Pricing Asian Options with an Efficient Convergent Approximation Algorithm

Tian-Shyr Dai¹, Guan-Shieng Huang², and Yuh-Dauh Lyuu³

¹ Corresponding author. Department of Applied Mathematics, Chung-Yuan Christian University, 22 Pu-Jen, Pu-chung Li, Chung-Li, Tao-Yuan county, Taiwan 320. E-mail: cameldai@cycu.edu.tw.
² Department of Computer Science & Information Engineering, National Chi Nan University, Taipei, Taiwan. E-mail: shieng@turing.csie.ntu.edu.tw
³ Department of Finance and Department of Computer Science & Information Engineering (preferred address), National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei, Taiwan. E-mail: lyuu0.csie.ntu.edu.tw The author was supported in part by NSC grant 92-2213-E-002-016.

Summary. Asian options are popular path-dependent derivatives in the financial market. However, how to price them efficiently and accurately has been a long-standing research and practical problem. No known exact pricing formulas are available to price the Asian option. Although approximate pricing formulas exist, they lack accuracy guarantees. Asian options can be priced on the lattice. A lattice divides a time interval into \( n \) equal-length time steps. It is known that the value computed by the lattice converges to the true option value as \( n \to \infty \). Unfortunately, only subexponential-time algorithms are available if Asian options are to be priced on the lattice without approximations. Efficient approximation algorithms are available for the lattice. The best known in the literature is an \( O(n^{3.5}) \)-time approximation lattice algorithm and an \( O(n^3) \)-time approximation PDE algorithm. Our paper suggests an \( O(n^{2.5}) \)-time lattice algorithm. Our algorithm uses a novel technique based on the method of Lagrange multipliers to minimize the approximation error. Numerical results verify the accuracy and the excellent performance of our algorithm.

1 Introduction

Derivative securities are financial instruments whose values depend on some underlying assets. Derivative securities are essential to speculation and the management of financial risk. Options are financial derivatives that give their buyers the right but not the obligation to buy or sell the underlying assets for a contractural price called the exercise price. A path-dependent option is an option whose payoff depends nontrivially on the price history of the underlying asset, which we will assume it to be stock for convenience. This paper focuses
on a particular type of path-dependent option, the Asian option, that is known to be difficult to price.

Asian options were originally traded on Asian markets, particularly in Tokyo [16]. The payoff of an Asian option depends on the average price of the underlying asset. It is 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. Hence the averaging feature is popular in many thinly-traded markets and embedded in many derivatives like convertible bonds.

There are no exact closed-form pricing formulas to price the Asian option. We call this price the true option value for brevity. Approximate closed-form solutions have been proposed under various assumptions. These formulae are surveyed by [7, 10, 16]. They show rather conclusively that most approximate closed-form formulas lack the accuracy guarantees and some even produce large pricing errors under certain circumstances. [15] gives a semi-analytical method that is very accurate and is generally considered “exact.”

Since no exact closed-form formulas exist for the Asian option, the development of efficient numerical algorithms becomes critical. The option value can be approximated by numerical methods such as the lattice and the discretized PDE methods. These models divide the time horizon of the option into \( n \) discrete time steps and discretize the stock prices at each time step. Take a 2-time-step CRR lattice in Fig. 1 as an example. (The details of the CRR lattice will be described later.) The time interval is evenly divided into 2 time steps. The stock price at time step 0 is \( S_0 \) (at node \( N(0, 0) \)). The stock price can either move up to \( S_0u \) (at node \( N(1, 0) \)) or down to \( S_0d \) (at node \( N(1, 1) \)) at the first time step. Similarly, each stock price can either move up or move down again in the following time steps. Discretization error is introduced because both the time and the possible stock prices are discretized. Since the discretization error goes to zero at rate \( O(n^{-1}) \) [6], the option values computed by the CRR lattice converge to the true option value. The remaining key issue is whether such convergence can be achieved efficiently.

To price an Asian option on a lattice without introducing errors other than the discretization error, the most efficient algorithm currently known runs in time subexponential in \( n \) [5]. To see intuitively why pricing the Asian option on the lattice can be so time-consuming, assume that the random walk for the stock price is binomial as in Fig. 1. After \( n \) time steps, the price history contains \( 2^n \) possible paths, each with its own average stock price. As the payoff of the Asian option depends on the average stock price, there are \( 2^n \) possible payoffs at time step \( n \). The average stock price of a price path starting from the root and ending at an arbitrary node \( N(i, j) \) will be called a state for \( N(i, j) \).

To strike a better balance between efficiency and accuracy, the approximation algorithm proposed by Hull and White reduces the number of states on each node in the lattice by selecting some representative states for each node (called allocated states) [8]. To be more specific, only the option value corre-
sponding to an allocated state is evaluated. The option value corresponding to a missing state is interpolated from those of the two nearest allocated states. Interpolation errors are thus introduced. This influential paradigm has been followed by most approximation lattice algorithms [2, 9, 13]. The major problem with the Hull-White paradigm is convergence: Improperly implemented, the calculated option values may not converge to the true option value [6]. Efficient and convergent approximation algorithms on the lattice are available. Forsyth, Vetzal, and Zvan present an $O(n^{3.5})$-time approximation lattice algorithm and an $O(n^3)$-time discretized PDE method that both converge at rate $O(n^{-1})$ [6].

The major contribution of our paper is a new approximation lattice algorithm with a running time of $O(n^{2.5})$ and a convergence rate of $O(n^{-1})$. Two key ideas are exploited to achieve this goal. First, the option values of many states are evaluated exactly by a simple formula without resorting to interpolation. This dramatically reduces the interpolation errors accumulated during backward induction. The algorithm therefore focuses the computational efforts on the states that can not be evaluated directly. This idea is pioneered in [1]. The second idea is to allocate the number of states in such a way that the interpolation error can be minimized. Intuitively, the states are distributed based on the importance of each node. In our proposed algorithm, the number of states allocated at each node is calculated by applying the method of Lagrange multipliers to minimize the interpolation error.

The paper is organized as follows. The stock price dynamics is described in Section 2. How to price Asian options on the lattice and the efficiency
problems are also dealt with in the same section. Section 3 presents our efficient approximation algorithm and proves the performance and convergence rate claims. Numerical results are given in Section 4 to support these claims. Section 5 concludes this paper.

2 Model, Lattice, and Pricing

Assume that the Asian option initiates at 0 (in year) and matures at $T$ (in year). Define $S(t)$ as the stock price at year $t$. $S(t)$ follows the log-normal diffusion process:

$$S(t + dt) = S(t) \exp[(r - 0.5\sigma^2) dt + \sigma dW_t],$$

where $W_t$ is the standard Wiener process, $r$ is the risk-free interest rate per annum, and $\sigma$ denotes the volatility of the stock price. Note that both $r$ and $\sigma$ are positive constants in the model.

The payoff of an Asian option depends on the average stock price defined as $A_T \equiv \frac{\int_0^T S(r)dr}{T}$. Let $X$ be the exercise price. The payoffs of an Asian call and an Asian put at maturity date are defined as $\max(A_T - X, 0)$ and $\max(X - A_T, 0)$, respectively. The option’s value is equal to the discounted expected payoff of this option at the maturity date. Thus the values of an Asian call and an Asian put are therefore $e^{-rT}\mathbb{E}[\max(A_T - X, 0)]$ and $e^{-rT}\mathbb{E}[\max(X - A_T, 0)]$, respectively. There are no exact, simple closed-form solutions to the option values. Efficient discrete-time approximation methods are thus necessary.

A discrete-time approximation model partitions the time between year 0 and year $T$ into $n$ equal-length time steps. The length of a time step $\Delta t$ is therefore $T/n$. Let $S_i$ denote the stock price at time step $i$, which corresponds to $S(i\Delta t)$ in the continuous-time model. The average stock price from time step 0 to time step $i$ in a discrete-time model is defined as $A(i) \equiv \frac{S_0 + S_1 + \ldots + S_i}{i+1}$. The payoff of an Asian option at maturity in the discrete-time model is

$$\text{payoff} = \begin{cases} 
\max(A(n) - X, 0) & \text{for an Asian call,} \\
\max(X - A(n), 0) & \text{for an Asian put.}
\end{cases}$$

Thus the option value evaluated in the discrete-time model is $e^{-rT}\mathbb{E}[\text{payoff}]$. Our task is to compute $e^{-rT}\mathbb{E}[\max(A(n) - X, 0)]$ (or $e^{-rT}\mathbb{E}[\max(X - A(n), 0)]$) so that the values converge to the true option values $e^{-rT}\mathbb{E}[\max(A_T - X, 0)]$ (or $e^{-rT}\mathbb{E}[\max(X - A_T, 0)]$) as $n$ increases. This paper focuses on calls; the extension to puts is straightforward.

Pricing on the CRR Lattice

The CRR lattice model is due to Cox, Ross, and Rubinstein [4]. In the CRR lattice model, $S_{i+1}$ equals $S_i u$ with probability $p$ and $S_i d$ with probability
1 – p, where d < u. The stock price at time step i that results from j down moves and i – j up moves therefore equals $S_0 u^{i-j} d^j$. A 2-time-step CRR lattice is depicted in Fig. 1(a).

Let node $N(i, j)$ stand for the node at time step i reachable from the root with j cumulative down moves. Its associated stock price is $S_0 u^{i-j} d^j$. The stock price can move from $N(i, j)$ to $N(i + 1, j)$ with probability p and to $N(i + 1, j + 1)$ with probability 1 – p. Node $N(i, j)$ can be reached with probability $\binom{i}{j} p^j (1 – p)^i$. See Fig. 1(b) for illustration. For pricing purposes, the probability p for an up move is set to $(e^{r\Delta t} – d)/(u – d)$, where r is the risk-free interest rate, $u = e^{\sigma \sqrt{\Delta t}}$, and $d = 1/u$.

Pricing on the CRR lattice is done by backward induction. The option value at the maturity date can be evaluated directly by Eq. (2). Let $(i, j, A)$ denote a state with an average stock price $A$ (from time step 0 to time step i) at node $N(i, j)$ and $v(i, j, A)$ the corresponding option value. If this stock price moves up to node $N(i + 1, j)$ at time step $i + 1$, the average stock price becomes $A' \equiv \frac{(i+1)A + Su^{i+1-j}d^j}{i+2}$. If the stock price moves down to node $N(i + 1, j + 1)$, the average stock price becomes $A'' \equiv \frac{(i+1)A + Su^{i-j}d^{j+1}}{i+2}$. The option value $v(i, j, A)$ equals

$$v(i, j, A) = e^{-r\Delta t} [p \times v(i + 1, j, A') + (1 – p) \times v(i + 1, j + 1, A'')] \quad (3)$$

The above formula can be applied inductively from time step $n – 1$ to time step 0 with $v(0, 0, S_0)$ at the root node giving the pricing result.

**Interpolation**

Although the option value computed by the lattice model converges to the true option value, how the pricing problem can be solved efficiently poses a challenge. There are $\binom{d}{i-j}j!$ price paths that reach node $N(i, j)$, and each such path gives rise to a distinct average price (state). The number of states therefore rises dramatically, making the computation via Eq. (3) very time consuming. To address this problem, we follow the Hull-White paradigm by lowering the number of states at each node. When state $(i, j, A)$ is missing, its corresponding option value will be estimated by linear interpolation from its two nearest allocated states $(i, j, A^-)$ and $(i, j, A^+)$ via:

$$v(i, j, A) \approx \frac{A - A^-}{A^+ - A^-} v(i, j, A^+) + \frac{A^+ - A}{A^+ - A^-} v(i, j, A^-) \quad (4)$$

where $A^- < A < A^+$. The term “interpolation error” denotes the error from estimating $v(i, j, A)$ by linear interpolation.

**3 The New $O(n^{2.5})$-Time Pricing Algorithm**

Two key techniques are exploited by the algorithm. First, $v(i, j, A)$ can be evaluated directly when $A$ exceeds a certain easily calculated bound. This
result helps reducing the state count. Second, a state allocation scheme is
developed by applying the Lagrange’s multiplier to minimize the interpolation
error. The final state count turns out to be \( O(n^{2.5}) \), which is also the running
time.

**Pruning Unnecessary States**

The corresponding price sum for state \((\ell, m, A)\) is \((\ell + 1)A\), as \(A\) is the average
stock price from time step 0 to time step \(\ell\). The following theorem states that
\(v(\ell, m, A)\) can be described by a simple formula when \((\ell + 1)A \geq (n + 1)X\).

**Theorem 1.** Suppose the price sum associated with state \((\ell, m, A)\) is \((n + 1)X + \epsilon\) for some \(\epsilon \geq 0\). Then the option value \(v(\ell, m, A)\) is equal to

\[
\begin{align*}
& e^{-rT} \left[ \epsilon + S_0 u^{\ell-m} d^m e^{r\Delta t} \frac{1 - e^{(n-\ell)r\Delta t}}{1 - e^{r\Delta t}} \right] / (n + 1) \\
& \quad \text{Proof.} \quad \text{The expected value of the future price sum } S_{\ell+1} + S_{\ell+2} + \cdots + S_n \\
& \quad \text{equals} \\
& S_0 u^{\ell-m} d^m \left[ e^{r\Delta t} + e^{2r\Delta t} + \cdots + e^{(n-\ell)r\Delta t} \right] = S_0 u^{\ell-m} d^m e^{r\Delta t} \frac{1 - e^{(n-\ell)r\Delta t}}{1 - e^{r\Delta t}}. \\
& \quad \text{The option value } v(\ell, m, A) \text{ therefore equals} \\
& e^{-rT} E \left[ \max (A_{\text{avg}}(n) - X, 0) \left| \sum_{i=0}^{\ell} S_i = (n + 1)X + \epsilon, S_\ell = S_0 u^{\ell-m} d^m \right. \right] \\
& = e^{-rT} \left[ \epsilon + S_0 u^{\ell-m} d^m e^{r\Delta t} \frac{1 - e^{(n-\ell)r\Delta t}}{1 - e^{r\Delta t}} \right] / (n + 1). \\
& \quad \text{Now that the value } v(\ell, m, A) \text{ can be calculated without introducing
interpolation errors if } A \geq (n + 1)X, \text{ a pricing algorithm only needs to evaluate}
\text{v} (\ell, m, A) \text{ for } A < (n + 1)X. \text{ This improves efficiency by pruning unnecessary
states and accuracy for not resorting to interpolation.}

**The State Allocation Scheme**

Let \(k_{i,j}\) stand for the number of states allocated at node \(N(i, j)\). Define \(k\) as the average number of states for each node. The total number of states is
equal to \(\sum_{0 \leq j \leq i \leq n} k_{i,j} \approx n^2k/2\) as there are approximately \(n^2/2\) nodes. The running
time is therefore \(O(n^2k)\). Theorem 1 says that \(v(i, j, A)\) can be easily
evaluated if \((i + 1)A > (n + 1)X\). Thus at node \(N(i, j)\), all \(k_{i,j}\) states have
average stock prices not more than \((n + 1)X/(i + 1)\). These \(k_{i,j}\) states divide
the range \([0, (n+1)X/(i+1)]\) evenly. The difference of the average stock prices
of two adjacent states at node \(N(i, j)\) is \(\frac{(n+1)X}{(i+1)k_{i,j}} \leq \frac{nX'}{dk_{i,j}}, \text{ where } X' \equiv 2X.\)
The state allocation scheme introduces interpolation error because only $k_{i,j}$ states are allocated for $N(i, j)$ instead of the full $\frac{1}{2}(i-j)^2$ states and because interpolation formula (Eq. (4)) is employed. The interpolation error can be analyzed as follows. When we calculate $v(i, j, A)$ with Eq. (3), the option values for the nonexistent states $(i+1, j, A')$ and $(i+1, j+1, A'')$ are estimated by interpolation as in Eq. (4). Thus $v(i+1, j, A')$ is interpolated from the values of the two bracketing states $(i+1, j, A'_-)$ and $(i+1, j, A'_+)$, where $A'_- < A' < A'_+$. The interpolation error can be estimated by the Taylor series expansion:

$$\frac{(A'_+ - A')(A'_+ - A'_-)}{2} \frac{\partial^2 V(\eta)}{\partial A^2},$$

where $\frac{\partial^2 V(\eta)}{\partial A^2}$ denotes the second partial derivative of the option value $V$ with respect to the average price $A$ and $\eta = [A'_-, A'_+]$. We follow Forsyth, Vetzal, and Zvan (2002) in postulating that $|\frac{\partial^2 V(\eta)}{\partial A^2}|$ is bounded by a constant $M$.

The interpolation error for estimating $v(i+1, j, A')$ is therefore bounded by $M(nX'/ik_{i+d,j})^2$. Similarly, the interpolation error for $v(i+1, j+1, A'')$ is bounded by $M(nX'/ik_{i+d,j+d})^2$. Thus the accumulated interpolation error $E_a(i, j, A)$ for $(i, j, A)$ is bounded above inductively by

$$pM(nX'/ik_{i+d,j})^2 + (1 - p)M(nX'/ik_{i+d,j+d})^2$$

$$+ pE_a(i+1, j, A') + (1 - p)E_a(i+1, j+1, A'').$$

The accumulated interpolation error at $(0, 0, S_0)$ is bounded by

$$E_{\text{int}} = \sum_{0 \leq j \leq i \leq n} (i^j) p^{i-j}(1 - p)^j M(nX'/ik_{i,j})^2 = X'^2M \sum_{i=d}^n \sum_{j=0}^i \frac{B(i, j; p)n^2}{i^2k_{ij}^2},$$

where $B(i, j, p) \equiv (i^j) p^{i-j}(1 - p)^j$. To minimize $E_{\text{int}}$ subject to the condition $\sum_{0 \leq j \leq i \leq n} k_{i,j} = n^2k/2$, $k_{i,j}$ is solved by the Lagrange’s multipliers as

$$k_{i,j} = \frac{n^2k}{2} \times \frac{(B(i, j; p)/i^2)^{d/3}}{\sum_{0 \leq m \leq l \leq n}(B(l, m; p)/l^2)^{d/3}}.$$

The minimized $E_{\text{int}}$ is then

$$X'^2M \sum_{0 \leq j \leq i \leq n} \frac{B(i, j; p)n^2}{i^2n^4k^4/4} \times \frac{[n^2B(i, j; p)/i^2]^{2/3}}{[\sum_{0 \leq m \leq l \leq n}[n^2B(l, m; p)/l^2]^{1/3}]}$$

$$= \frac{4X'^2M}{n^4k^2} \left\{ n^{2/3} \sum_{0 \leq i \leq n} \left[i^{-2/3} \sum_{0 \leq j \leq i} B(i, j; p)^{d/3}\right] \right\}^{3}.$$

A result of Bender (1974), p. 489 implies that
\[
\sum_{0 \leq j \leq i} B(i, j; p)^{1/3} \leq \sum_{0 \leq j \leq i} B(i, j; 0.5)^{1/3} \sim (1/2)^{1/3} 3^{1/2} \pi^{1/3} i^{1/3} = b i^{1/3},
\]

where \(b = (1/2)^{1/3} 3^{1/2} \pi^{1/3}\). Since

\[
n^{2/3} \sum_{0 \leq i \leq n} i^{-2/3} \times b i^{1/3} \sim n^{2/3} \int_0^n b^{-1/3} d i = (3/2) b n^{4/3},
\]

\(E_{\text{int}}\) is bounded above by

\[
\frac{4X^2 M b^3 (3/2)^3 n^4}{n^4 k^2} = \frac{4X^2 M b^3 (3/2)^3}{k^2} = O(k^{-2}).
\]

Note that the lattice discretization error introduced by discretizing both the time and the possible stock prices converge at rate \(O(n^{-1})\) [6]. To ensure that the convergence rate of our algorithm is \(O(n^{-1})\), we let \(E_{\text{int}} = O(n^{-1})\) by setting \(k\) to be proportional to \(n^{0.5}\). As \(n^2 k = O(n^{2.5})\), our proposed algorithm runs in time \(O(n^{2.5})\).

4 Numerical Results

When the exercise price of the Asian option is zero, a closed-form solution exists for the Asian option [2]. The simple numerical test in Fig. 2 highlights that our algorithm converges at rate \(O(n^{-1})\). Observe that our prices lie essentially on a straight line. The limiting price when \(n\) tends to infinity can be estimated by extrapolation. The extrapolation value 98.760352 is extremely close to the true option value 98.76035189.

The following numerical test demonstrates that the extrapolated values obtained by our method are accurate. Zhang (2001) provides a very accurate semi-analytical model for pricing the Asian option. Zhang (2003) uses this method as the benchmark and compares many different pricing methods. His numerical data are repeated in Table 1 with the extrapolated option values computed by our lattice method added. Both the root-mean-squared errors and the maximum absolute errors of our extrapolated results are much less than those of other methods. Our method generates much lower pricing errors than other methods when the volatility is large.

5 Conclusion

This paper proposes a new approximation algorithm for pricing Asian options on a lattice. Our algorithm runs in time \(O(n^{2.5})\) with the convergence rate \(O(n^{-1})\), which is superior to existing lattice algorithms with the same convergence rate. Our claims are proved rigorously, and numerical results are provided to support the performance and convergence claims.
Fig. 2. Estimating the True Option Value by Extrapolation. The stock price is 100, the exercise price is 0, the risk-free rate is 10%, the time to maturity is 0.25 year, and the volatility of the underlying stock is 0.1. The \( x \)-axis denotes \( 1/n \). The \( y \)-axis denotes the option value. Four pricing results \((n = 50, 100, 200, 400)\) are marked with squares with the calculated option values listed next to the squares. The straight line is computed by linear regression.

Table 1. Comparison with Analytical Approximations.

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( X )</th>
<th>Exact</th>
<th>Extrapolation</th>
<th>AA3</th>
<th>J-TE</th>
<th>PM-J3</th>
<th>PM-J4</th>
<th>CT-GC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>95</td>
<td>8.8088392</td>
<td>8.808871</td>
<td>8.80884</td>
<td>8.80884</td>
<td>8.80884</td>
<td>8.80884</td>
<td>8.80884</td>
</tr>
<tr>
<td>0.1</td>
<td>95</td>
<td>8.9118509</td>
<td>8.911908</td>
<td>8.91184</td>
<td>8.91190</td>
<td>8.91175</td>
<td>8.91186</td>
<td>8.91183</td>
</tr>
<tr>
<td>0.3</td>
<td>95</td>
<td>11.6558858</td>
<td>11.655841</td>
<td>11.65518</td>
<td>11.65565</td>
<td>11.65663</td>
<td>11.65500</td>
<td>11.65475</td>
</tr>
<tr>
<td>105</td>
<td>8.8287588</td>
<td>8.828706</td>
<td>8.82900</td>
<td>8.82886</td>
<td>8.83183</td>
<td>8.82792</td>
<td>8.82755</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>95</td>
<td>15.4127163</td>
<td>15.442573</td>
<td>15.44557</td>
<td>15.43806</td>
<td>15.44623</td>
<td>15.43448</td>
<td>15.43707</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.000101</td>
<td>0.000129</td>
<td>0.00434</td>
<td>0.00561</td>
<td>0.00339</td>
<td>0.00268</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAE</td>
<td>0.000225</td>
<td>0.00315</td>
<td>0.01231</td>
<td>0.01621</td>
<td>0.00824</td>
<td>0.00587</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exact is the option value obtained in [15]; Extrapolation is the extrapolated value computed by our method; AA3 is the fourth-order approximation method given in [16]; J-TE is the Taylor expansion method given in [10]; PM-J3 is the shifted lognormal fitting method in [12]; PM-J4 is the shifted arcsinh-normal fitting method in [11]; GT-GC is the continuous limits of the geometric conditioning method given in [14]; RMSE is the root-mean-squared errors. MAE is the maximum absolute error.
References