Accurate Approximations for European-Style Asian Options

Prasad Chalasani Somesh Jha Ashok Varikooty

May 1997
CMU-CS-97-140

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In the binomial tree model, we provide efficient algorithms for computing an accurate lower bound for the value of a European-style Asian option with either a fixed or a floating strike. These algorithms are inspired by the continuous-time analysis of Rogers and Shi [10]. Specifically we consider lower bounds on the option value that are given by the expectation of the conditional expectation of the payoff conditioned on some random variable Z. For a specific Z, Rogers and Shi estimate this conditional expectation numerically in continuous time, and show experimentally that their lower bound is very accurate. We consider a modified random variable Z that gives a strictly better lower bound. In addition, we show that this lower bound can be computed *exactly* in the n-step binomial tree model in time proportional to n^7. We show that computing the approximation is equivalent to counting paths of various types, and that this can be done efficiently by a dynamic programming technique. We present other choices of Z that yield accurate and efficiently-computable lower bounds. We also show algorithms to compute a bound on the error of these approximations, so that we can compute an upper bound on the option value as well.

This article has been submitted to the Journal of Derivatives. We would like to thank Krishna Ramaswamy (Wharton) and Rui Kan (CS First Boston) for useful comments and discussions. Any errors are our own responsibility. The affiliations of the authors are: **Prasad Chalasani**, Los Alamos National Laboratory, chal@lanl.gov, http://www.c3.lanl.gov/~chal; **Somesh Jha**, Carnegie Mellon University, sjha@cs.cmu.edu, http://www.cs.cmu.edu/~sjha; **Ashok Varikooty**, CS First Boston, avarikoo@fir.fbc.com.

Keywords: Computational finance, option pricing, asian options, dynamic programming

1 Introduction

The binomial tree model of Cox, Ross and Rubinstein [3] is commonly used as a discrete-time approximation to a continuous-time geometric Brownian motion. The pricing of simple calls and puts in this model can be done efficiently. However, *path-dependent* options such as Asian options are notoriously hard to price in this model. The Black-Scholes differential equation for the price of such options has no known closed form solution.

The payoff of an Asian option is based on the *arithmetic average* of the stock price. In particular, suppose the option expires at time T and let A_T denote the arithmetic average of the stock prices up to time T. Then the payoff of a European style Asian call with strike L is $(A_T - L)^+$ where x^+ denotes $\max\{x, 0\}$. Similarly the payoff of a European-style Asian put with strike L is $(L - A_T)^+$. For the so-called "floating-strike" versions of these options, the L is replaced by S_T, the stock price at time T. Such options are of obvious appeal to a company which must buy a commodity at a fixed time each year, yet has to sell it regularly throughout the year [13]. These options allow investors to eliminate losses from movements in an underlying asset without the need for continuous rehedging. Asian options are commonly used for currencies [13], interest-rates and commodities such as crude oil [5].

Several researchers have devised approximation algorithms for Asian options. This work has been of three general types. Monte Carlo methods and Fourier transform techniques have been explored by Kemna and Vorst [6], and Carverhill and Clewlow [2] respectively. The second category of approaches consists of attempts to approximate the distribution of the arithmetic average (for which no simple specification is known) by a more tractable one. Examples of such work include those by Ruttiens [11], Levy [7], [8], Levy and Turnbull [9], and Turnbull and Wakeman [12]. The third approach, considered by Yor [14] and Geman and Yor [4], is to derive formulas for the Laplace transform of an Asian option. However, there have been no numerical studies of the inversion of this Laplace transform, and no simple analytical inversion has been found. None of the above methods offers an approximation for the Asian option price that is both fast and very accurate.

The work in the present paper is most closely related to that of Rogers and Shi [10]. They provide a method for computing *lower bounds* on the price of an Asian option in the continuous-time geometric Brownian motion model. Their main idea is the following. Let W_T^+ be the final payoff of a European style Asian (call or put) option. For instance, for a European style Asian call option, $W_T = A_T - L$. Then for any random variable Z, we can lower-bound the option value using Jensen's inequality:

$$\begin{aligned} \mathbf{E}(W_T^+) &= \mathbf{E}\left(\mathbf{E}(W_T^+|Z)\right) \\ &\geq \mathbf{E}\left(\mathbf{E}(W_T|Z)^+\right) \end{aligned} \tag{1}$$

The lower-bound on the right in (1) can be used as an approximation for $\mathbf{E}(W_T^+)$. The quality of the approximation clearly depends on the choice of the random variable Z. This approximation is an integral in continuous time, and Rogers and Shi *estimate* this integral numerically for various choices of Z. In the present paper, we show that for several choices of the random variable Z we can efficiently compute the approximation (1) *exactly* in the binomial tree model. We show this for random variables Z that are related to, but not exactly the same as, the ones considered by Rogers and Shi.

One way to think of the above approximation in the n-step Binomial tree model is the following. We let W_n and A_n denote the random-variables corresponding to the continuous-time random-variable W_T and A_T respectively. The difficulty in computing the exact option value $\mathbf{E}(W_n^+)$ comes from the fact that nearly every stock price path has a different (arithmetic) average stock price (and therefore a different W_n), and there are 2^n paths. The approximation (1) can be thought of as the value of an option with W_n replaced by W'_n, defined as follows. Let C_i be the collection of all paths with the same value of the random variable $Z = z_i$. Now let the W'_n on *every* path in C_i be equal to the *average* value of W_n over the collection C_i. In other words, in using (1) to estimate the option value, we are making the approximation that all paths with $Z = z_i$ have the same average stock price, equal to the *average of the average* stock price over these paths. This view will be useful to understand intuitively why some choices of Z yield better lower bounds than others.

Rogers and Shi find that a particularly good lower bound is obtained by taking

$$Z = \int_0^T B_u \, du,$$

which is T times the average value of the Brownian from time 0 to time T. In the n-step binomial tree model, the Brownian average corresponds to the average of the underlying *random walk.* The steps of this random walk are given by the random variables $X_i = \pm 1$, $i = 1, 2, \ldots, n$. The position of this random walk at time k is given by $Y_0 = 0$, and

$$Y_k = \sum_{i=1}^{k} X_i, \quad k = 1, 2, \ldots, n.$$

The average position from time 0 to time n is Z_n/n where

$$Z_k = \sum_{i=0}^{k} Y_i, \quad k = 0, 2, \ldots, n.$$

The problem with choosing $Z = Z_n/n$ is that one cannot compute the expression (1) efficiently in the binomial model. However, if we take $Z = (Z_n, S_n)$ where S_n is the stock price at time n, then (1) is a strictly better approximation, *and* it can be computed exactly in time proportional to n^7 (Section 5). We also show by means of numerical experiments (Section 7) that the approximation (1) for this choice of Z agrees excellently with the exact option value in the binomial model.

Note that the *geometric* stock price average is a simple function of Z_n. This helps explain intuitively why conditioning on (Z_n, S_n) yields a good approximation: paths with the same geometric stock-price average that end at the same stock price will likely have arithmetic stock-price averages that do not differ too much. The reason why this approximation can be computed in time proportional to only n^7 (rather than 2^n) is that the number of possible values of Z_n is only proportional to n^2. By contrast the number of possible values of the stock price average A_n is 2^n. As we show in Section 5, this allows use to use dynamic programming to compute the approximation in time proportional to n^7.

In this paper we also consider $Z = S_{n_1}, S_{n_2}, \ldots, S_{n_d}, S_n$, where $0 < n_1 < n_2 < \ldots < n_d < n$ are fixed integers. We are thus conditioning on the stock prices at certain fixed times. We show

(Section 4) that for a fixed d, the expression (1) can be computed exactly in the binomial tree model in time proportional to dn^{d+4}. Thus as we increase d we get increasingly better approximations, if we are willing to spend more computation time. In Section 6 we describe a hybrid approximation where we condition on Z_n in addition to these stock prices. We show that for all our choices of Z, our algorithms require time proportional to some constant power of n. Thus our running times are *polynomial* in n rather than exponential.

Rogers and Shi show that the error in the above approximation is bounded above by

$$\tfrac{1}{2}\mathbf{E}[\mathrm{var}(W|Z)^{\frac{1}{2}}]. \tag{2}$$

We show that this bound can also be computed efficiently for our choices of Z. Thus we can efficiently compute both a lower-bound and an upper bound on the option price. Rogers and Shi do not show how to compute their approximations or errors efficiently in the binomial tree model.

The paper is organized as follows. In Section 2 we describe the binomial model and related notation. In Sections 3, 4, 5 and 6 we describe polynomial-time algorithms to compute the lower and upper bounds on the price of an Asian option. Throughout we illustrate our ideas for a fixed strike Asian call, and it should be clear how to extend these to Asian puts, and floating-strike Asian options. In Section 7 we show numerical results.

2 The binomial model for stock prices

In this paper we consider an Asian option written on an underlying risky asset, which we refer to as the "stock". Following Black and Scholes [1], we make the standard assumption that the price of the stock at time t satisfies the stochastic differential equation for geometric Brownian motion:

$$dS(t) = \mu S(t)\ dt + \sigma S(t)\ dB(t),$$

where the drift rate μ and volatility σ are constant. $B(t)$ is a Brownian motion process. We will also assume that the risk-free interest rate is r, a constant. The derivative security expires at time T.

The continuous-time function $S(t)$ can be approximated in the form of a Cox-Ross-Rubinstein [3] binomial tree, as follows. The life of the option is divided into n time steps of length $\Delta t = T/n$. In time Δt the stock price moves up by a factor u with probability p, or down by a factor $d = 1/u$ with probability $q = 1 - p$, where

$$u = e^{\sigma\sqrt{\Delta t}},$$

$$p = \frac{e^{\mu\Delta t} - d}{u - d}.$$

Time k in this discrete model corresponds to continuous-time $k\Delta t$. Also we assume that a unit of currency lent or borrowed at time k will be worth $1 + r$ units at time $k + 1$. For brevity we write R to denote $1 + r$. This binomial tree model will be assumed throughout the paper.

More formally, we have a sample space Ω of all possible sequences of n coin-tosses (i.e. H's and T's), where an H corresponds to an up-tick of the stock and a T corresponds to a down-tick. A typical sequence (or "path") in Ω is written $\omega = \omega_1\omega_2\ldots\omega_n$, where ω_i represents the ith coin-toss and $\omega_i \in \{H, T\}$. A **k-prefix** of a path ω is the subsequence $\omega_1\omega_2 \ldots \omega_k$ constisting of the first k coin-tosses. For $k = 1, 2, \ldots, n$, define the random variable X_k on Ω as follows:

$$X_k(\omega) = \begin{cases} 1 & \text{if } \omega_k = H, \\ -1 & \text{if } \omega_k = T. \end{cases}$$

We will need the following random variables:

$$Y_0 = 0,$$

$$Y_k = \sum_{i=1}^{k} X_i, \qquad k \geq 1, \tag{3}$$

$$Z_k = \sum_{i=0}^{k} Y_i, \qquad k \geq 0, \tag{4}$$

Also let $H_k(\omega)$ denote the number of heads in the k-prefix of ω, i.e., the number of heads up to and including time k. We define $H_0(\omega) = 0$ for all ω. In this paper it will be useful to consider the **binomial lattice** where a point with coordinates (k, h) represents all paths ω with $H_k(\omega) = h$. We say that the value of a random variable (e.g., stock price S_k) at the point (k, h) is x if the value of the random variable at time k is x on a path ω such that $H_k(\omega) = h$, i.e., the value of the random variable at time k is x when there have been h heads up to and including time k.

The stock price at time k is denoted S_k, $k = 0, 1, \ldots, n$, and is given by

$$S_k = S_0 u^{X_1 + X_2 + \ldots + X_k} = S_0 u^{2H_k - k}.$$

Thus the stock price at point (k, h) is $S_0 u^{2h-k}$. The average stock price at time k is defined as

$$A_k = (S_0 + S_1 + \ldots + S_k)/(k + 1).$$

The payoff of an **Asian call** with strike L at time n is $(A_n - L)^+$ and the payoff of an **Asian put** is $(L - A_n)^+$. The payoff of a **floating-strike Asian call** at time n is $(A_n - S_n)^+$, and for a **floating-strike Asian put**, the payoff is $(S_n - A_n)^+$. In general we write W_n^+ to denote the payoff of any of the above options at time n. The **value** of such an option (at time 0) is defined as

$$V = \mathbf{E}(W_n^+)/R^n.$$

In this paper we are concerned with estimating $\mathbf{E}(W_n^+)$, and we will ignore the discounting factor R^n.

Let $\mathcal{F}$ denote the σ-algebra consisting of all subsets of Ω, and $\mathcal{F}_k$ denote the σ-algebra generated by the first k coin-tosses, i.e., by $X_1, \ldots, X_k$. We will always assume the probability measure **P** on $(\Omega, \mathcal{F})$ given by:

$$\mathbf{P}[X_k = 1] = p, \quad k = 1, 2, \ldots, n.$$

Thus for any random variable Z, $\mathbf{E}(Z|\mathcal{F}_k)$ denotes the conditional expectation of Z given the fist k coin tosses. A random variable X is **$\mathcal{F}_k$-measurable** if it depends only on the first k coin tosses. An $\mathcal{F}_0$-measurable random variable is non-random. A sequence of random variables $\{Z_k\}_{k=0}^n$ is called a **process**, and we will often refer to it simply as "the process $\{Z_k\}$". A process $\{Z_k\}$ is said to be **adapted** if each Z_k is $\mathcal{F}_k$-measurable. For instance the stock price process $\{S_k\}$ is adapted.

3 Approximations based on conditional expectations

In this and the remaining sections we describe algorithms for approximating the value of a European-style Asian call option in the binomial tree model. As mentioned in the introduction, all the algorithms are inspired by the continuous-time analysis of Rogers and Shi [10]. In this section we describe a general scheme for computing the approximation (1) and the error bound (2). In the next three sections we show how to apply this scheme to specific random variables Z.

In all of our approximations, the random variable we condition on will be a combination of H_n and one or more other random variables, which we collectively denote by Z. Our approximations (i.e., lower bounds) are therefore of the form

$$\mathbf{E}\left(\mathbf{E}(W_n|Z,H_n)^+\right), \tag{5}$$

where the random variable Z could be multi-dimensional. The reason we always condition on H_n is that all paths with the same value of H_n have the same probability; this allows us to reduce the computation of the expectation (5) to a path-counting problem.

To describe how to compute (5), we introduce some useful notation. Let $N(k,z)$ denote the number of paths with $H_n = k$ and $Z = z$. Equivalently, $N(k,z)$ is the number of paths with $Z = z$ that go through the lattice point (n,k). Let $M_1(k,z;\ t,h)$ denote the number of paths with $Z = z$ that go through (n,k) and (t,h). Let $M_2(k,z;\ t_1,h_1;\ t_2,h_2)$ be the number of paths with $Z = z$ that go through (n,k), (t_1,h_1) and (t_2,h_2). Let $C(k)$ denote the set of possible values of the random variable Z on paths with $H_n = k$. Then for an Asian call our approximation can be written

$$\begin{aligned}
&\mathbf{E}(\mathbf{E}(W_n \mid Z, H_n)^+)\\
&= \mathbf{E}(\mathbf{E}(A_n - L \mid Z, H_n)^+)\\
&= \mathbf{E}(\mathbf{E}(A_n \mid Z, H_n)^- L)^+\\
&= \sum_{k=0}^{n} \sum_{z\in C(k)} \mathbf{P}(Z=z, H_n=k)\Big[\mathbf{E}(A_n \mid H_n=k, Z=z) - L\Big]^+\\
&= \sum_{k=0}^{n} \sum_{z\in C(k)} p^k q^{n-k} N(k,z)\Big[\mathbf{E}(A_n \mid H_n=k, Z=z) - L\Big]^+\\
&= \sum_{k=0}^{n} \sum_{z\in C(k)} p^k q^{n-k}\Big[N(k,z)\mathbf{E}\left(A_n \mid H_n=k, Z=z\right) - N(k,z)L\Big]^+.
\end{aligned} \tag{6}$$

If the size of the sets $C(k)$ is bounded by a constant times $f(n)$, then the number of iterations of the two summations above is proportional to $n\ f(n)$. Note that

$$W(k,z) = N(k,z)\mathbf{E}\left(A_n \mid H_n=k, Z=z\right)$$

is the *sum* of A_n over all paths with $H_n = k$ and $Z = z$. When we write down the expression for this sum, we notice that the stock price at the lattice point (t,h) (which is $S_0 u^{2h-t}$) appears in this sum with a coefficient

$$\frac{M_1(k,z;\ t,h)}{(n+1)},$$

so we can write $W(k,z)$ as

$$W(k,z) = \frac{S_0}{(n+1)} \sum_{t=0}^{n} \sum_{h=0}^{t} M_1(k,z;\ t,h) u^{2h-t}. \tag{7}$$

Thus the approximation for the Asian call is given by the expressions (6) and (7). To complete the description of the approximations, we only need to show how to compute the quantities $N(k,z)$, $M_1(k,z;\ t,h)$ and the sets $C(k)$. In the following sections we do this for three specific random variables Z. Suppose the computation of each $M_1(k,z;\ t,h)$ takes time proportional to $T_1(n)$. Then since the time to compute each $M_1(k,z;\ t,h)$ dominates that required to compute $N(k,z)$, it is easy to see that the overall approximation takes time proportional to

$$n^3 f(n) T_1(n). \tag{8}$$

We now consider the computation of the error bound (2). Again in the case of an Asian call we can write the expectation in the error bound as

$$\begin{aligned} \mathbf{E}[\mathrm{var}(W_n \mid H_n, Z)^{\frac{1}{2}}] &= \mathbf{E}[\mathrm{var}(A_n - L \mid H_n, Z)^{\frac{1}{2}}] \\ &= \sum_{k=0}^{n} \sum_{z \in C(k)} \mathbf{P}(H_n = k, Z = z)[\mathrm{var}(A_n - L \mid H_n, Z)^{\frac{1}{2}}] \\ &= \sum_{k=0}^{n} \sum_{z \in C(k)} p^k q^{n-k} N(k,z)[\mathrm{var}(A_n - L \mid H_n, Z)^{\frac{1}{2}}] \end{aligned}$$

where

$$\mathrm{var}(A_n - L \mid H_n, Z) = \mathbf{E}[(A_n - L)^2 \mid H_n, Z] - [\mathbf{E}(A_n - L \mid H_n, Z)]^2. \tag{9}$$

We already know how to compute the second term in (9) above, and the first term is

$$\begin{aligned} \mathbf{E}[(A_n - L)^2 \mid H_n, Z] &= \mathbf{E}[A_n^2 - 2A_n L + L^2 \mid H_n, Z] \\ &= \mathbf{E}(A_n^2 \mid H_n, Z) + L^2 - 2L\mathbf{E}(A_n \mid H_n, Z), \end{aligned}$$

where we already know how to compute the last term, so we only need to show how to compute $N(k,z)\mathbf{E}(A_n^2 \mid H_n, Z)$. This is the *sum* of A_n^2 over all paths with $H_n = k$ and $Z = z$. When we write down this sum, we notice that there are two kinds of terms. The first kind are those involving the square of a stock price. The square of the stock price at lattice point (t,h) (which is $S_0^2 u^{2(2h-t)}$) appears in this sum with a coefficient

$$\frac{M_1(k,z;\ t,h)}{(n+1)^2}.$$

The remaining terms involve cross products of stock prices at different points on the lattice. The product of the stock price at (t_1,h_1) and (t_2,h_2), $t_1 \neq t_2$ (which is $S_0^2 u^{2h_1-t_1} u^{2h_2-t_2}$) appears with a coefficient

$$\frac{2M_2(k,z;\ t_1,h_1;\ t_2,h_2)}{(n+1)^2}.$$

Thus we may write

$$N(k,z)\mathbf{E}(A_n^2 \mid H_n) =$$

$$\frac{S_0^2}{(n+1)^2}\sum_{t=0}^{n}\sum_{h=0}^{t} M_1(k,z;\ t,h)u^{2(2h-t)}$$

$$+\,2\frac{S_0^2}{(n+1)^2}\sum_{t_1=0}^{n}\sum_{h_1=0}^{t_1}\sum_{t_2=t_1+1}^{n}\sum_{h_2=h_1}^{t_2} M_2(k,z;\ t_1,h_1;\ t_2,h_2)u^{2h_1-t_1}u^{2h_2-t_2}. \quad (10)$$

It is straightforward to check that the time required to compute the error bound is proportional to

$$n^5 f(n)T_2(n), \quad (11)$$

where the computation of $M_2()$ takes time proportional to $T_2(n)$.

In the following sections we show how the quantities $N()$, $M_1()$ and $M_2()$ can be computed efficiently, for various choices of Z.

4 Conditioning on the stock price at selected points.

For the purpose of illustration we consider first an approximation based on conditioning only on H_n, the number of heads (i.e., up-ticks) up to time n. In other words we consider the approximation (5) without the random variable Z. Accordingly we suppress the argument z in $N()$, $M_1()$ and $M_2()$. It is easy to see that

$$N(k) = \binom{n}{k},$$

$$M_1(k;\ t,h) = \binom{t}{h}\binom{n-t}{k-h},$$

$$M_2(k;\ t_1,h_1;\ t_2,h_2) = \binom{t_1}{h_1}\binom{t_2-t_1}{h_2-h_1}\binom{n-t_2}{k-h_2},$$

where, as in the rest of the paper, we interpret the binomial coefficient $\binom{t}{h}$ to be 0 if $t < 0$ or $h < 0$ or $t < h$. Each of these expressions can be computed in time proportional to n. Thus from expression (8) we see that the time required to compute this approximation is proportional to n^4. Also, from (11), the error bound can be computed in time proportional to n^6.

We now present a more general approximation, where we choose Z to be the combination of d random variables

$$H_{n_1}, H_{n_2}, \ldots, H_{n_d},$$

for some chosen values of $n_1 < n_2 < \ldots < n_d < n$. Thus the approximation for an Asian call is the right hand side of:

$$\mathbf{E}(A_n - L)^+ \geq \mathbf{E}\Big[(\mathbf{E}(A_n \mid H_{n_1}, H_{n_2}, \ldots, H_{n_d}, H_n) - L)^+\Big]. \quad (12)$$

It should be clear that we get increasingly good approximations as we use larger values of d. Of course this comes at the expense of an increased computation time, as we see below. The size of the sets $C(k)$, in expression (6), is proportional to $f(n) = n^d$. As mentioned before we only need to show how to compute the quantities $N()$ and $M_1()$ efficiently. Since the Z in this case is multi-dimensional, we write $N(k, h_1, \ldots, h_d)$ to denote the number of paths with $H_n = k$ and

$$H_{n_1} = h_1, H_{n_2} = h_2, \ldots, H_{n_d} = h_d.$$

The notation $M_1(k, h_1, h_2, \ldots, h_d;\ t, h)$ is defined similarly. It is straightforward that

$$N(k, h_1, \ldots, h_d) = \binom{n_1}{h_1}\binom{n_2 - n_1}{h_2 - h_1} \cdots \binom{n_d - n_{d-1}}{h_d - h_{d-1}}\binom{n - n_d}{k - h_d} \tag{13}$$

(Recall that we are interpreting the value of a binomial coefficient $\binom{k}{h}$ to be 0 if $h < 0, k < 0$ or $k < h$.) Also,

$$M_1(k, h_1, h_2, \ldots, h_d; t, h) =$$
$$\begin{cases} N(k, h_1, h_2, \ldots, h_e, h, h_{e+1}, \ldots, h_d) & \text{if } n_e \le t < n_{e+1} \text{ for some } 1 \le e \le d \\ N(k, h, h_1, \ldots, h_d) & \text{if } t < n_1 \\ N(k, h_1, h_2, \ldots, h_d, h) & \text{if } n_d \le t, \end{cases} \tag{14}$$

which represents the number of paths through $(n_1, h_1), (n_2, h_2), \ldots, (n_d, h_d), (n, k)$ *and* (t, h). The computation of this expression takes time proportional to $T_1(n) = dn$. From expression (8), the total time required to compute this approximation is proportional to

$$n^3 f(n) T_1(n) = dn^{d+4}.$$

The quantity $M_2(k, h_1, h_2, \ldots, h_d;\ t_1, h_1;\ t_2, h_2)$ can be computed by an expression similar to the one above, in time proportional to dn. Thus from (11), the error bound can be computed in time proportional to dn^{d+6}. We omit the tedious details.

5 Conditioning on the average of the underlying random walk.

We now consider the approximation (1) where we take Z to be the random variable Z_n, which is *sum* of the random-walk positions from time 0 to time n. Thus for an Asian call, our approximation is given by the right hand side of

$$\mathbf{E}(A_n - L)^+ \ge \mathbf{E}\Big[\mathbf{E}(A_n \mid H_n, Z_n) - L\Big]^+.$$

Note that the value of Z_n determines the *average* position of the underlying random walk (not the stock price) between time 0 and time n. This in turn determines the *geometric average* of the stock prices, which is simply $G_n = S_0 u^{Z_n/n}$. As mentioned in the introduction, an intuitive reason why conditioning on (H_n, Z_n) might yield a good approximation is this: paths with the same geometric stock-price average that end at the same stock price will likely have arithmetic stock-price averages

n	H_n	Z_n	Paths	G_n	Min. A_n	$\mathbf{E}(A_n \vert Z_n, H_n)$	Max. A_n	$\mathrm{var}(A_n \vert Z_n, H_n)$
8	6	18	4	115.191	115.575	115.832	116.346	0.09895
	4	0	8	100.000	100.111	100.167	100.334	0.00929
10	8	33	5	120.893	121.554	121.907	122.614	0.16226
	5	1	20	100.577	100.666	100.725	100.812	0.00404
12	10	52	6	125.978	126.983	127.416	128.283	0.22334
	6	0	58	100.000	100.077	100.150	100.488	0.00683

Figure 1: Illustrating the small variance of the arithmetic stock-price average A_n over the set of paths that have the same (a) geometric stock-price average G_n and (b) up-ticks H_n. The parameters assumed are $S_0 = 100,\ L = 90,\ \sigma = .2$, and $\mu = .05$. "Paths" indicates the number of paths with the indicated value of H_n and Z_n. "Min. A_n," "Max. A_n," and $\mathbf{E}(A_n|Z_n, H_n)$ respectively indicate the smallest, largest and average value of A_n over this set of paths; all these paths have the same value of G_n. $\mathrm{var}(A_n|Z_n, H_n)$ is the variance of A_n over this set of paths. For each n, the first line corresponds to the values of Z_n and H_n that give rise to the maximum variance of A_n; in the second line we choose $H_n = n/2$, and Z_n so that the number of paths in the set is maximized.

that do not differ too much. Figure 1 illustrates this point. In [10], Rogers and Shi consider an approximation based on conditioning on Z_n alone rather than both Z_n and H_n. As we show below, conditioning on both random variables has the advantage that it allows efficient computation in the binomial tree model. Moreover, the resulting approximation is strictly better than the one obtained by conditioning on Z_n alone.

To describe the computation of the quantities $N(), M_1()$ and $M_2()$ we introduce some useful notation. Let $C_i(k)$ denote the *set* of possible values of Z_i on paths such that $H_i = h$. In other words, $C_i(k)$ is the set of possible values of the random-walk sum Z_i on paths that go through the lattice point (i, k). Note that $C_n(k) = C(k)$. Also, let $N_i(k, z)$ denote the number of paths with $H_i = k$ and $Z_i = z$. Note that $N_n(k, z) = N(k, z)$. Since the maximum possible absolute value of Z_i is

$$1 + 2 + 3 + \ldots + i = i(i + 1)/2,$$

and Z_i is always an integer, there cannot be more than $i(i + 1)$ possible values of Z_i. Therefore the sum of the sizes of the sets $C_i(h)$ for $h = 0, 1, \ldots , i$ cannot exceed $i(i + 1)$. In particular, the size of $C_n(k) = C(k)$ is at most a constant times $f(n) = n^2$.

We now show that the quantities $N(), M_1()$ and $M_2()$ can be computed efficiently.

First we consider the quantities $N_i(h, z)$. We assume that we maintain a three-dimensional $n \times n \times n(n + 1)$ array to store the values $N_i(h, z)$: the first two dimensions correspond to i and h and the third dimension corresponds to the z. Note that for $i > 0$ all paths with $H_i = h$ have $Y_i = 2h - i$, and H_{i-1} is either $h - 1$ or h. In either case,

$$Z_{i-1} \ = \ Z_i - Y_i \ = \ Z_i - 2h + i.$$

Thus $N_i(h, z)$ satisfies the recurrence

$$N_i(h, z) \ = \ N_{i-1}(h - 1, z - 2h + 1) + N_{i-1}(h, z - 2h + 1).$$

Starting with $N_0(0, 0) = 0$, all the $N_i(h, z)$ can be computed inductively (or equivalently, by dynamic programming) using this recurrence. Since there are $n(n+1)/2$ points (i, h) in the binomial lattice, and each such point has number of possible Z_i values proportional to n^2, the computation of all the $N_i(h, z)$ takes time porportional to n^4. In the implementation, we first compute all $N_i(h, z)$ values, which are used in the computation of the M_1 quantities, as we now describe.

We now turn to the computation of the $M_1(k, z;\ t, h)$, the number of paths through (t, h), (n, k) that have $Z_n = z$. We can partition these paths according to their Z_t value. The number of such paths with $Z_t = s$ is equal to $N_t(h, s)$ times X where X is the number of paths between (t, h) and (n, k) such that $Y_{t+1} + Y_{t+2} + \ldots + Y_n = z - s$. Note that all paths through (t, h) have $Y_t = 2h - t$, so X is the same as the number of paths between $(0, 0)$ and $(n - t, k - h)$ such that

$$Z_{n-t} \;=\; z - s - (n-t)Y_t \;=\; z - s - (n-t)(2h-t).$$

Thus

$$M_1(k, z;\ t, h) = \sum_{s \in C_t(h)} N_t(h, s) N_{n-t}(k - h, z - s - (n-t)(2h-t)),$$

which can be computed in time proportional to $T_1(n) = n^2$, since the size of $C_t(h)$ is no more than some constant times t^2. Thus the total time required to compute this approximation is porportional to n^4 (for pre-computing the $N_i(h, z)$ values) plus expression (8), which is

$$n^3 f(n) T_1(n) = n^7.$$

Since n^7 dominates n^4, the time required is proportional to n^7.

We mention also that the quantity $M_2(k, z;\ t_1, h_1;\ t_2, h_2)$ can be computed using similar ideas, in time $T_2(n) = n^4$. Thus the error bound can be computed in time proportional to $n^5 f(n) T_2(n) = n^{11}$.

6 A hybrid algorithm

We now briefly describe an approximation based on conditioning on a random variable Z that is a combination of the two random variables described in the previous two sections. That is, we take Z to be the $(d+1)$-dimensional random variable

$$H_{n_1}, H_{n_2}, \ldots, H_{n_d}, Z_n.$$

This gives an approximation that is strictly better than either of the previous two approximations. While the details of the computation of the quantities $N()$,$M_1()$ and $M_2()$ are rather tedious, we illustrate the basic idea by means of the case $d = 1$. That is, we condition on the random variables

$$H_n, H_t, Z_n,$$

where t is a fixed integer lying strictly between 0 and n. Our approximation for an Asian call is the right hand side of

$$\mathbf{E}(A_n - L)^+ \geq \mathbf{E}\Big[\mathbf{E}(A_n \mid H_n, H_t, Z_n) - L\Big]^+.$$

We write $N(k, h, z)$ to denote the number of paths with $Z_n = z$ that go through (n, k) and (t, h). Also we write $M_1(k, h, z;\ t_1, h_1)$ to denote the number of paths with $Z_n = z$ that go through (t, h),

(n, k) and (t_1, h_1). Recall that $N(k, h, z)$ is exactly the quantity $M_1(k, z;\ t, h)$ that we computed in Section 5 where the random variable Z was simply Z_n.

Now we describe the computation of $M_1(k, h, z;\ t_1, h_1)$. We assume that $t < t_1$. The other cases can be handled similarly. By arguments similar to those used to compute $M_1()$ in the previous section, we see that

$$M_1(k, h, z;\ t_1, h_1) = \sum_{s \in C_t(h)} \sum_{s_1 \in C_{t_1}(h_1)} \Big[N_t(h, s) \times N_{t_1 - t}\left(h_1 - h,\ s_1 - s - (t_1 - t)(2h - t)\right) \times N_{n - t_1}\left(k - h_1,\ z - s_1 - (n - t_1)(2h_1 - t_1)\right) \Big], \quad (15)$$

which can be computed in time proportional to n^4 since each of the sets $C_t(h)$ and $C_{t_1}(h_1)$ has size at most some constant times n^2. For general d, the computation of $M_1()$ takes time at most a constant times $T_1(n) = n^{2(d+1)}$. Also, the size of the sets $C(k)$ in expression (6) is at most a constant times $f(n) = n^{d+2}$. So from expression (8), the time required to compute the approximation is proportional to

$$n^3 f(n) T_1(n) = dn^{3d+7}.$$

The computation of the quantity $M_2()$ can be carried out using similar ideas.

7 Numerical experiments

Throughout we assume that the option expires in time $T = 1$, the number of steps $n = 10$, and the initial stock price $S_0 = 100$. We vary the strike L, the volatility σ, and the drift μ. The results are displayed in Figures 2 and 3. In both tables, each parameter value is displayed only when it is different from the one in the previous line. For ease of comparison we use parameter values similar to those in Rogers and Shi [10].

Figure 2 shows a comparison of some of our approximations with the exact value of an Asian call in the binomial tree model. V_0 denotes the approximation of section 4 with $d = 0$, i.e., with conditioning on just H_n. V_3 denotes the approximation of that section with $d = 3$. More specifically, we condition on

$$(\ H_{n_1}, H_{n_2}, H_{n_3}, H_n\ ),$$

where we choose $n_i = i\lfloor n/4 \rfloor$, $i = 1, 2, 3$. V_{av} denotes the approximation of Section 5, i.e., we condition on the pair (Z_n, H_n). Finally, V denotes the *exact* value of the option in the binomial model given by $\mathbf{E}(A_n - L)^+$. In all cases we see that the agreement between V and V_{av} is excellent.

In Figure 3 we show some comparisons of our approximations with the Monte Carlo results of Levy and Turnbull [9]. Again we see that the agreement with V_{av} is very good.

Strike L	Vol σ	Drift μ	u	p	Apx V_0	Apx V_3	Apx V_{av}	Exact V
95	0.05	0.05	1.0159	0.6546	7.1769	7.1781	7.1783	7.1783
100					2.5941	2.6860	2.6939	2.6939
105					0.1637	0.2440	0.2515	0.2515
95		0.09		0.7819	8.8150	8.8151	8.8151	8.8151
100					4.2617	4.2859	4.2881	4.2881
105					0.6158	0.7365	0.7452	0.7452
95		0.15		0.9739	11.1115	11.1115	11.1115	11.1115
100					6.8080	6.8080	6.8080	6.8080
105					2.5091	2.5179	2.5184	2.5184
90	0.10	0.05	1.0321	0.5713	11.9375	11.9445	11.9451	11.9451
100					3.3319	3.5937	3.6159	3.6159
110					0.1485	0.2462	0.2552	0.2552
90		0.09		0.6350	13.3859	13.3883	13.3885	13.3885
100					4.6674	4.8616	4.8783	4.8783
110					0.3227	0.4726	0.4853	0.4853
90		0.15		0.7310	15.4151	15.4154	15.4154	15.4154
100					6.8971	6.9825	6.9901	6.9901
110					0.8727	1.0934	1.1098	1.1098
90	0.50	0.05	1.1713	0.4763	16.1753	17.2525	17.3243	17.3252
100					10.8140	12.1138	12.2208	12.2232
110					7.2794	8.3115	8.4067	8.4082
90		0.09		0.4890	17.0228	18.0257	18.0934	18.0943
100					11.5780	12.8308	12.9343	12.9365
110					7.9038	8.9217	9.0159	9.0173
90		0.15		0.5081	18.2776	19.1687	19.2301	19.2308
100					12.7439	13.9163	14.0138	14.0158
110					8.8794	9.8660	9.9577	9.9591

Figure 2: Comparison of approximations with the exact value of an Asian call in the binomial tree model

Strike L	Vol σ	Drift μ	V_0	V_3	V_{av}	Monte Carlo
95	0.10	0.09	8.86	8.90	8.91	8.91
100			4.66	4.86	4.89	4.91
105			1.75	1.94	1.96	2.06
90	0.30		14.45	14.90	14.93	14.96
100			7.93	8.70	8.77	8.81
110			3.77	4.54	4.60	4.68
90	0.50		17.02	18.03	18.09	18.14
100			11.58	12.83	12.93	12.98
110			7.90	8.92	9.02	9.10

Figure 3: Comparison with Monte Carlo results.

References

[1] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Political Economy*, 81:637–654, 1973.

[2] A. Carverhill and L. Clewlow. Flexible convolution. *RISK*, pages 25–29, April 1990.

[3] J. Cox, S. Ross, and M. Rubinstein. Option pricing: A simplified approach. *J. Financial Economics*, 7:229–264, 1979.

[4] H. Geman and M. Yor. Bessel processes, asian options and perpetuities. In *FORC Conference, Warwick, UK*, 1992.

[5] B. Heenk, A. Kemna, and A. Vorst. Asian options on oil spreads. *Review of Futures Markets*, 9(3):510–528, 1990.

[6] A. Kemna and A. Vorst. A pricing method for options based upon average asset values. *J. Banking and Finance*, pages 113–129, March 1990.

[7] E. Levy. Asian arithmetic. *RISK*, pages 7–8, May 1990.

[8] E. Levy. Pricing european average rate currency options. *J. Internat. Money Finan.*, 11:474–491, 1992.

[9] E. Levy and S. Turnbull. Average intelligence. *Risk Magazine*, Feburary 1992.

[10] L. Rogers and Z. Shi. The value of an asian option. *J. Appl. Prob.*, 32:1077–1088, 1995.

[11] A. Ruttiens. Classical replica. *RISK*, pages 5–9, Feb 1992.

[12] S. Turnbull and L. Wakeman. A quick algorithm for pricing european average options. *J. Financial and Quantitative Analysis*, 26(3):377–390, 1991.

[13] P. Wilmott, J. DeWynne, and S. Howison. *Option pricing: mathematical models and computation*. Oxford Financial Press, 1993.

[14] M. Yor. On some exponential functionals of brownian motion. *Adv. Appl. Prob.*, 1992.