

[2] Fu, M. C., D. B. Madan, and T. Wang, "Pricing Continuous Asian Options: a Comparsion of Monte Carlo and Laplace Transform Inversion Methods," Journal of Computational Finance, Vol. 2, No. 2, pp. 49–74, Winter 1998.

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

[4] Hull, J. and A. White, "Efficient Procedures for Valuing European and American Path-Dependent Options," Journal of Derivatives, Vol. 1, No. 1, pp. 21–31, Fall 1993.

[5] Hull, J. and A. White, "Numerical Procedures for Implementing Term Structure Models I: Single Factor Models," Journal of Derivatives, Vol. 2, No. 1, pp. 7–16, Fall 1994.

[6] Hull, J. and A. White, "One-Factor Interest Rate Models and the Valuation of Interest Rate Derivatives Securities," Journal of Financial and Quantitative Analysis, Vol. 28, No. 2, pp. 235–254, Jun. 1993.

[7] Kemna, A. and A. Vorst, "A Pricing Method Based on Average Asset Values," Journal of Banking and Finance, Vol. 14, No. 1, pp. 113–129, Mar. 1990.

[8] Levy, E., "Pricing European Average Rate Currency Options", Journal of International Money and Finance, Vol. 11, No. 5, pp. 474–491. Oct. 1992.

[9] Lyuu, Y.-D. "Very Fast Algorithms for Barrier Option Pricing and the Ballot Problem." The Journal of Derivatives, Vol. 5, No. 3, pp. 68–79, Spring 1998.

[10] Omberg, E., "A Note on the Convergence of Binomial-Pricing and Compound-Option Models," Journal of Finance, Vol. 42, No. 2, pp. 463–469, Jun. 1987.

[11] Vasicek, O. "An Equilbrium Characterization of the Term Structure," Journal of Financial Economics, Vol. 5, No. 2, pp. 177–188, Aug. 1977.