Edwin H. Neave

George L. Ye

School of Business
Queen's University
Kingston, Ontario

Neave: neave@qsilver.queensu.ca
Phone: 613-545-2348
Fax: 613-545-2321

Ye: 4lsy@qsilver.queensu.ca
Phone: 416-325-0823
Fax: 416-204-7933
VALUATION OF ARITHMETIC AVERAGE CALLS:
FURTHER RESULTS

Edwin H. Neave
George L. Ye

Abstract

Revised December 3, 1998

This paper extends the generating function approach to valuing arithmetic average fixed strike calls on a recombining binomial process. Neave (1997) and Neave-Stein (1998) define a forward induction procedure that gives both lower bounds on value and a means of finding exact solutions for European options. This paper introduces a new forward induction procedure that bounds the values of European and American options from both above and below. The new bounds, which can be tightened at the user’s discretion, provide more nearly accurate valuations than other methods, especially when the underlying processes have relatively high volatilities. The solutions can be found with a relatively high degree of accuracy using just a PC and no more than a couple of minutes computing time. Since the main equipment limitation is available RAM, improved accuracy and even faster times can be obtained by installing our C++ programs on a workstation. In sum, the new methods provide practical solutions to both the European and the American valuation problems.

A Gaussian generating function helps to minimize calculations and to improve solution accuracy. First, the generating function suggests a natural way of organizing the data into sets called sub-bundles. Second, a sub-bundle may contain many paths, but all its paths are defined by a single geometric average, and attain a relatively small number of distinct arithmetic averages. Third, for valuations we only need the mean of the path arithmetic averages in a sub-bundle - unless the sub-bundle’s arithmetic averages fall both above and below the strike price, in which case we say it is divided. To find either an exact solution or a refined approximation, we need to determine, or approximate as the case may be, the frequency distribution of the sub-bundle’s distinct arithmetic averages. The sub-bundles which need this further investigation are easily identified and relatively few in number. Fourth, if we have enough RAM to store all the distinct arithmetic averages in the sub-bundles, our forward induction obtains an exact valuation. Fifth, if there are too many distinct averages to be stored individually, we classify them by intervals so as to create approximate frequency distributions from which upper and lower bounds on the solution can be calculated. If we need greater accuracy we introduce more classification intervals, although doing so increases computation time.

—350—

1 This is a working paper intended for limited circulation. It should not be cited or quoted without written permission of the authors.
VALUATION OF ARITHMETIC AVERAGE CALLS:
FURTHER RESULTS

Edwin H. Neave
George L. Ye

1. Introduction
   1.1 Review of Literature
   1.2 The Process
   1.3 Problem Formulation
2. Theoretical Results
   2.1 Sub-Bundle Means Give Lower Bound on Value
   2.2 Recursive Estimation of Arithmetic Averages Can Be Used for Upper Bound.
3. Computational Results: European Options
   3.1 Accuracy and Computation Time
   3.2 Accuracy of Competing Methods
   3.3 Accuracy and Volatility
   3.4 Stochastic Volatility Options
4. Computational Results: American Options
5. Conclusion
VALUATION OF ARITHMETIC AVERAGE CALLS:
FURTHER RESULTS

Edwin H. Neave
George L. Ye

1. INTRODUCTION

Neave (1997) uses a Gaussian generating function to structure the valuation of arithmetic average options defined on a recombining binomial asset process. Neave and Stein (1998) show that good approximations to the value of European options can be obtained.

This paper extends the Neave and Stein (1998). First, we introduce a new forward induction method\(^1\) that both refines the earlier approximations and permits studying the tradeoff between approximation error and computation time. Second, we show how our methods extend to valuing American options. Unlike most other treatments, this paper controls any approximation error. Third we provide evidence that other methods of approximation are likely to be relatively accurate only if the underlying asset price process has low volatility.

Our methods apply to a variety of options, but this paper focuses on fixed strike average calls. The rest of this section specifies the asset price process, defines the options, and outlines the valuation methods. Section 2 elaborates some of the underlying theory. Section 3 reports computational results for European options, Section 4 for American options. Section 5 concludes.

1.2 The Process

We use a recombining binomial price process derived from an underlying Bernoulli process \( \{X_t\}, t = 1, 2, \ldots, T \). Let \( \{X_t\} \) be independent, identically distributed random variables:

\[
X_t = \begin{cases} 
0; & p \\
1; & 1 - p 
\end{cases}
\]  \hspace{1cm} (1.1)

Next, let

\[
N_t = \sum_{s=1}^{t} X_s, t = 1, 2, \ldots, T, \quad (1.2)
\]

\(^1\)Hull and White (1993) and Chalasani et al. (1998), who also use forward induction methods, reduce the numbers of state variable values using linear interpolation. We use properties revealed by the generating function to manage the number of state variable values while obtaining greater accuracy.
and

\[ J_t^* = 2N_t - t; t = 1, 2, \ldots, T. \]  

(1.3)

Henceforth the \( J_t \) are called \textit{index variables}, and the realized values of the \( J_t \), \textit{node indices}.

The asset price process \( \{ S_t \}, \ t \in \{ 1, 2, \ldots, T \} \), is defined by:

\[ S_t = u^{J_t}; \ t = 1, 2, \ldots, T \]  

(1.4)

with \( S_0 = 1 \) and \( u > 1 \). Hereafter, the time \( t \) realized prices and their corresponding indices are called path \textit{end points} and path \textit{end indices} respectively.

We define the process' geometric averages using the entire price history:

\[
C_t = \left[ \prod_{s=0}^{t} S_s \right]^{1/(t+1)}
\]

\begin{align*}
= & \prod_{s=0}^{t} \left( u^{J_s} \right)^{1/(t+1)} \\
= & \left( u^{V_t/(t+1)} \right).
\end{align*}

(1.5)

where

\[ V_t = \sum_{s=0}^{t} J_s, \]  

(1.6)

t = 1, 2, \ldots, T. \] The realized values of the \( V_t \) are called \textit{index sums}. Values of the \( C_t \) can be determined from the \( V_t \) as indicated by (1.5). The arithmetic average is also defined over the entire price history:

\[
A_t \equiv \left[ \sum_{s=0}^{t} S_s \right]/(t+1) = \left[ \sum_{s=0}^{t} u^{J_s} \right]/(t+1).
\]  

(1.7)
The realized values of the \( A_s \), can be determined from sequences of the realized values of \( J_s \), \( s = 0, \ldots, t \).

1.3 Problem Formulation

The payoff to a European fixed strike average call with exercise date \( T \) is

\[
( B_T - K )^+
\]

where \( B_T \) is a process average and \( X^+ = \max \{ X, 0 \} \). Taking \( B_T = C_T \) in (1.8) defines the call payoffs for the geometric average, taking \( B_T = A_T \) defines them for the arithmetic average. Given an equivalent martingale measure \( Q \), the time zero values of the options are

\[
G_T^Q(K) = R^{-T} E_Q( B_T - K )^+
\]

where \( E_Q \) denotes expectation under \( Q \) and \( R^t = (1 + r)^t \) indicates the \( t \)-period accumulation of \$1\) at the constant, single-period risk-free interest rate \( r \).

To see how computational difficulties arise, write (1.9) as

\[
G_T^Q(K) = R^{-T} \sum_{j=0}^{T} P_j q^{T-j} \sum_{i=1}^{T(T-j)/2} \left( b_{2j-T} - K, 0 \right)^+
\]

where:

\( B \in \{ A_T, C_T \} \) indicates the type of average;
\( R = 1 + r \) is a discount factor, determined by a constant interest rate \( r \),
\( p, q = 1 - p; 0 < p < 1 \) indicate the martingale measure \( Q \), and
\( b_{2j-T} \) is a realized value of the average \( B \).

Since each of the right-hand summations in (1.10) involves \( T!/(T-j)!j! \) terms, the total number of paths is exponential:

\[
\sum_{j=0}^{T} \binom{T}{j} = 2^T
\]

Each realized average \( b_{2j-T} \) may be attained by many more than one of the \( T!/(T-j)!j! \) paths ending at \( 2j-T, j = 0, \ldots, T \). To avoid calculating individual paths, we use a Gaussian generating function to help organize problem data; cf. Neave (1997) and Neave and Stein (1998). This paper develops a new forward induction for valuing European options, as well as a new backward induction for valuing American options. Following Neave (1997), we employ Markovian formulations whose numbers of state variable values are much smaller than the
number of individual paths.

The set of all possible paths with a given end point is called a bundle, denoted $B(t, j)$. The subset of all paths in $B(t, j)$ whose indices add to $v$ is called a sub-bundle, and denoted $B(t, j, v)$. By definition, the paths in a sub-bundle all attain the same geometric average. Our valuations are based on organizing the $2^T$ paths of the problem into $T(T+1)/2 + 1$ sub-bundles, then rewriting (1.10) as

$$
\Gamma_T^2 (K) = R^{-T} \sum_{j=0}^{T} p^j q^{r-j} \sum_{v=V_{j}}^{V_{j},v} \{ c_{T, j, T, v} - K, 0 \}^+ g_{T, j, T, v},
$$

where in addition to previously used variables:

$c_{T,v}$ is a sub-bundle geometric average (a realized value of $C_T$), i.e.
$c_{T, v} = u^{T} \cdot v$ a component value of $v_{T, j}$,
$v_{T, j}$ and $v_{T, j}^{+}$ are extremal values of $v_{T, j}$, and
$g_{T, j, v}$ is the number of paths in $B(T, j, v)$.

Formulation (1.12) repeats some of the component values of $v_T$, but even with this duplication the number of distinct records needed for exact valuation of the geometric average call is

$$
\sum_{j=0}^{T} \left\lfloor \frac{j(T - j) + 1}{T + 1} \right\rfloor = (T^3 + 5T + 6)/6.
$$

To help interpret the approach, Figure 1 shows the relations between a bundle, its sub-bundles, and the behaviour of path arithmetic averages within sub-bundles. Each cell represents a path, and each (horizontal) bar of cells represents a sub-bundle. The length of a bar gives the number of paths in the sub-bundle; the cell height indicates the arithmetic average of that path. The different cell heights within a bar indicate the distinct arithmetic averages attained by the paths in a sub-bundle.

Apart from the values of the arithmetic averages, the remaining information conveyed by the graph can be obtained analytically, and all these features are invariant with respect to volatility. Representing the fixed strike price by a plane parallel to the $x$-$y$ plane of Figure 1 suggests how valuation can be simplified. First, recall that each sub-bundle attains only a single geometric average, which shows why valuing the geometric average instrument is a relatively easy task. Building on this observation, the sub-bundle means of the arithmetic averages would appear to offer a good approximate solution for the arithmetic average instrument. Second, using

---

3 In Figure 1, the different averages are attributable to process volatility of $\sigma = 0.80$. 

---
sub-bundle means would only introduce error if the strike price plane actually lay between
different altitudes in a sub-bundle, and such conditions would appear to arise relatively
infrequently. Third, the methods of linear interpolation used by Hull and White (1993) and more
recently by Chalasani et. al. (1998) appear capable of introducing errors whose magnitude
increases with volatility. All three of these conjectures are supported by the results we report
below.

Since by definition all paths in \( B(t, j, v) \) have the same probability of occurrence, the
mean of the paths’ arithmetic averages is

\[
m_{t,j,v} = \left( \frac{1}{n} \sum_{i=1}^{n} m_{t,j,v} + \ldots + n_g m_{t,j,v} \right) / n,
\]

where \( n_i \) is the number of paths attaining the arithmetic average \( m_{t,j,v} \), and

\[
\sum_{i=1}^{n} n_i = g_{t,j,v}.
\]

Section 2 shows that the value of the arithmetic average call is bounded from below by:

\[
\hat{\Gamma}_t^A(K) = R^{-T} \sum_{j=0}^{T} \sum_{w=0}^{g_{t,j,v}} \mu_{t,j,v} - K, 0) \right) \times g_{t,j,v}.
\]

Approximation (1.15) can be interpreted in terms of Figure 1 as ignoring any height differences
within the bar; i.e., making use of the sub-bundle means of the arithmetic averages.

To find an exact (or nearly exact) solution, we must find (or approximate) the frequency
distributions of arithmetic averages within divided sub-bundles, a task that is much less
burdensome than enumerating individual paths. For example, when \( T = 20 \), the problem’s largest
sub-bundle contains 5448 paths which attain 52 distinct arithmetic averages, cf. Neave and Stein

To see how frequency distributions are used, let \( m_{t,j,v}^i \), \( i = 1, 2, \ldots, g ; 1 \leq g \leq g_{t,j,v} \),
denote the distinct arithmetic averages attained by the paths in \( B(T, j, v) \). Conditional on \( j \) and \( v \),
denote the ordered averages by:

\[
m_{t,j,v}^i \leq \ldots \leq m_{t,j,v}^g.
\]

Let

\[
k = \arg \min_{n} \{ m_{t,j,v}^n : m_{t,j,v}^n > K \}
\]
Determining the exact contribution to option value of the paths in a divided sub-bundle $B(T, j, v)$ then involves replacing the terms $\{\mu_{T, j, v} - K\}^* g_{T, j, v}$ in (1.15) with the terms

$$(m_{T, j, v}^* - K)(n_b + \ldots + n_y),$$

(1.18)

where

$$m_{T, j, v}^* = (n_b m_{T, j, v}^* + \ldots + n_y m_{T, j, v}^*) / (n_b + \ldots + n_y).$$

(1.19)

Finding the $m_{T, j, v}^*$ involves enumerating combinatorial possibilities, and recent computational experience indicates that it is no more time consuming to use our new recursion and compute upper and lower valuation bounds instead. Moreover, the new recursion adapts readily to valuing American options. We develop both these approaches below.

2. THEORETICAL RESULTS

We call the combination of an upper and lower valuation bound a set of bounds. The difference between an upper and a lower bound is called a gap, which measures the maximum approximation error created by the method of calculating the bounds. We describe the bounds themselves being loose or tight, and choosing between them involves tradeoffs. Tight bounds can be made progressively tighter, but loose bounds cannot. Loose bounds are computationally cheaper to generate, and a loose upper bound will be relatively less accurate than its tight counterpart. However, our loose lower bound is about as good its tighter counterparts, and in valuing European options it is computationally cheaper to employ a loose lower bound irrespective of whether we use a tight or a loose upper bound. On the other hand, we employ tight lower and upper bounds when valuing American options, because our backward recursion uses the data that generate those bounds.

2.1 Loose Lower Bound

Neave and Stein (1998) show that (1.15) defines the loose lower bound. For convenience, their lemma is reproduced here.

**Lemma:** Choose any sub-bundle $B(t, j, v)$. Then its contribution to approximation error, given by the difference between the exact solution and (1.15), is non-negative:

$$\sum_{i=1}^{\lambda} n_i (m_i^* - K)^* p_i^* q_i^{1-i} - n(\mu - K)^* p^* q^{1-i} \geq 0.$$  

(2.1)
Proof: i) If either $K \neq m'_{t,j,v}$ or $m^k_{t,j,v} \neq K$, it follows immediately from the definition of $\mu$ that (2.1) holds with equality.

ii) Now suppose $m'_{t,j,v} < K < m^k_{t,j,v}$, and rewrite the contribution to approximation error as

$$
\sum_{i=1}^{\delta} n_i (m' - K)^+ p^i q^{-i} - \{\sum_{i=1}^{\delta} n_i (m' - K)^+ p^i q^{-i} \}^2 \geq 0.
$$

(2.2)

Negative terms are valued as zero in the left-hand summation, but not in the right-hand summation. Hence the inequality follows.

2.2 Loose Upper Bound

Our loose upper bound, defined using a quadratic approximation, is developed in Ye (1998). Let $B(i,j,v)$ be a divided sub-bundle, and let $m$ be a distinct arithmetic average in $B(T_i,j,v)$. That is,

$$
m = \frac{1}{T + 1} \sum_{t=0}^{T} S_t.
$$

(2.3)

Define the squared average of $m$ as

$$
\|m\|^2 = \frac{1}{T + 1} \sum_{t=0}^{T} S_t^2.
$$

(2.4)

Then define

$$
f(m, K) = a(K)\|m\|^2 - 2m_{t,j,v} + (m_{t,j,v})^2 + b(K)(m - m_{t,j,v}).
$$

(2.5)

where,

$$
a(K) = \min\{m^k_{t,j,v} - K, K - m^k_{t,j,v}\}/(\|m^k_{t,j,v}\|^2 - 2m^k_{t,j,v}m_{t,j,v} + (m^k_{t,j,v})^2)
$$

(2.6)

$$
b(K) = (m^k_{t,j,v} - K - a(K))/(m^k_{t,j,v} - m^k_{t,j,v}).
$$

(2.7)

Lemma: Given any strike price $K$, then $f(m, K) \geq \max(m - K, 0)$, for any $m$ in a divided sub-bundle.

Proof: A proof can be obtained from Ye at the e-mail address given on the cover page.
We then define an upper bound by:

\[
UB(T, j, v; K) = \alpha (\nu_{T, j, v} - 2m_{T, j, v}M_{T, j, v} + (m_{T, j, v}^2) + b(M_{T, j, v} - m_{T, j, v})
\]

(2.8)

where

\[
\nu_{T, j, v} = E[\|m\|^2 | B(T, j, v)]
\]

\[
= \frac{1}{T+1} \sum_{t=0}^{T} E_{T, j, v}(S_t^2)
\]

(2.9)

and the expectation is taken over all paths in the sub-bundle. The \(\nu_{T, j, v}\) can be calculated from the same data as the \(\mu_{T, j, v}\). Section 3 shows the loose upper and lower bounds of this section provide a more nearly accurate estimate of option value than do the point estimates of Hull and White (1993), and without incurring additional computation costs.

2.3 Tight Upper And Lower Bounds

Our tight upper and lower bounds are calculated using a new forward recursion. First let

\[ H_t = (t+1)A_t \]

Where \(A_t\) is defined in (1.7). Since

\[ H_{t+1} = H_t + \mu_t \]

the values of \(H_t\), for all price paths ending at index \(J\), have the same time \(t\) increment. Examining the Gaussian generating function suggests calculating these values using the forward induction scheme illustrated next.

Forward induction scheme

| \(B(t-1, j+2, v)\) | \(B(t, j+1, v+j+1)\) |
| \(B(t-1, j, v)\) |

The scheme displayed in the inset applies at time \(t-1\) to values of \(j\) such that \(-t+1 < j < -t-1\). If \(j = -t+1\), the lower branch of the figure is ignored in performing the forward induction, since the lowest permissible index value at time \(t\) will be \(j = -t\). Correspondingly, if at time \(t-1\) the index \(j = t-1\), the upper branch of the figure is ignored in carrying out the induction forward to time \(t\). If there are \(m\) distinct values of \(H_{t,1}\) in \(B(t-1, j+2, v)\), and \(n\) distinct values of \(H_{t,1}\) in \(B(t-1, j, v)\), then the maximal number of distinct values in \(B(t, j+1, v)\), will be \(m + n\). We need only tabulate the distinct averages and their frequencies.
The tabulation procedure is organized to combine ordered vectors of distinct averages and their frequencies. This procedure means that existing distinct averages need only be scanned once in each forward inductive step, improving computation time. When we calculate approximate solutions, we limit the number of distinct averages stored. (This limit is subsequently indicated by the value of the parameter $L$.) Rather than forcing the averages into a grid, we use existing distinct values and combine frequencies in an approach that minimizes the approximation errors created thereby. Suppose, for example, we have a vector of distinct averages $H \equiv H_{(a,)}$ and its frequencies $F \equiv F_{(a,)}$. To reduce the number of stored values by one, we first find the smallest frequency. We then combine the smallest frequency with the adjacent larger frequency if defining an upper bound, the adjacent smaller frequency if defining a lower bound.

The scheme is shown in the following inset. Let $h_{n,}$, $h_m$ and $h_{m+1}$ be components of $H$, arranged in increasing order, while $f_{n,}$, $f_m$ and $f_{m+1}$ are the corresponding (positive) frequencies in $F$. If we do not wish to carry the value $h_n$ forward to the next iteration, the upper bound calculation proceed as shown next. The analogue for the lower bound calculation is obvious. Note that the manner in which the scheme is defined gives an immediate proof that with enough storage space, the upper and lower limits will both converge to the exact solution. With enough storage space, the exact frequency distribution of the distinct averages is stored for each sub-bundle.

<table>
<thead>
<tr>
<th>Original stored values</th>
<th>$h_{n,}$</th>
<th>$h_n$</th>
<th>$h_{n+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original frequencies</td>
<td>$f_{n,}$</td>
<td>$f_n$</td>
<td>$f_{n+1}$</td>
</tr>
<tr>
<td>New frequencies for</td>
<td>$f_{n,}$</td>
<td>0</td>
<td>$f_n + f_{n+1}$</td>
</tr>
<tr>
<td>upper bound</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>calculation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. The amendments if we wish to make more reductions are obvious.
2. In the case of a tie, we take the frequency associated with the smallest average.
3. If the chosen frequency is a maximum or a minimum we select the second smallest frequency instead, thus ensuring that we always have the value of the maximal and minimal averages in a sub-bundle.
4. Unlike our calculations using loose bounds, we do not here distinguish between divided sub-bundles and those which are not divided. Keeping track of the difference requires as much computation time as generating the frequency distributions for all sub-bundles.
We use the above recursion and the time $t-1$ frequency distributions of distinct arithmetic averages inductively to generate the time $t$ frequency distributions. By design, the recursions are efficient with respect to calculating and storing the frequency distributions of distinct arithmetic averages. However, when $T$ is sufficiently large the number of distinct arithmetic averages can become too great to store in random access memory. Moreover, writing values to disc and then reading them back increases computation time considerably. Computation times can be improved by approximating the frequency distributions with upper (and lower) distributions. We generate the upper (and lower) distributions by respectively increasing (decreasing) some of the averages so they equal other observed values. This strategy keeps the computation times reasonable, keeps the number of recorded distinct averages below a specified maximum, and bounds the true value of the option from above (below). We define the maximal number of distinct averages stored to be $L$. Increasing values of $L$ imply both increased computational accuracy and increased computation time.

3. COMPUTATIONAL RESULTS: EUROPEAN OPTIONS

This section examines the comparative performance of our methods. First, we consider tradeoffs between accuracy and computation time. Second, we compare the present method's accuracy with that of Hull and White (1993). Third, we examine the effect of volatility on both option values and the accuracy of the approximations.

3.1 Accuracy and Computation Time

Since our methods exploit the structure of the data, they are both computationally more efficient and more nearly accurate than competing methods. The speed and accuracy of our two sets of bounds are compared in Table 1. If the user can tolerate errors in the fourth significant digit, the loose bound approach is particularly suitable. For example, it might be employed when many options needed to be valued quickly. On the other hand, the tight bound approach might be preferred in the case of a single option with a relatively large value. Figure 3 shows how increasing $L$ tightens the bounds.

The increase in computation time needed to obtain tight bounds is not inconsequential, both because calculating tighter bounds involves more floating point calculations and because we are only using a PC with a Pentium chip rather than a work station. Using a work station with say, 200MB of RAM can improve computation times substantially over those reported here. Regardless, by changing $L$ we can find a desired trade-off between reasonable computation times and reasonable approximation errors for whatever equipment is used.

---

9 We can only offer direct comparisons for the volatilities reported in the literature, and these are relatively low in relation to the ones we can and do study.
Table 1 European Asian Call Option Prices and Computing Times

<table>
<thead>
<tr>
<th>T</th>
<th>Loose Lower Bound</th>
<th>Loose Upper Bound</th>
<th>Gap</th>
<th>CT</th>
<th>Loose Lower Bound</th>
<th>Tight Upper Bound</th>
<th>Gap</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.136519</td>
<td>0.136519</td>
<td>0</td>
<td>0</td>
<td>0.136519</td>
<td>0.136519</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0.137026</td>
<td>0.137048</td>
<td>0.000022</td>
<td>0</td>
<td>0.137026</td>
<td>0.137036</td>
<td>0.000010</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0.137213</td>
<td>0.137265</td>
<td>0.000052</td>
<td>0</td>
<td>0.137213</td>
<td>0.137231</td>
<td>0.000018</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>0.137322</td>
<td>0.137422</td>
<td>0.000100</td>
<td>0</td>
<td>0.137322</td>
<td>0.137355</td>
<td>0.000033</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>0.137391</td>
<td>0.137543</td>
<td>0.000152</td>
<td>1</td>
<td>0.137391</td>
<td>0.137442</td>
<td>0.000051</td>
<td>14</td>
</tr>
<tr>
<td>36</td>
<td>0.137439</td>
<td>0.137645</td>
<td>0.000206</td>
<td>1</td>
<td>0.137439</td>
<td>0.137502</td>
<td>0.000063</td>
<td>31</td>
</tr>
<tr>
<td>42</td>
<td>0.137474</td>
<td>0.137735</td>
<td>0.000261</td>
<td>2</td>
<td>0.137474</td>
<td>0.137560</td>
<td>0.000086</td>
<td>65</td>
</tr>
<tr>
<td>48</td>
<td>0.137501</td>
<td>0.137816</td>
<td>0.000315</td>
<td>3</td>
<td>0.137501</td>
<td>0.137605</td>
<td>0.000104</td>
<td>118</td>
</tr>
<tr>
<td>54</td>
<td>0.137522</td>
<td>0.137894</td>
<td>0.000372</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>0.137538</td>
<td>0.137965</td>
<td>0.000427</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>0.137552</td>
<td>0.138035</td>
<td>0.000473</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>0.137564</td>
<td>0.138104</td>
<td>0.000540</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>0.137574</td>
<td>0.138170</td>
<td>0.000596</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>0.137582</td>
<td>0.138235</td>
<td>0.000653</td>
<td>68</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>0.137590</td>
<td>0.138300</td>
<td>0.000710</td>
<td>95</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.137600</td>
<td>0.138405</td>
<td>0.000805</td>
<td>170</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1) Initial stock price $1.00, strike price $1.00, risk-free interest rate 0.10 per annum, stock price volatility 0.40 annually, time to maturity 1.5 years. CT is computation time in seconds. In calculations of the tight bounds, L = 100.
2) Valuation bounds are calculated using a C++ program in a Pentium 200 computer. The gaps are the differences between the upper and lower bounds.

3.2 Comparisons of Accuracy

Table 2 compares the accuracy of our methods to those of Hull and White (1993). Not only can our loose bounds be calculated very quickly, but for low volatilities they produce roughly the values as the methods of Hull and White. However, Hull-White use linear interpolation of averages' value, and our investigations show that the distribution of averages is both nonlinear and relatively irregular when compared, say, to the lognormal (cf. Figures 1 and 2). Thus for higher volatilities we would expect our methods to give greater accuracy than can linear.

* Of course, the sum of lognormally distributed variables is not itself lognormal. Our comparison is intended only to show how very irregular the discrete distribution can be.
### Table 2 Comparisons of Accuracy

<table>
<thead>
<tr>
<th>Years to Maturity</th>
<th>Strike price</th>
<th>40</th>
<th>45</th>
<th>50</th>
<th>55</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H&amp;W upper bound</td>
<td>10.755</td>
<td>6.363</td>
<td>3.012</td>
<td>1.108</td>
<td>0.317</td>
</tr>
<tr>
<td></td>
<td>Loose upper bound</td>
<td>10.75480</td>
<td>6.36145</td>
<td>3.00839</td>
<td>1.10557</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Tight upper bound</td>
<td>455</td>
<td>78</td>
<td>62</td>
<td>56</td>
<td>0.31552</td>
</tr>
<tr>
<td></td>
<td>Loose lower bound</td>
<td>439</td>
<td>44</td>
<td>33</td>
<td>13</td>
<td>0.314</td>
</tr>
</tbody>
</table>

|                   | H&W upper bound | 11.545 | 7.616 | 4.522 | 2.420 | 1.176 |
|                   | Loose upper bound | 11.54539 | 7.61578 | 4.52185 | 2.42008 | 70 |
|                   | Tight upper bound | 415 | 98 | 84 | 750 | 1 |
|                   | Loose lower bound | 351 | 15 | 00 | 2.41632 | 09 |

|                   | Loose upper bound | 12.28 | 8.671 | 5.743 | 3.585 | 2.128 |
|                   | Tight upper bound | 678 | 73 | 4489 | 69 | 60 |
|                   | Loose lower bound | 12.28320 | 37 | 89 | 21 | 063 |

|                   | Loose upper bound | 12.95718 | 9.58589 | 9627 | 77 | 87 |
|                   | Tight upper bound | 358 | 56 | 28 | 51 | 6 |
|                   | Loose lower bound | 186 | 961 | 05 | 958 | 38 |

Notes: The initial stock price is $50, the risk-free interest rate is 10% per year, the stock price volatility is 30% per year. The 40-time-steps binomial tree model is used in computations. In calculations of the tight bounds, $L = 100$. H&W upper bounds are quoted from Hull and White (1993). This Table uses loose rather than tight lower bounds, since they are very nearly the same. The interested reader can compare the loose lower bounds in this Table with the corresponding tight lower bounds reported in the next Table.
interpolations, and the subsequent data show this generally to be true, especially for deep in the money options where the Hull-White (1993) method creates relatively large errors. Hull-White offer no means of controlling approximation error, although the extensions of Chalasani et. al. (1988) do.

In addition, the Hull-White methods create systematic biases, as indicated in Table 2a. In contrast, our methods define intervals within which the option value is guaranteed to lie, and while these intervals increase with time and volatility, there does not appear to be any other systematic bias in them. Moreover, our interval estimates can be sharpened by using the tight bound approach and high values of the parameter L. Finally, the cost of obtaining the better estimates is a modest an increase in computation time, as Table 1 already showed.

<table>
<thead>
<tr>
<th>Years/Strike</th>
<th>40</th>
<th>45</th>
<th>50</th>
<th>55</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1.000042</td>
<td>1.0000349</td>
<td>1.001456</td>
<td>1.003114</td>
<td>1.004691</td>
</tr>
<tr>
<td>1.0</td>
<td>1.000074</td>
<td>1.0000265</td>
<td>1.000476</td>
<td>1.001034</td>
<td>1.001618</td>
</tr>
<tr>
<td>1.5</td>
<td>1.000053</td>
<td>1.000144</td>
<td>1.000277</td>
<td>1.000179</td>
<td>1.000283</td>
</tr>
<tr>
<td>2.0</td>
<td>0.999955</td>
<td>1.000046</td>
<td>1.000106</td>
<td>1.000106</td>
<td>0.999817</td>
</tr>
</tbody>
</table>

### 3.3 Accuracy and Volatility

Table 3 shows how both the option values and the accuracy of our bounds are affected by increases in volatility. There are no similar published results concerning the effects of higher volatility on the valuations obtained by competing methods. Figure 4 shows how our methods’ approximation errors compare to the differences in value from modest changes in volatility. These results suggest to us that greater payoffs to future research are more likely to come from improving volatility estimates rather than from further attempts to improve the accuracy of our bounds.
Table 3 Accuracy and Volatility for Tight Bounds

<table>
<thead>
<tr>
<th>T</th>
<th>Volatility 0.4</th>
<th></th>
<th>Volatility 0.8</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tight Lower Bound</td>
<td>Tight Upper Bound</td>
<td>Gap</td>
<td>Tight Lower Bound</td>
</tr>
<tr>
<td>6</td>
<td>0.136519</td>
<td>0.136519</td>
<td>0.000000</td>
<td>0.231945</td>
</tr>
<tr>
<td>12</td>
<td>0.137036</td>
<td>0.137036</td>
<td>0.000000</td>
<td>0.232966</td>
</tr>
<tr>
<td>18</td>
<td>0.137231</td>
<td>0.137231</td>
<td>0.000000</td>
<td>0.233474</td>
</tr>
<tr>
<td>24</td>
<td>0.137339</td>
<td>0.137355</td>
<td>0.00016</td>
<td>0.233682</td>
</tr>
<tr>
<td>30</td>
<td>0.137399</td>
<td>0.137442</td>
<td>0.000043</td>
<td>0.233796</td>
</tr>
<tr>
<td>36</td>
<td>0.137441</td>
<td>0.137502</td>
<td>0.000061</td>
<td>0.233822</td>
</tr>
<tr>
<td>42</td>
<td>0.137476</td>
<td>0.137560</td>
<td>0.000084</td>
<td>0.233901</td>
</tr>
<tr>
<td>48</td>
<td>0.137502</td>
<td>0.137605</td>
<td>0.000103</td>
<td>0.233963</td>
</tr>
</tbody>
</table>

Notes: The initial stock price is $1, the strike price is $1, the risk-free interest rate is 10% per year, time to maturity is 1.5 years, and L = 100.
3.4 Stochastic Volatility Options

Since our methods can generate both upper and lower bounds for given volatility, we can provide a range within which values would fall if volatility were stochastic. Since option values increase with volatility, all we need do is take the upper bound for the maximum volatility, and the minimum bound for the minimum attainable volatility, to obtain a quick estimate of the values that could be attained by a stochastic volatility option. While these values might not be very precise, they are computationally much cheaper than recognizing the effects of stochastic volatility more systematically. Moreover as we have already pointed out, getting the volatility process correct is highly important, and yet that problem has probably received less attention than that of calculating valuations.

4. COMPUTATIONAL RESULTS: AMERICAN OPTIONS

Since our methods for calculating tight bounds use forward recursion to generate the necessary data, we can readily adapt them to develop a backward recursion to value American options. Table 4 reports computing accuracy and speed are reported. The time steps are from 6 up to 30, and L=100 and 200. For time steps not greater than 18, with L = 100, we get exact values, which we specify to the fifth significant digit. For time steps equal to 24 or 30, with L=200, we get fourth significant digit accuracy. Computing times are small, even on a PC. For example, when 24 time steps are used and L=100, we value the American call in one minute, and our solution is accurate to the fourth significant digit.

Table 4 American Asian Call Option Prices and Computing Times

<table>
<thead>
<tr>
<th>T</th>
<th>L=100</th>
<th>L=200</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower bound</td>
<td>Upper bound</td>
</tr>
<tr>
<td>6</td>
<td>0.141268</td>
<td>0.141268</td>
</tr>
<tr>
<td>12</td>
<td>0.146479</td>
<td>0.146479</td>
</tr>
<tr>
<td>18</td>
<td>0.148574</td>
<td>0.148633</td>
</tr>
<tr>
<td>24</td>
<td>0.149445</td>
<td>0.150294</td>
</tr>
<tr>
<td>30</td>
<td>0.150001</td>
<td>0.151745</td>
</tr>
</tbody>
</table>

Notes: Initial stock price $1.00, strike price $1.00, risk-free interest rate 0.10 per annum, stock price volatility 0.40 annually, time to maturity 1.5 years.
Table 5 compares the accuracy of our method\footnote{For reasons of space, we only compare our method with the two indicated. Many other solutions have also been proposed, as our references show.} with Hull and White (1993) and Chalasani, et al. (1998). Table 5 shows that our upper bound calculated with \( L=100 \) is similar to H&W with \( h=0.005 \); and our upper bounds calculated with \( L \geq 150 \) are tighter than H&W with \( h=0.003 \). Next, we get both the upper bounds and lower bounds, while Hull and White only get upper bounds. Furthermore, when \( L=300 \) is used, the bounds converge to the exact value, as by design they must. Since Hull and White do not indicate computing times, we cannot offer this type of performance comparison. In comparison to those obtained in Chalasani, et al., our upper bound with \( L \geq 150 \) and lower bound with \( L \geq 200 \) are tighter. In other words, we get a more nearly accurate solution for the \( T=20 \) problem in only 15 seconds on a PC.

<table>
<thead>
<tr>
<th>( L )</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
<th>300</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper bound</td>
<td>4.81552</td>
<td>4.81343</td>
<td>4.81273</td>
<td>4.81258</td>
<td>4.81248</td>
</tr>
<tr>
<td>Lower bound</td>
<td>4.80988</td>
<td>4.81161</td>
<td>4.81225</td>
<td>4.81240</td>
<td>4.81248</td>
</tr>
<tr>
<td>Gap</td>
<td>0.00564</td>
<td>0.00182</td>
<td>0.00048</td>
<td>0.00018</td>
<td>0.00000</td>
</tr>
<tr>
<td>CT</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>20</td>
<td>47</td>
</tr>
</tbody>
</table>

Chalasani, et al. Bounds

| Upper bound | 4.814 |
| Lower bound | 4.812 |
| Gap | 0.002 |

H&W Upper Bounds

| \( H \) | 0.1 | 0.05 | 0.01 | 0.005 | 0.003 |
| Upper bound | 5.197 | 4.971 | 4.823 | 4.815 | 4.814 |

Notes: The initial stock price is $50, the strike price is $50, the risk-free interest rate is 0.10 per annum, \( \sigma = 0.30 \), \( T = 20 \). H&W upper bounds are quoted from Hull and White(1993) Chalasani, et al. bounds are quoted from Chalasani, et al. (1998).
5. CONCLUSION

This paper provided upper and lower bounds on the values of both European and American arithmetic average fixed strike calls. For European options, our loose bounds are better than those in the literature and as quick or quicker to calculate. Our tight bounds, calculated using forward induction, take longer to calculate but can obtain as close an approximation to the true value as desired. Our values for American options are more nearly accurate than those of competing methods and are also obtained with only small amounts of computing time.
References


Ye, George L., "Pricing Asian Options Using A Path Bundling Technique," PhD Dissertation, Queen's School of Business, Queen’s University, 1998 (in progress).

Figure 1: A Bundle and its Sub-bundles for B(8, 4), \( \sigma = 0.80 \).
Figure 2 The Cumulated Probabilities of The Exact Distinct Arithmetic Averages Distribution in $B(20,10,105)$ and The Lognormal Distribution

Notes: The cumulated probabilities of the exact distinct arithmetic average (DAA) is calculated by using our forward induction procedure, the lognormal distribution is one with the same mean and variance as those of the exact distribution.
Figure 3 Upper Bounds and Lower Bounds in B(20, 10, 105)

(a) upper bounds

(b) lower bounds

Notes: "Exact" is the subbundle expected payoffs as a function of the strike price "U-L", "L-L" are the upper bound and lower bound calculated by our tight bound approach with the number of the intervals to be L, L=20, 100.
Notes: The Asian option is an European call with 1.5 years to maturity. The initial stock price is $1, the strike price is $1, the interest rate is 10% per year, the volatilities from 40% to 42.5% per year are used with an increment of 0.5%. T is the number of time steps in the binomial tree, which is increased up to 48.