Acta Informatica (2007) 44:23–39
DOI 10.1007/s00236-006-0033-9

ORIGINAL ARTICLE

An exact subexponential-time lattice algorithm for Asian options

Tian-Shyr Dai · Yuh-Dauh Lyuu

Received: 3 January 2006 / Accepted: 18 November 2006 /
Published online: 3 January 2007

Abstract Asian options are popular financial derivative securities. Unfortunately, no exact pricing formulas exist for their price under continuous-time models. Asian options can also be priced on the lattice, which is a discretized version of the continuous-time model. But only exponential-time algorithms exist if the options are priced on the lattice without approximations. Although efficient approximation methods are available, they lack accuracy guarantees in general. This paper proposes a novel lattice structure for pricing Asian options. The resulting pricing algorithm is exact (i.e., without approximations), converges to the value under the continuous-time model, and runs in subexponential time. This is the first exact, convergent lattice algorithm to break the long-standing exponential-time barrier.

1 Introduction

Derivative securities are financial instruments whose value depends on some underlying assets. They are essential to speculation and the management of financial risk. The underlying assets can be as diverse as stocks, indexes, currencies, commodities,

An early version of this paper appeared in the Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, 2004.

T.-S. Dai was supported in part by NSC grant 94-2213-E-033-024.
Y.-D. Lyuu was supported in part by NSC grant 94-2213-E-002-088.

T.-S. Dai
Department of Information and Finance Management, National Chiao-Tung University,
1001 Ta Hsueh Road, Hsinchu, Taiwan 300, Republic of China
e-mail: d88006@csie.ntu.edu.tw

Y.-D. Lyuu (✉)
Department of Computer Science and Information Engineering and Department of Finance,
National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan
e-mail: lyuu@csie.ntu.edu.tw

bonds, mortgages, other derivatives, temperatures, or countless others [20]. The most common derivatives are futures, swaps, and options. Futures contracts oblige the owners to buy certain assets for a fixed contractual price at some point in the future. Swaps let parties exchange assets in the future according to a rule. The focus of the paper is the third type of derivatives: options. Options are derivative securities that give their buyer the right, but not the obligation, to buy or sell the underlying assets for a contractual price called the exercise price. Take the standard stock option for example. Suppose an investor buys a call option which gives him the right to buy 100 shares of XYZ stock at $10 per share 60 days from now. If the stock price finishes above $10 then, say $25, then the buyer realizes a profit of $100 \times (25 - 10) = 1,500$ dollars by exercising the option. If the stock price finishes below $10, the buyer simply gives up the option and receives nothing. The payoff of this call option is therefore $100 \times \max(S - 10, 0)$, where S is the stock price 60 days from now and is a random variable.

In practice, options more complex than the standard ones have been structured to meet specific financial needs. Many of them are path-dependent. A path-dependent option is an option whose payoff depends strongly on the price history of the underlying asset, which we will assume to be stock throughout the paper for convenience. The payoff function of a path-dependent option may depend on, for example, the maximum stock price, the minimum stock price, or the average stock price. It may also depend on whether the stock price ever hits a given target price or whether the stock price ever stays below a given target price for a given length of time [31]. The possibilities are limited only by imagination and marketability.

As an option conveys a right, it must command a positive price called premium. How to assign a fair price to an option given a continuous-time stochastic process for the stock price has been investigated since 1900 [27]. Finally in 1973, Black and Scholes [4] settle the option pricing problem in a way that is considered intellectually satisfactory. Although an option must have a unique theoretical price, calculating that price may be computationally difficult. For example, Chalasani et al. [9] show that a path-dependent option can be so defined that its pricing problem is #P-hard. Even though no traded options have been proved to be that hard to price, provably convergent polynomial-time solutions for some remain elusive.

The focus of this paper is a particular type of path-dependent option called the Asian option, which is known to be difficult to price. Asian options seem to be suggested first by Ingersoll [22] and were originally traded in Asian markets, particularly Tokyo [39]. The payoff of an Asian option depends on the average stock price. It is hence useful for hedging transactions whose cost is related to the average price of the underlying asset (such as crude oil). Its price is also less subject to price manipulation. As a result, the average-price feature is popular in many thinly-traded markets and embedded in quite a few derivatives like convertible bonds.

The Asian option can be further categorized into fixed-strike Asian options and floating-strike Asian options. The payoff function of a fixed-strike Asian option is the maximum of the difference between the average stock price and the exercise price and zero. The payoff of a floating-strike Asian option, in contrast, is defined as the maximum of the difference between the stock price and the average stock price and zero.

There are no simple, exact closed-form solutions to the Asian option's price under the standard continuous-time model. We call this price the true option value for simplicity. Many approximate closed-form solutions have been proposed under various assumptions [25,29,35]. Geman and Yor [17] derive an analytical expression for the Laplace transform of the Asian call. Numerical inversion of this transform is considered in [16,34]. Some inversion algorithms based on the Euler and Post-Widder methods can be found in [1]. Rogers and Shi [33] provide lower and upper bounds. Zhang [38] gives a semi-analytical method that is very accurate for the options tested. However, approximate closed-form formulas lack accuracy guarantees. Indeed, some can produce large pricing errors under certain settings [15,39]. Furthermore, the analytical approach does not apply to American-style Asian options. (Definitions will be given later. Basically, European-style options do not allow early exercise, whereas American-style options do. Early exercise results in early payoff and is related to the so-called optimal stopping problem.)

As no simple exact closed-form formulas exist yet for the Asian option, the development of efficient numerical algorithms becomes critical. First, there are the popular Monte Carlo and quasi-Monte Carlo methods [5–7,23]. But they all suffer from the inability to price American-style Asian options without bias. Recently, a least-square Monte Carlo approach to address this issue has been proposed [26]. But the analysis of the method is very intricate. Other problems with the Monte Carlo method are that it is inefficient and the result is probabilistic.

The true option value we are seeking must satisfy a certain partial differential equation (PDE) with proper boundary conditions. Hence a popular numerical approach is the lattice and the related discretized PDE method. This general approach can handle early exercise. A lattice consists of nodes and edges connecting them. It divides the time interval during which the option is alive into n equal discrete time steps and sets up a discrete-time random walk for the stock price. It is known that if the lattice converges to the continuous-time stock price process as n goes to infinity, the option value given by the lattice via backward induction (to be introduced later) also converges to the true option value [13]. The PDE can be solved numerically by the finite difference method on a grid structure with a proper specification of the boundary conditions. Compared to the PDE method, the lattice method contains more financial intuitions. Numerically, the lattice method can be viewed as an explicit method of solving the PDE [27]. The key remaining issue is whether convergence can be achieved efficiently.

For the Asian option, only exponential-time lattice pricing algorithms are known if approximation is not used during backward induction. (In this paper as well as in the literature, all running times are in terms of n.) We call a lattice algorithm exact if it does not adopt approximation during backward induction. The option value computed by an exact lattice pricing algorithm will be referred to as the desired option value, which converges to the true option value at a rate of $O(n^{-1})$ [13].

To see intuitively why pricing on the lattice can be time-consuming, let us assume a trinomial random walk for the stock price. Hence three prices follow each stock price. Recall that the payoff of the Asian option depends on the average stock price. This implies that after n time steps, the price history contains 3^n possible price paths, each with a different average price. Worse, every one of these average prices is involved in determining the option price. For simpler options, we may be able to take advantage of particular structures in the random walk and the payoff function to lower the number of paths to a polynomial in n. Unfortunately, for Asian options, the exponential count has not been dramatically lowered before this paper.

To solve the inefficiency problem of the exact pricing algorithm, Hull and White [21] propose approximation algorithms. Their influential paradigm has been followed by most lattice algorithms [3,24,32,40]. The major problem with the Hull–White paradigm is convergence. It is known that an improper use of approximations can result in divergence or convergence to the wrong value [14]. Chalasani et al. [8] provide an $O(n^4)$ approximation algorithm that brackets the desired option value. But the difference between the upper bound and the lower bound of the option value may not converge to 0. Aingworth et al. [2], Chalasani et al. [9], Dai et al. [11], and Ohta et al. [30] provide polynomial-time approximation algorithms that have convergence guarantees for European-style Asian options. Večeř [36] suggests an $O(n^2)$-time algorithm for European-style Asian options and American-style floating-strike Asian options, which is the best in the literature. But his technique can not handle American-style fixed-strike Asian options. For American-style fixed-strike Asian options, all polynomial-time algorithms keep running averages as states and use interpolation in the approximation. Whether they converge is hard to analyze.

An alternative paradigm due to Dai and Lyuu [12] constructs a trinomial lattice called the multiresolution lattice. The multiresolution lattice converges to the true option value because it is exact. Numerical results show that the method can work for n up to 160. Note that an exponential-time algorithm cannot possibly work with an n this large because $3^{160} \sim 2.185 \times 10^{76}$. Hence it has been conjectured that the multiresolution lattice gives rise to a subexponential-time algorithm. There are two drawbacks with the multiresolution lattice. First, no proof is available to show that it indeed runs in subexponential time. Second, its construction relies partially on an ad hoc local search, and no proof exists to guarantee that the search must succeed.

This paper will propose the first exact lattice pricing algorithm that provably breaks the exponential-time barrier (see Theorem 1 in Sect. 4). The running time of our algorithm is $2^{O(\sqrt{n})}$. Our algorithm has convergence guarantees for both European-style and American-style Asian options, whereas no other polynomial-time algorithms have such guarantees for all American-style Asian options. Here is an overview of the major steps of our algorithm. Since the payoff of an Asian option is determined by the average stock price, the pricing algorithm is faster if the number of possible average stock prices is fewer. Towards the goal of reducing the number of average stock prices from 3^n to something subexponential in n, we construct a lattice composed of *integer* stock prices. The stock price sum of a price path on the lattice is therefore an integer. This critical property results in the reduction of the number of possible average stock prices. Exact backward induction can then be applied with a complexity that is only subexponential in n. We remark that our algorithm can price partial-averaging Asian options, whose payoff depends on the average stock price over only a part of the option's life, again in subexponential time. Note that the proposed algorithms contain no approximations other than the discretization of the continuous-time model; in other words, they are exact. Thus they converge to the true option values at a rate of $O(n^{-1})$ [13,14].

The homogeneous property of the option value is crucial to the construction of the said lattice. Suppose we multiply the stock prices by a constant K before pricing the option. The homogeneous property says that the resulting option value divided by K gives the original desired option value [28]. This property holds for efficient markets and essentially says the price of K merchandises is K times that of one. This detour increases the dynamic range of the stock prices so that proper integer stock prices can always be found. In our paper, a K of reasonable magnitude is found that guarantees the existence of an integer lattice. To ensure that the pricing results based on our lattice converge to the true option value, we have to make sure that our lattice converges to the continuous-time stock price process. For that, the lattice must match the mean and the variance of the continuous-time stock price process at each time step [13]. This paper will prove that an integer lattice that satisfies above constraints can be efficiently constructed.

We comment that although a subexponential-time algorithm is still far from being polynomial-time, it is much more practical than an exponential-time algorithm. For example, the multiresolution lattice algorithm, which probably runs in subexponential time, is capable of calculating prices for n that are beyond any exponential-time algorithms. As another example, the typical reason cited for the superiority of a cryptosystem based on elliptic curves is that currently the best algorithms to break it have exponential running times, whereas subexponential-time algorithms are known for the integer factorization problem [18].

The paper is organized as follows. The mathematical model is described in Sect. 2. We will review how a typical lattice is constructed and how the Asian option is priced on the lattice in Sect. 3. In Sect. 4, we will show how to construct the proposed lattice. Proofs in Sect. 5 show that our pricing algorithm is the first exact pricing algorithm that breaks the exponential-time barrier. Sample running times are given in Sect. 6 to show the performance of our algorithm. Section 7 concludes this paper.

2 Modeling and definitions

Assume that the Asian option initiates at 0 (in years) and matures at T (in years). Define $S(t)$ as the price of the stock at year t. The stock price is assumed to follow the continuous-time lognormal diffusion process:

$$S(t+dt) = S(t)\,\mathrm{e}^{(r-0.5\sigma^2)\,dt+\sigma\,dW_t}, \tag{1}$$

where W_t is the standard Wiener process, r is the risk-free interest rate per annum, and σ is the annual volatility. It is useful to think of dW_t as normally distributed with mean 0 and variance dt.

The discrete-time approximation to the lognormal diffusion partitions the time between year 0 and year T into n time steps. The length of each time step Δt thus equals T/n. Let S_i denote the stock price at (discrete) time i, which corresponds to $S(i\Delta t)$ in the continuous-time model. The payoff of the Asian option depends on the average stock price. Define

$$A_{\text{avg}}(i) \equiv \frac{S_0+S_1+\cdots+S_i}{i+1}.$$

A European-style option can only be exercised at the maturity date. The payoff to exercise a European-style fixed-strike Asian option at the maturity date is

$$\text{exercise value} = \begin{cases} A_{\text{avg}}(n) - X & \text{for call,} \\ X - A_{\text{avg}}(n) & \text{for put,} \end{cases} \tag{2}$$

where X is the exercise price. On the other hand, the payoff to exercise a European-style floating-strike Asian option at the maturity date is

$$\text{exercise value} = \begin{cases} S_n - A_{\text{avg}}(n) & \text{for call,} \\ A_{\text{avg}}(n) - S_n & \text{for put,} \end{cases} \tag{3}$$

In either case, the holder can give up the option at the maturity date if the exercise value is negative. Thus the final payoff of the European-style Asian option is max(exercise value, 0). A fundamental theorem in finance says that the option value at year 0 equals the discounted expected payoff at year T [19], or

$$\mathrm{e}^{-rT}\mathrm{E}\left[\,\max(\text{exercise value}, 0)\,\right].$$

We shall concentrate on calls as puts can be treated similarly. Our task is therefore to compute $\mathrm{e}^{-rT}\mathrm{E}[\,\max(A_{\mathrm{avg}}(n) - X, 0)\,]$ so that the value converges to the true option value $\mathrm{e}^{-rT}\mathrm{E}[\,\max(\frac{1}{T}\int_0^T S(t)\,\mathrm{d}t - X, 0)\,]$ as n increases.

The American-style option gives the owner the right to exercise the option before maturity, thus an earlier payoff. The exercise value for a fixed-strike and a floating-strike American-style call option at time i is

$$\text{exercise value} = A_{\mathrm{avg}}(i) - X, \tag{4}$$

and

$$\text{exercise value} = S_i - A_{\mathrm{avg}}(i), \tag{5}$$

respectively. An option will be exercised early by the owner if the option's continuation value (the value if the option is held) is smaller than its exercise value.

3 Preliminaries

Before introducing our new lattice, we first review the principle behind building a lattice that converges to the lognormal diffusion process. Next we will show how to price the Asian option on a lattice. We will also see why the traditional exact pricing algorithm explodes exponentially and why the approximation approach of Hull and White [21] is problematic. Finally, we will summarize the ideas of the integer lattice that will be useful later.

3.1 How to construct a lattice

We use the well-known Cox–Ross–Rubinstein (CRR) binomial lattice [9] to illustrate how a convergent lattice is constructed in principle. A three-time-step CRR binomial lattice is illustrated in Fig. 1. At each time step, the stock price S can either become Su—the up move—with probability P_u or Sd—the down move—with probability $P_d \equiv 1 - P_u$. The relation

$$ud = 1 \tag{6}$$

is enforced by the CRR binomial lattice. The logarithmic stock price's mean (μ) and variance (Var) one time step from now are derived from Eq. (1) as

$$\mu \equiv (r - 0.5\sigma^2)\,\Delta t, \tag{7}$$

$$\mathrm{Var} \equiv \sigma^2 \Delta t. \tag{8}$$

To make sure that the lattice converges to the continuous-time lognormal diffusion process, μ and Var should match the mean and variance of the lattice at least asymptotically:

$$P_u \ln u + P_d \ln d = \mu, \tag{9}$$

$$P_u(\ln u - \mu)^2 + P_d(\ln d - \mu)^2 = \mathrm{Var}. \tag{10}$$

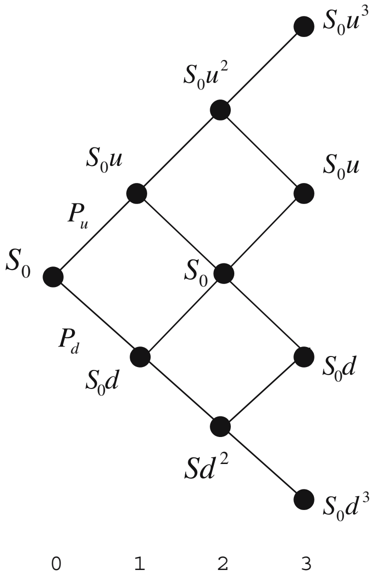


Fig. 1 A three-time-step CRR binomial lattice. The initial stock price is S_0. The upward and the downward multiplicative factors for the price process are u and d, respectively. The upward and the downward branching probabilities are P_u and P_d, respectively

In addition,

$$P_u + P_d = 1 \tag{11}$$

must hold. The four parameters P_u, P_d, u, and d are uniquely obtained by solving Eqs. (6) and (9)–(11). The branching probabilities P_u and P_d must also lie between 0 and 1. These requirements can always be met by suitably increasing n [27].

In general, if each node in a lattice branches to ℓ nodes at the next time step, we call the lattice ℓ-nomial. Ideas similar to the above can be applied to construct an ℓ-nomial lattice. Note that 2ℓ degrees of freedom are provided by an ℓ-nomial lattice. They include ℓ price multiplicative factors (like the u and d in the CRR binomial lattice) and ℓ branching probabilities (like the P_u and P_d in the CRR binomial lattice). These branching probabilities must also lie between 0 and 1. We therefore need 2ℓ independent equations to determine these 2ℓ variables uniquely. The matching of the mean and variance gives two equations. The branching probabilities sum to 1, giving another one. Additional $2\ell - 3$ equations must be added. For example, Eq. (6) is the extra equation used in the CRR binomial model. This paper will construct trinomial lattices (i.e., $\ell = 3$).

3.2 Pricing Asian options on a trinomial lattice

In a trinomial lattice, each node can branch to three successor nodes in the next time step. Let $S_{i,j}$ denote the $(j+1)$th largest stock price of the nodes at time i. A two-time-step trinomial lattice is illustrated in Fig. 2. Take the root node as an example. Its stock price is $S_{0,0}$. The stock price can move upward to $S_{1,0}$ with probability P_u, move flatly to $S_{1,1}$ with probability P_m, and move downward to $S_{1,2}$ with probability P_d. The branching probabilities will vary for different nodes. However, we drop the dependence on i and j from P_u, P_m, P_d for brevity. In general, there are $2i + 1$ nodes at time i.

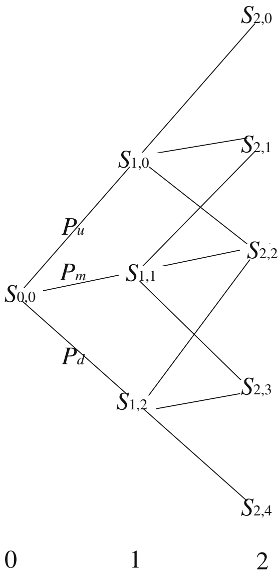


Fig. 2 A two-time-step trinomial lattice. The initial stock price is $S_{0,0}$. $S_{i,j}$ denotes the $(j+1)$th largest stock price at time i. P_u, P_m, and P_d denote the branching probabilities

Define the sum of a partial price path started at time 0 and ended at time j, $S_0 \rightarrow S_1 \rightarrow \cdots \rightarrow S_j$, as $\sum_{i=0}^{j} S_i$. We call this sum a prefix sum. As both the final payoff and the exercise value of the Asian option depend on the average stock price, the option value associated with a partial price path depends on the prefix sum of this partial price path. To price the Asian option, enough numbers of states are required at each node to keep the option values corresponding to different prefix sums. Specifically, the option value at time n can be calculated by Eq. (2) for fixed-strike Asian options or Eq. (3) for floating-strike Asian options. Define $\mathrm{V}(S, C)$ as the option value whose corresponding prefix sum is C and whose stock price is S. Again, let P_u, P_m, and P_d denote the branching probabilities of the node with stock price $S_{i,j}$. The pricing formula for the European-style options can be defined recursively as follows:

$$\begin{aligned}\mathrm{V}(S_{i,j}, C) = \mathrm{e}^{-r\Delta t}\big[&P_u \mathrm{V}(S_{i+1,j}, C + S_{i+1,j}) + P_m \mathrm{V}(S_{i+1,j+1}, C + S_{i+1,j+1})\\ &+ P_d \mathrm{V}(S_{i+1,j+2}, C + S_{i+1,j+2})\big].\end{aligned} \tag{12}$$

The pricing formula for the American-style options can be defined similarly

$$\begin{aligned}\mathrm{V}(S_{i,j}, C) = \max\big(\mathrm{e}^{-r\Delta t}\big[&P_u \mathrm{V}(S_{i+1,j}, C + S_{i+1,j}) + P_m \mathrm{V}(S_{i+1,j+1}, C + S_{i+1,j+1})\\ &+ P_d \mathrm{V}(S_{i+1,j+2}, C + S_{i+1,j+2})\big], E\big),\end{aligned} \tag{13}$$

where exercise value E is defined in Eq. (4) for fixed-strike Asian options or Eq. (5) for floating-strike Asian options. The above formulas rely on the principle of backward induction and can be applied in a backward fashion from time n to time 0. Eventually, $\mathrm{V}(S_{0,0}, S_{0,0})$ gives the desired option value.

The backward-induction procedure implies that the overall running time is proportional to the total number of prefix sums on the lattice. Unfortunately, the number of partial price paths grows exponentially in n, giving as many prefix sums. This makes an exact pricing algorithm explode exponentially. Hull and White [21] suggest to limit the number of prefix sums at each node to some constant k. The resulting approximation algorithm is efficient, with a running time of $O(kn^2)$. But it must resort to interpolation in backward induction because not enough prefix sums have been allocated. Thus interpolation errors are introduced and accumulated at each time step. Applied without care, the Hull–White approach may not converge to the true option value as $n \to \infty$ [14]. No polynomial-time approximation algorithms have rigorously proved convergence guarantees for American-style Asian options. This paper presents a subexponential-time algorithm with such guarantees.

3.3 An integer lattice

We will construct a trinomial lattice composed of integer stock prices. Note that if the stock price of each node in the lattice is an integer, all possible prefix sums must be integers because integers are closed under addition.

The stock price at the root node S is the only price on the lattice that will not be required to be an integer. Express S as $S' + a$ for integer $S' \equiv \lfloor S \rfloor \geq 0$ and real number $a = S - S'$ where $0 \leq a < 1$. Assume that the maximum prefix sum in the lattice is F. Then all possible prefix sums must belong in the set

$$\{\hat{S} : 1 \leq \hat{S} \leq F, \hat{S} = m + a, m \in \mathbb{N}\}.$$

The key to showing that our algorithm breaks the exponential-time barrier is to prove that the size of this set is bounded by a subexponential function in n.

4 Lattice construction

To reduce the number of prefix sums at each node in our lattice, the stock price of each node (except the root node) is restricted to be an integer. To ensure that the stock price process given by our lattice converges to the stock price process (1), the mean and the variance of the logarithmic stock price process are matched at each node in the lattice. In this section, we will first show how the lattice is constructed step by step in order to meet the above two requirements. Proof will be given in the next section to show that our lattice provides a subexponential-time algorithm for Asian options.

The homogeneous property says that

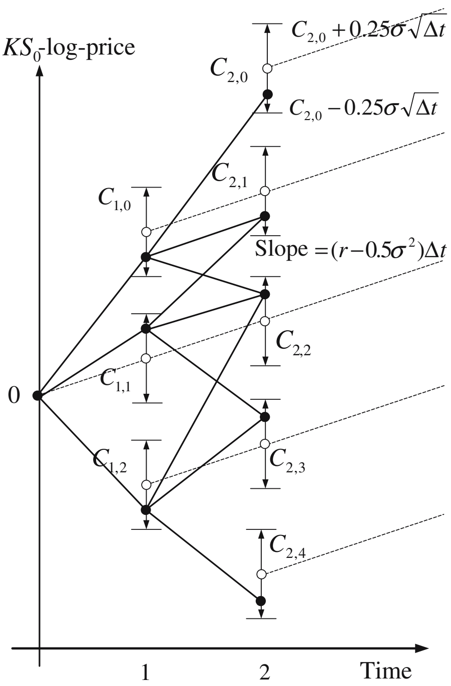
$$E[\max(A - X, 0)] = \frac{E[\max(KA - KX, 0)]}{K}.$$

Thus we can multiply the stock price (S_0) and the exercise price X by a constant K and price this hypothetical option. The desired option value is then obtained by dividing this hypothetical option price by K. To ensure that a proper integer price can be assigned to each node (again, except the root node), let

$$K \equiv (0.25 S_0 \sigma)^{-1} \sqrt{n/T} \exp\left[(0.5\sigma^2 - r)\, T + 2\sigma\sqrt{Tn} \right]. \tag{14}$$

Note that $K \in e^{O(\sqrt{n})}$. We will show later that this K works.

Next a trinomial lattice is constructed to price this hypothetical option. The stock price of the root node ($S_{0,0}$) is equal to KS_0. Our goal is to find integer stock prices $S_{i,j}$ for $0 < i \leq n, 0 \leq j \leq 2i$. Define the V-log-price of stock price V' as $\ln(V'/V)$. Hence a V-log-price of z implies a stock price of Ve^z. Let $c_{i,j} \equiv (r - 0.5\sigma^2)\, i\Delta t + 2(i-j)\, \sigma\sqrt{\Delta t}$. $S_{i,j}$ will be some integer whose KS_0-log-price belongs to the following interval centered



KS_0-log-price
$C_{2,0}+0.25\sigma\sqrt{\Delta t}$
$C_{2,0}$
$C_{2,0}-0.25\sigma\sqrt{\Delta t}$
$C_{2,1}$
$C_{1,0}$
Slope $=(r-0.5\sigma^2)\Delta t$
$C_{2,2}$
0
$C_{1,1}$
$C_{1,2}$
$C_{2,3}$
$C_{2,4}$
1
2
Time

Fig. 3 A two-time-step trinomial lattice over KS_0-log-prices

around $c_{i,j}$:

$$\left(c_{i,j} - 0.25\sigma\sqrt{\Delta t}, c_{i,j} + 0.25\sigma\sqrt{\Delta t}\right).$$

We call $c_{i,j}$ the log-price center for $S_{i,j}$. It will be shown later (Lemma 1) that there always exists an integer at each node whose KS_0-log-price falls within the said interval.

Take the two-time-step trinomial lattice in Fig. 3 as an example. The x-axis marks the time step in the lattice, and the y-axis denotes KS_0-log-prices. Each log-price center is depicted as a hollow circle. Each dotted line segment begins at $c_{i,j}$ and ends at $c_{i+1,j+1}$. The slopes of these dotted lines represent the expected growth rate of the logarithmic stock price: $(r - 0.5\sigma^2)\,\Delta t$. The integer stock price for each node is depicted as a solid circle. Take $S_{2,0}$ as an example. Because $c_{2,0} = (r-0.5\sigma^2)\,2\Delta t + 4\sigma\sqrt{\Delta t}$, the KS_0-log-price of $S_{2,0}$ should fall within the interval $(c_{2,0}-0.25\sigma\sqrt{\Delta t}, c_{2,0}+0.25\sigma\sqrt{\Delta t})$.

The branching probabilities for each node are computed as follows. Take a node with price $S_{i,j}$. Recall that the probabilities for the stock price moving to $S_{i+1,j}$, $S_{i+1,j+1}$, and $S_{i+1,j+2}$ are P_u, P_m, and P_d, respectively. Define α, β, and γ as follows:

$$\alpha \equiv \ln(S_{i+1,j}/S_{i,j}) - \mu, \tag{15}$$

$$\beta \equiv \ln(S_{i+1,j+1}/S_{i,j}) - \mu, \tag{16}$$

$$\gamma \equiv \ln(S_{i+1,j+2}/S_{i,j}) - \mu, \tag{17}$$

where μ is defined in Eq. (7). The branching probabilities satisfy

$$P_u\alpha + P_m\beta + P_d\gamma = 0, \tag{18}$$

$$P_u\alpha^2 + P_m\beta^2 + P_d\gamma^2 = \text{Var}, \tag{19}$$

$$P_u + P_m + P_d = 1, \tag{20}$$

where Var is defined in Eq. (8). Equations (18) and (19) match the mean and the variance of the logarithmic stock price, respectively. Hence our lattice converges to the lognormal stock price process. That Eqs. (18)–(20) give valid branching probabilities will be proved later (Lemma 2).

The key result of this paper is stated below.

Theorem 1 *The integral lattice discussed in this section can be constructed, and the pricing algorithms for Asian options on this lattice run in* $2^{O(\sqrt{n})}$ *time.*

5 Proof of Theorem 1

The proof of Theorem 1 consists of two parts. First, we will prove that a valid lattice can be constructed. Next we will show that our pricing algorithms based on this lattice run in $2^{O(\sqrt{n})}$ time.

When constructing the trinomial lattice, an integral stock price is assigned to each node (except the root node). The following lemma shows that we can always find a proper integral stock price for each $S_{i,j}$.

Lemma 1 *For each* $S_{i,j}$*, these exists an integer whose* KS_0*-log-price falls in* $(c_{i,j} - 0.25\sigma\sqrt{\Delta t}, c_{i,j} + 0.25\sigma\sqrt{\Delta t})$*, where* $0 < i \leq n$ *and* $0 \leq j \leq 2i$*.*

Proof To ensure that such an integer exists, it suffices to show that

$$KS_0\big[\exp\big(c_{i,j} + 0.25\sigma\sqrt{\Delta t}\big) - \exp\big(c_{i,j} - 0.25\sigma\sqrt{\Delta t}\big)\big] > 1 \tag{21}$$

with our choice of K in Eq. (14).

Without loss of generality, only the case where $i = n$ and $j = 2n$ is considered because it minimizes the term within the brackets in Eq. (21). Thus it suffices to show that our K satisfies

$$K > S_0^{-1}\mathrm{e}^{-c_{n,2n}}\left(\mathrm{e}^{0.25\sigma\sqrt{\Delta t}} - \mathrm{e}^{-0.25\sigma\sqrt{\Delta t}}\right)^{-1}.$$

Indeed,

$$\begin{aligned}
&S_0^{-1}\mathrm{e}^{-c_{n,2n}}\left(\mathrm{e}^{0.25\sigma\sqrt{\Delta t}} - \mathrm{e}^{-0.25\sigma\sqrt{\Delta t}}\right)^{-1}\\
&\quad= S_0^{-1}\exp\big[\,(0.5\sigma^2 - r)\,T + 2\sigma\sqrt{Tn}\,\big]\\
&\qquad\times\left(\mathrm{e}^{0.25\sigma\sqrt{\Delta t}} - \mathrm{e}^{-0.25\sigma\sqrt{\Delta t}}\right)^{-1}\\
&\quad< S_0^{-1}\exp\big[\,(0.5\sigma^2 - r)\,T + 2\sigma\sqrt{Tn}\,\big]/\big(0.25\sigma\sqrt{T/n}\big)\\
&\quad\leq (0.25S_0\sigma)^{-1}\sqrt{n/T}\exp\big[\,(0.5\sigma^2 - r)\,T + 2\sigma\sqrt{Tn}\,\big]\\
&\quad= \mathrm{e}^{O(\sqrt{n})},
\end{aligned} \tag{22}$$

where Eq. (22) holds because

$$\mathrm{e}^{0.25\sigma\sqrt{\Delta t}} - \mathrm{e}^{-0.25\sigma\sqrt{\Delta t}} > \big(1 + 0.25\sigma\sqrt{\Delta t}\big) - 1 = 0.25\sigma\sqrt{\Delta t}.$$

□

Next, we show that all the branching probabilities of our lattice are valid.

Lemma 2 *Equations* (18)–(20) *result in valid branching probabilities for the stock price* $S_{i,j}$*, where* $0 < i \leq n$ *and* $0 \leq j \leq 2i$.

Proof It suffices to prove that $P_u, P_m, P_d > 0$. We first derive constraints on α, β, γ defined in Eqs. (15)–(17). Then we show that valid branching probabilities are obtained by solving Eqs. (18)–(20) given the constraints on α, β, and γ.

We use the plot in Fig. 4 to aid the proof. Observe that the KS_0-log-price of x becomes the $S_{i,j}$-log-price of $x + \frac{\ln KS_0}{\ln S_{i,j}}$. The following computations are in $S_{i,j}$-log-prices unless stated otherwise. The log-price center of $S_{i,j}$, i.e., c', equals $c_{i,j} + \frac{\ln KS_0}{\ln S_{i,j}}$, whereas c, the log-price center of $S_{i+1,j+1}$, equals $c' + (r - 0.5\sigma^2)\Delta t$. The log-price centers of $S_{i+1,j}$ and $S_{i+1,j+2}$ are $c + 2\sigma\sqrt{\Delta t}$ and $c - 2\sigma\sqrt{\Delta t}$, respectively. The $S_{i,j}$-log-price of $S_{i,j}$ is 0. The conditional mean of the stock price one time step after it reaches $S_{i,j}$ is $\mu = (r - 0.5\sigma^2)\Delta t$. By construction, the distance between the $S_{i,j}$-log-price of $S_{i,j}$, which equals 0, and its log-price center c' is smaller than $0.25\sigma\sqrt{\Delta t}$. This implies that $|c'| < 0.25\sigma\sqrt{\Delta t}$. Hence,

$$\begin{aligned} |\mu - c| &= \left|(r - 0.5\sigma^2)\Delta t - \left[c' + (r - 0.5\sigma^2)\Delta t\right]\right| \\ &= |c'| < 0.25\sigma\sqrt{\Delta t}. \end{aligned}$$

Thus μ falls within interval $(c - 0.25\sigma\sqrt{\Delta t}, c + 0.25\sigma\sqrt{\Delta t})$. Now $\beta = \ln(S_{i+1,j+1}/S_{i,j}) - \mu$ falls within interval $(-0.5\sigma\sqrt{\Delta t}, 0.5\sigma\sqrt{\Delta t})$ as the $S_{i,j}$-log-price of $S_{i+1,j+1}$, i.e.,

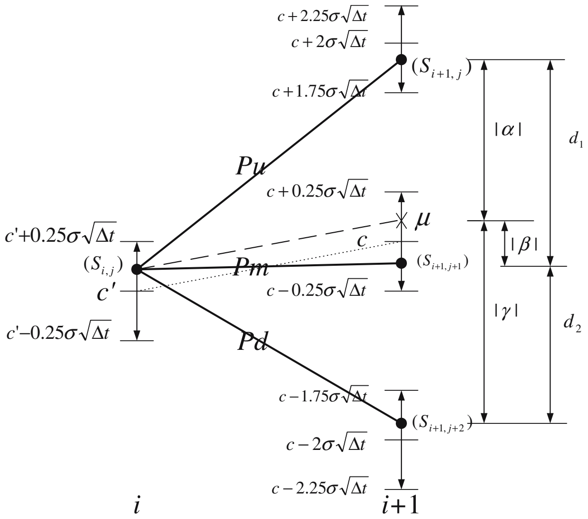

$c+2.25\sigma\sqrt{\Delta t}$
$c+2\sigma\sqrt{\Delta t}$
$(S_{i+1,j})$
$c+1.75\sigma\sqrt{\Delta t}$
$|\alpha|$
d_1
Pu
$c+0.25\sigma\sqrt{\Delta t}$
μ
$c'+0.25\sigma\sqrt{\Delta t}$
c
$|\beta|$
$(S_{i,j})$
Pm
$(S_{i+1,j+1})$
c'
$c-0.25\sigma\sqrt{\Delta t}$
$|\gamma|$
d_2
$c'-0.25\sigma\sqrt{\Delta t}$
Pd
$c-1.75\sigma\sqrt{\Delta t}$
$(S_{i+1,j+2})$
$c-2\sigma\sqrt{\Delta t}$
$c-2.25\sigma\sqrt{\Delta t}$
i
$i+1$

Fig. 4 Branching probabilities for the node with price $S_{i,j}$. All the values in this figure are $S_{i,j}$-log-prices except the ones that are *parenthesized*. The nodes with stock prices $S_{i,j}$, $S_{i+1,j}$, $S_{i+1,j+1}$, and $S_{i+1,j+2}$ are represented by *solid circles*. The branches that connect these nodes are represented by *thick lines*. The log-price centers of $S_{i,j}$ and $S_{i+1,j+1}$ are c' and c, respectively. P_u, P_m, and P_d denote the branching probabilities for the upper, middle, and lower branches from the node with price $S_{i,j}$, respectively. Values α, β, and γ are defined in Eqs. (15)–(17). Finally, d_1 denotes the distance between $S_{i+1,j}$'s and $S_{i+1,j+1}$'s $S_{i,j}$-log-prices, and d_2 denotes the distance between $S_{i+1,j+1}$'s and $S_{i+1,j+2}$'s $S_{i,j}$-log-prices

$\ln(S_{i+1,j+1}/S_{i,j})$, falls within interval $(c - 0.25\sigma\sqrt{\Delta t}, c + 0.25\sigma\sqrt{\Delta t})$. Figure 4 illustrates a case where $\beta < 0$.

We next represent α and γ in terms of β. Define d_1 as the distance between $S_{i+1,j}$'s and $S_{i+1,j+1}$'s $S_{i,j}$-log-prices and d_2 as the distance between $S_{i+1,j+1}$'s and $S_{i+1,j+2}$'s $S_{i,j}$-log-prices. Thus α and γ can be represented as

$$\begin{aligned}\alpha &= \ln(S_{i+1,j}/S_{i,j}) - \ln(S_{i+1,j+1}/S_{i,j}) + \ln(S_{i+1,j+1}/S_{i,j}) - \mu\\ &= d_1 + \beta,\\ \gamma &= \ln(S_{i+1,j+2}/S_{i,j}) - \ln(S_{i+1,j+1}/S_{i,j}) + \ln(S_{i+1,j+1}/S_{i,j}) - \mu\\ &= -d_2 + \beta.\end{aligned}$$

Note that $1.5\sigma\sqrt{\Delta t} < d_1 < 2.5\sigma\sqrt{\Delta t}$ because

$$\begin{gathered}c + 1.75\sigma\sqrt{\Delta t} < \ln(S_{i+1,j}/S_{i,j}) < c + 2.25\sigma\sqrt{\Delta t},\\ c - 0.25\sigma\sqrt{\Delta t} < \ln(S_{i+1,j+1}/S_{i,j}) < c + 0.25\sigma\sqrt{\Delta t},\\ d_1 = \ln(S_{i+1,j}/S_{i,j}) - \ln(S_{i+1,j+1}/S_{i,j}).\end{gathered}$$

By the same argument, $1.5\sigma\sqrt{\Delta t} < d_2 < 2.5\sigma\sqrt{\Delta t}$. Note also that $\alpha = d_1 + \beta > \sigma\sqrt{\Delta t} > 0$ as $\beta \in (-0.5\sigma\sqrt{\Delta t}, 0.5\sigma\sqrt{\Delta t})$ and $d_1 \in (1.5\sigma\sqrt{\Delta t}, 2.5\sigma\sqrt{\Delta t})$. Similarly, $\gamma = -d_2 + \beta < -\sigma\sqrt{\Delta t} < 0$ as $d_2 \in (1.5\sigma\sqrt{\Delta t}, 2.5\sigma\sqrt{\Delta t})$. Thus $\alpha > \beta > \gamma$.

We now show that positive branching probabilities are obtained given the constraints on α, β, and γ derived above and summarized below:

$$\begin{aligned}&\beta \in (-0.5\sigma\sqrt{\Delta t},\ 0.5\sigma\sqrt{\Delta t}),\\ &\alpha = d_1 + \beta, \quad \text{where } d_1 \in (1.5\sigma\sqrt{\Delta t}, 2.5\sigma\sqrt{\Delta t}),\\ &\gamma = -d_2 + \beta, \quad \text{where } d_2 \in (1.5\sigma\sqrt{\Delta t}, 2.5\sigma\sqrt{\Delta t}).\end{aligned}$$

Define

$$\begin{aligned}\det &= (\beta - \alpha)(\gamma - \alpha)(\gamma - \beta),\\ \det_u &= (\beta\gamma + \mathrm{Var})(\gamma - \beta),\\ \det_m &= (\alpha\gamma + \mathrm{Var})(\alpha - \gamma),\\ \det_d &= (\alpha\beta + \mathrm{Var})(\beta - \alpha).\end{aligned}$$

Then $P_u = \det_u/\det$, $P_m = \det_m/\det$, and $P_d = \det_d/\det$ by applying Cramer's rule to Eqs. (18)–(20). Note that $\det < 0$ as $\alpha > \beta > \gamma$. To show that the branching probabilities are valid, we have to show that $P_u, P_m, P_d > 0$. As $\det < 0$, it is sufficient to show $\det_u, \det_m, \det_d < 0$. Finally, as $\gamma - \beta < 0$, $\alpha - \gamma > 0$, and $\beta - \alpha < 0$, we only need to show that $\beta\gamma + \mathrm{Var} > 0$, $\alpha\gamma + \mathrm{Var} < 0$, and $\alpha\beta + \mathrm{Var} > 0$ instead.

1. $\beta\gamma + \mathrm{Var} > 0$: note that

$$\beta\gamma + \mathrm{Var} = \beta(\beta - d_2) + \sigma^2\Delta t = (\beta - 0.5d_2)^2 - 0.25d_2^2 + \sigma^2\Delta t.$$

For a given d_2, $\beta\gamma + \mathrm{Var}$ reaches its minimum when $\beta = 0.5d_2$. Recall the constraints that $0.5d_2 \in (0.75\sigma\sqrt{\Delta t}, 1.25\sigma\sqrt{\Delta t})$ and $\beta < 0.5\sigma\sqrt{\Delta t}$. Hence $\beta\gamma + \mathrm{Var}$ reaches its minimum for a given $d_2 \in (1.5\sigma\sqrt{\Delta t}, 2.5\sigma\sqrt{\Delta t})$ when $\beta \to 0.5\sigma\sqrt{\Delta t}$. Thus we have

$$\begin{aligned}\beta\gamma + \mathrm{Var} &= \beta(\beta - d_2) + \sigma^2\Delta t\\ &> 0.5\sigma\sqrt{\Delta t}\,(0.5\sigma\sqrt{\Delta t} - d_2) + \sigma^2\Delta t\\ &= -0.5\sigma\sqrt{\Delta t}\,d_2 + 1.25\sigma^2\Delta t\\ &> -1.25\sigma^2\Delta t + 1.25\sigma^2\Delta t = 0.\end{aligned}$$

2. $\alpha\gamma + \mathrm{Var} < 0$: note that

$$\alpha\gamma + \mathrm{Var} = (\beta + d_1)(\beta - d_2) + \sigma^2\Delta t.$$

As $\beta + d_1 > \sigma\sqrt{\Delta t}$ and $\beta - d_2 < -\sigma\sqrt{\Delta t}$, we have

$$(\beta + d_1)(\beta - d_2) + \sigma^2\Delta t < (\sigma\sqrt{\Delta t})(-\sigma\sqrt{\Delta t}) + \sigma^2\Delta t = 0.$$

3. $\alpha\beta + \mathrm{Var} > 0$: the proof is similar to the one for $\beta\gamma + \mathrm{Var} > 0$. Note that

$$\alpha\beta + \mathrm{Var} = (\beta + d_1)\beta + \sigma^2\Delta t = (\beta + 0.5d_1)^2 - 0.25d_1^2 + \sigma^2\Delta t.$$

For a given d_1, $\alpha\beta + \mathrm{Var}$ reaches its minimum when $\beta = -0.5d_1$. Recall the constraints that $-0.5d_1 \in (-1.25\sigma\sqrt{\Delta t}, -0.75\sigma\sqrt{\Delta t})$ and $\beta > -0.5\sigma\sqrt{\Delta t}$. Hence $\alpha\beta + \mathrm{Var}$ reaches its minimum for a given $d_1 \in (1.5\sigma\sqrt{\Delta t}, 2.5\sigma\sqrt{\Delta t})$ when $\beta \to -0.5\sigma\sqrt{\Delta t}$. Thus

$$\begin{aligned}\alpha\beta + \mathrm{Var} &= (\beta + d_1)\beta + \sigma^2\Delta t\\ &> (-0.5\sigma\sqrt{\Delta t} + d_1)(-0.5\sigma\sqrt{\Delta t}) + \sigma^2\Delta t\\ &= -0.5\sigma\sqrt{\Delta t}\,d_1 + 1.25\sigma^2\Delta t\\ &> -1.25\sigma^2\Delta t + 1.25\sigma^2\Delta t = 0.\end{aligned}$$

□

Finally, the time complexity is proved.

Lemma 3 *Our pricing algorithm for European-style Asian options runs in* $2^{O(\sqrt{n})}$ *time.*

Proof Note that our pricing algorithm will evaluate the option values for all the prefix sums on the lattice by Eq. (12). Thus the time complexity of our algorithm is proportional to the total number of prefix sums on the lattice. Thus the pricing algorithm is subexponential in n if the total number of prefix sums is bounded by a subexponential function. We will first show that the maximum prefix sum in our lattice is bounded by a subexponential function. Then we will show that the total number of prefix sums is, too.

The maximum stock price $S_{n,0}$ is bounded by $KS_0e^{c_{n,0}+0.25\sigma\sqrt{\Delta t}}$, where

$$\begin{aligned}c_{n,0} &= (r - 0.5\sigma^2)n\Delta t + 2(n-0)\sigma\sqrt{\Delta t}\\ &= (r - 0.5\sigma^2)\,T + 2\sigma\sqrt{Tn}\,.\end{aligned}$$

Thus $S_{n,0}$ is bounded by a subexponential function as both K and $\mathrm{e}^{c_{n,0}+0.25\sigma\sqrt{\Delta t}}$ are subexponential in n. The maximum prefix sum in the lattice is equal to $\sum_{i=0}^{n} S_{i,0} \leq (n+1)S_{n,0}$. Note that $(n+1)S_{n,0}$ is also subexponential in n. Let $F \equiv (n+1)S_{n,0}$ for later use.

The next goal is to show that the total number of prefix sums is bounded by a subexponential function. Recall that the stock price for each node except the root node is an integer. The stock price at the root node (KS_0) can be represented as $S' + a$ for some integer $S' \geq 0$ and some real number a where $0 \leq a < 1$. Thus all possible prefix sums must belong to the set

$$\{\hat{S} : 1 \leq \hat{S} \leq F, \hat{S} = m + a, m \in \mathbb{N}\}.$$

The cardinality of the set is at most $\lceil F \rceil$. Thus the maximum number of prefix sums for each node is bounded from above by $\lceil F \rceil$. Since there are $(n+1)^2$ nodes in an n-time-step trinomial lattice, the total number of prefix sums is bounded from above by $(n+1)^2 \lceil F \rceil$, which is subexponential. The time complexity of our algorithm is thus subexponential in n. □

Corollary 1 *Our pricing algorithm for American-style Asian options runs in* $2^{O(\sqrt{n})}$ *time.*

Proof Our pricing algorithm will evaluate the option values for all the prefix sums on the lattice by Eq. (13). Equation (13) can be evaluated in constant time since the exercise value E can also be evaluated in constant time by Eq. (4) for fixed-strike Asian options or Eq. (5) for floating-strike Asian options. Thus our pricing algorithm runs in $2^{O(\sqrt{n})}$ time. □

Corollary 2 *Our pricing algorithm for partial-averaging Asian option runs in* $2^{O(\sqrt{n})}$ *time.*

Proof The payoff of a partial-averaging Asian option depends on the average stock price over only a part of the option's life. We simply need to keep track of all possible "partial"-average stock prices during backward induction. Obviously, the number of partial-average stock price is less than the number of possible average stock price. Exact backward induction can be applied on our lattice in $2^{O(\sqrt{n})}$ time. □

6 Numerical results

The numerical experiment illustrated in Table 1 gives an idea of the performance of our subexponential-time algorithm. The numerical data are obtained by running our program on an Intel Pentium IV 3.0 G computer with 1 G RAM. The quick convergence of our algorithm can be observed in Table 1 where the prices remain unchanged

Table 1 Pricing a European-style fixed-strike Asian option

n	`Value`	`Running time`
20	6.01	0.016
40	6.02	0.219
60	6.02	1.008
80	6.02	3.055
100	6.02	7.578

`Value` and `Running time` denote the pricing results and the running times of our algorithm, respectively. The stock price and the exercise price are 100, the volatility is 30%, the risk-free interest rate is 10%, and the time to maturity is 0.5 year down to pennies when the number of time steps is at least 40. The running time is reasonable and much less than 3^n. For example, the program finishes in 7.578 s for $n = 100$, whereas $3^{100} \approx 5.153 \times 10^{47}$, making the naive exponential-time algorithm hopeless.

7 Conclusions

This paper develops a new trinomial lattice particularly with the Asian option in mind. The lattice uses the notion of integrality of stock prices to reduce the time complexity of an exact pricing algorithm from exponential to subexponential. The proposed pricing algorithm is guaranteed to converge to the true option value at a rate of $O(n^{-1})$, where n denotes the number of time steps of our lattice. It is the first exact and convergent lattice algorithm to break the exponential-time barrier.

Acknowledgements The authors would like to thank the very helpful comments from two anonymous referees.

References

1. Abate, J., Whitt, W.: Numerical inversion of Laplace transforms of probability distributions. ORSA J. Comput. **7**, 36–43 (1995)
2. Aingworth, D., Motwani, R., Oldham, J.D.: Accurate approximations for Asian options. In: Proceedings of 11th Annual ACM-SIAM Symposium on Discrete Algorithms (2000)
3. Barraquand, J., Pudet, T.: Pricing of American path-dependent contingent claims. Math. Finance **6**, 17–51 (1996)
4. Black, F., Scholes, M.: The pricing of options and corporate liabilities. J. Polit. Econ. **81**, 637–659 (1973)
5. Boyle, P.P., Broadie, M., Glasserman, P.: Monte Carlo methods for security pricing. J. Econ. Dyn. Control **21**, 1267–1321 (1997)
6. Broadie, M., Glasserman, P.: Estimating security price derivatives using simulation. Manag. Sci. **42**, 269–285 (1996)
7. Broadie, M., Glasserman, P., Kou, S.: Connecting discrete and continuous path-dependent options. Finance Stoch **3**, 55–82 (1999)
8. Chalasani, P., Jha, S., Egriboyun, F., Varikooty, A.: A refined binomial lattice for pricing American Asian options. Rev. Deriv. Res. **3**, 85–105 (1999)
9. Chalasani, P., Jha, S., Saias, I.: Approximate option pricing. Algorithmica **25**, 2–21 (1999)
10. Cox, J., Ross, S., Rubinstein, M.: Option pricing: a simplified approach. J. Financ. Econ. **7**, 229–264 (1979)
11. Dai, T.-S., Huang, G.-S., Lyuu, Y.-D.: An efficient convergent lattice algorithm for European Asian options. Appl. Math. Comput. **169**, 1458–1471 (2005)
12. Dai, T.-S., Lyuu, Y.-D.: Efficient, exact algorithms for Asian options with multiresolution lattices. Rev. Deriv. Res. **5**, 181–203 (2002)
13. Duffie, D.: Dynamic Asset Pricing Theory, 2nd edn. Princeton University Press, Princeton (1996)
14. Forsyth, P.A., Vetzal, K.R., Zvan, R.: Convergence of numerical methods for valuing path-dependent options using interpolation. Rev. Deriv. Res. **5**, 273–314 (2002)
15. Fu, M.C., Dilip, D.B., Wang, T.: Pricing continuous Asian options: a comparison of Monte Carlo and Laplace transform inversion methods. J. Comput. Finance **2**, 49–74 (1998/9)
16. Geman, H., Eydeland, A.: Domino effect. Risk **8**, 65–67 (1995)
17. Geman, H., Yor, M.: Bessel processes, Asian options, and perpetuities. Math. Finance **3**, 349–375 (1993)
18. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, Heidelberg (2004)
19. Harrison, J.M., Pliska, S.R.: Martingales and stochastic integrals in the theory of continuous trading. Stoch. Process. Appl. **11**, 215–260 (1981)

20. Hull, J.: Options, Futures, and Other Derivatives, 4th edn. Prentice-Hall, Englewood Cliffs (2000)
21. Hull, J., White, A.: Efficient procedures for valuing European and American path-dependent options. J. Deriv. **1**, 21–31 (1993)
22. Ingersoll, J.E.: Theory of Financial Decision Making. Rowman & Littlefield, Savage (1987)
23. Kemna, A.G.Z., Vorst, A.C.F.: A pricing method based on average asset values. J. Bank. Finance **14**, 113–129 (1990)
24. Klassen, T.R.: Simple, fast and flexible pricing of Asian options. J. Comput. Finance **4**, 89–124 (2001)
25. Levy, E.: Pricing European average rate currency options. J. Int. Money Finance **11**, 474–491 (1992)
26. Longstaff, F.A., Schwartz, E.S.: Valuing American options by simulation: a simple least-squares approach. Rev. Financ. Stud. **14**, 113–147 (2001)
27. Lyuu, Y.-D.: Financial Engineering and Computation: Principles, Mathematics, Algorithms. Cambridge University Press, Cambridge (2002)
28. Merton, R.C.: Continuous-Time Finance (revised edn). Blackwell, Cambridge (1994)
29. Milevsky, M.A., Posner, S.E.: Asian options, the sum of lognormals, and the reciprocal gamma distribution. J. Financ. Quant. Anal. **33**, 409–422 (1998)
30. Ohta, K., Sadakane, K., Shioura, A., Tokuyama, T.: A fast, accurate and simple method for pricing European-Asian and saving-Asian options. In: Proceedings of the 10th Annual European Symposium on Algorithms (2002)
31. Reiner, E., Rubinstein, M.: Breaking down the barriers. Risk **4**, 28–35 (1991)
32. Ritchken, P., Sankarasubramanian, L., Vijh, A.M.: The valuation of path dependent contracts on the average. Manage. Sci. **39**, 1202–1213 (1993)
33. Rogers, L.C.G., Shi, Z.: The value of an Asian option. J. Appl. Probab. **32**, 1077–1088 (1995)
34. Shaw, W.T.: Modeling Financial Derivatives with Mathematica. Cambridge University Press, Cambridge (1998)
35. Turnbull, S.M., Wakeman, L.M.: A quick algorithm for pricing European average options. Financ. Quant. Anal. **26**, 377–389 (1991)
36. Večeř, J.: A new PDE approach for pricing arithmetic average Asian options. J. Comput. Finance **4**, 105–113 (2001)
37. Večeř, J.: Unified Asian pricing. Risk **15**, 113–116 (2002)
38. Zhang, J.E.: A semi-analytical method for pricing and hedging continuously sampled arithmetic average rate options. J. Comput. Finance **5**, 59–79 (2001)
39. Zhang, J.E.: Pricing continuously sampled Asian options with perturbation method. J. Futures Mark. **23**, 535–560 (2003)
40. Zvan, R., Vetzal, K.: Discrete Asian barrier options. J. Comput. Finance **3**, 41–68 (1999)