ENHANCED MONTE CARLO ESTIMATES FOR AMERICAN OPTION PRICES

Mark Broadie*
Paul Glasserman*
Gautam Jain†

June 20, 2000

This paper appeared in the *Journal of Derivatives*, 1997, Vol. 5, No. 1 (Fall), 25–44.

ABSTRACT

A simulation-based methodology to price American options with finite exercise opportunities has recently been introduced by Broadie and Glasserman [1995a]. This method simulates the evolution of underlying assets via random trees that branch at each of the possible early-exercise dates. From these trees, two consistent and asymptotically unbiased price estimates, one biased high and one biased low, are obtained. These two estimates can be used to give a conservative confidence interval for the option price. In this paper, we develop several enhancements to improve the efficiency of the two estimates so that the resulting confidence interval is small.

Since branching can be computationally very expensive, we suggest "pruning" the trees by eliminating branching whenever possible, thus cutting down the simulation time and allowing for faster convergence of the estimates. In particular, it is shown that branching at the penultimate exercise point is certainly not required whenever a formula for pricing the corresponding European option is available.

Next, in order to further improve the estimators, we forego the idea of generating branches from independent samples. Indeed, we demonstrate that if half of the branches at a node are generated using the antithetic variates of the other half, both bias and variance are reduced significantly. Further enhancement is possible by combining pruning with this technique. It is also shown that by selecting the branches from Latin hypercube samples, better results are obtained.

We conclude by showing that an option with infinite exercise opportunities can be well approximated by extrapolating a series of options with finitely many exercise points.

*Postal address: Graduate School of Business, Uris Hall, Columbia University, New York, NY 10027
†Postal address: L.O.G. International Corp., 150 East 52nd Street, 22nd Floor, New York, NY 10022.

1 INTRODUCTION

Using Monte Carlo simulation for pricing options was first proposed by Boyle [1977]. One advantage of this method over the binomial method of Cox, Ross, and Rubinstein [1979] is that it does not grow exponentially with the number of state variables or underlying assets. In addition, path dependencies can be taken into account easily. The chief drawback, on the other hand, has been its inability to incorporate the early-exercise feature of American-style derivative securities.

In a recent paper, however, Broadie and Glasserman [1995a] proposed a way to use simulation for pricing American options with finitely many exercise opportunities. Two estimators—one biased high and the other biased low—are obtained from a simulated tree, which branches at each of the possible early-exercise points. A conservative confidence interval for the option price is obtained as a consequence. Details regarding the two estimators are presented in the next section. Broadie and Glasserman [1995a] prove that these estimators are consistent (i.e., converge in probability) and unbiased in the limit.

In this article, we present several enhancements to reduce the bias as well as variance of these estimators. The first one utilizes information about European option prices to reduce the number of branches in a simulated tree. This approach will be discussed in Section 3. Next, in Sections 4 and 5, it is shown that by combining either antithetic variates or Latin hypercube sampling with the original approach, significant improvement ensues. However, the proper implementation of these methods requires some care in the present context. Furthermore, in Section 6, we demonstrate that an option with infinitely many exercise opportunities can also be successfully priced using extrapolation. Finally, in Section 7, we illustrate all these enhancements numerically by pricing several options in higher dimensions.

2 MODEL AND ESTIMATORS

We use the following notation to formulate the problem of pricing American options:

- Let there be d exercise opportunities at times $0 = t_0 < t_1 < \cdots < t_{d-1} = T$, T being the time of expiration for the option initiated at time t_0 or today. Less formally, we will sometimes indicate $t = t_0, t_1, \ldots, t_{d-1}$ by $t = 0, 1, \ldots, T$.

- Consider a vector-valued Markov chain $\{S_t : t = 0, 1, \ldots, T\}$ consisting of all information required to determine the payoff from exercising an option. For simplicity, think of the com-

ponents of S_t as lognormally distributed stock prices. In practice, S_t would record all relevant information about asset prices, interest rates, exchange rates, and supplementary variables needed to eliminate path-dependence.

- e^{-R_t} is the discount factor from $t-1$ to t. We take R_t to be a component of the vector S_t and assume $R_t \geq 0$ for all t. In addition, let $R_{0t} = \sum_{i=1}^{t} R_i$.
- $h_t(s)$ is the payoff from exercise at time t in state s, $t = 0, \ldots, T-1$.
- $g_t(s) = E[e^{-R_{t+1}} f_{t+1}(S_{t+1}) | S_t = s]$ is the *continuation value* at time t in state s, $t = 0, \ldots, T-1$.
- $f_t(s) = \max\{h_t(s), g_t(s)\}$ is the option value at time t in state s, $t = 0, \ldots, T-1$.
- $f_T(s) = g_T(s) = h_T(s)$.

It is known that the price of an American option is the solution of the following optimal stopping problem:[1]

$$f_0(S_0) \;=\; \max_{\tau} E[e^{-R_{0\tau}} h_\tau(S_\tau)], \tag{1}$$

where the maximum is over all stopping times τ taking values in $\{0, 1, \ldots, T\}$. The optimal policy stops at

$$\tau^* \;=\; \inf\{t = 0, \ldots, T : h_t(S_t) \geq g_t(S_t)\};$$

i.e., the first time the immediate exercise value is at least as great as the continuation value.

Broadie and Glasserman [1995a] show, under reasonable restrictions, there is no unbiased estimator of (1). As an alternative, therefore, they introduce two estimators, one biased high and one biased low, both consistent and asymptotically unbiased. We discuss these next.

Estimators. In contrast to the Monte Carlo approach for pricing European options, the evolution of S_t is simulated using random trees rather than just sample paths. Given a value of the *branching parameter* b, the evolution of the tree can be described recursively as follows. From the (fixed) initial state S_0, we generate b independent samples $S_1^1, \ldots, S_1^b$ of the state at time $t = 1$. From each of these b samples, $S_1^{i_1}$, $i_1 = 1, \ldots, b$, in turn, generate b new samples, $S_2^{i_1 1}, \ldots, S_2^{i_1 b}$ at time $t = 2$; hence there are a total of b^2 nodes at $t = 2$. Repeat this procedure for every possible exercise time

[1] See, e.g., Section 6 of Karatzas [1989] (especially Theorem 6.5) for the formulation of American option pricing as an optimal stopping problem. We are assuming here and throughout that the underlying process S is simulated under the risk-neutral measure, so that E corresponds to expectation under this measure.

t. To be precise, from each node value $S_t^{i_1\cdots i_t}$ at time t, we generate b samples $S_{t+1}^{i_1\cdots i_t j}$, $j = 1, \ldots, b$, conditionally independent of each other given $S_t^{i_1\cdots i_t}$ and each having the distribution of S_{t+1} given $S_t = S_t^{i_1\cdots i_t}$. Thus, each sequence $S_0, S_1^{i_1}, S_2^{i_1 i_2}, \ldots, S_T^{i_1\cdots i_T}$ is a realization of the Markov chain S_t. Figure 1 illustrates a tree with parameters $b = 3$ and $M = 1$, where M denotes the dimension of the state vector. To understand the above notation, consider the node at $t = T = 2$ corresponding to the stock price of 87.59. In our notation it is represented as $S_2^{2,3}$—the subscript 2 indicates that we are looking at a node at time $t = 2$ and the superscripts indicate that it is the third branch among the branches emanating from the second node at $t = 1$. We therefore have the entire path to the node. To simplify notation, henceforth, we will indicate the path $i_1 \cdots i_t$ to a node at time t by α_t. Thus, $S_t^{i_1\cdots i_t}$ is written as $S_t^{\alpha_t}$, $i_1 \cdots i_t j$ is replaced by $\alpha_t j$, and so on.

The *high estimator* is simply the result of applying dynamic programming to the random tree. More precisely, working backwards through the tree using the recursions

$$\Theta_T^{\alpha_t} \;=\; h_T(S_T^{\alpha_t})$$

and

$$\Theta_t^{\alpha_t} \;=\; \max\left\{ h_t(S_t^{\alpha_t}),\; \frac{1}{b}\sum_{j=1}^{b} e^{-R_{t+1}^{\alpha_t j}}\Theta_{t+1}^{\alpha_t j} \right\}, \tag{2}$$

we compute the high estimator $\Theta = \Theta_0$. At any node, therefore, the high estimate is the maximum of the immediate exercise value and the average of discounted high estimates at successor nodes.

The high estimator uses all branches emanating from a node to approximate both the optimal action (exercise or continue) and the value of this decision. The *low estimator* differs in that it separates the branches used to determine the action from those used to determine the continuation value. Here is a verbal description:

1. At each node in the tree, reserve one successor node. Average the discounted low estimator values at the other $b - 1$ successor nodes.

2. If the average obtained is less than the immediate exercise value, set the node value equal to the immediate exercise value; otherwise, set the node value equal to the discounted value from the reserved node.

3. Average the resulting node value over all b ways of selecting the reserved successor node.

4. Repeat these steps backwards through the tree.

Now we give a more precise formulation. First let

$$\theta_T^{\alpha_t} \;=\; h_T(S_T^{\alpha_t}). \tag{3}$$

Next, set

$$\eta_t^{\alpha_t j} \;=\; \begin{cases} h_t(S_t^{\alpha_t}) & \text{if } h_t(S_t^{\alpha_t}) \;\geq\; \dfrac{1}{b-1}\displaystyle\sum_{\substack{i=1\\ i\neq j}}^{b} e^{-R_{t+1}^{\alpha_t i}}\theta_{t+1}^{\alpha_t i}, \\ e^{-R_{t+1}^{\alpha_t j}}\theta_{t+1}^{\alpha_t j} & \text{otherwise.} \end{cases} \tag{4}$$

Then let

$$\theta_t^{\alpha_t} \;=\; \frac{1}{b}\sum_{j=1}^{b}\eta_t^{\alpha_t j}, \tag{5}$$

for $t = 0, \ldots, T-1$. Finally, the low estimate θ is given by θ_0.

By taking the upper confidence limit of the high estimator and the lower confidence limit of the low estimator we obtain a conservative confidence interval for the true price.

To summarize, this method requires constructing random trees parameterized by b, the number of branches per node. Once these trees have been obtained,[2] work backwards through the tree to compute the option price. In this way, the early-exercise feature is incorporated in the simulation methodology. The option price itself is determined in terms of two estimates: one is biased high and one is biased low, but both are unbiased in the limit as b goes to infinity.

An example. Figure 1 depicts a random tree corresponding to a single simulation run. The call option we are trying to price has a single underlying stock with an initial price of \$100.00, and it has three exercise opportunities at times 0, $T/2$ and T. The strike price is \$100.00, time to maturity is 1 year, annual interest rate is 5%, volatility is 0.2, and the dividend rate is 10%. In this figure, the low and high estimates are reported in parentheses next to the random stock price at each node. It is easy to compute the two estimates at time T; e.g., at the topmost node at time T in Figure 1, the stock price is 112.66, so both estimates for this node are $112.66 - 100.00 = 12.66$. At the topmost node at time $T/2$, on the other hand, since the immediate exercise value is 5.67, the high estimate is the maximum of 5.67 and the discounted estimate of the continuation value; hence, it is $\max\{5.67, e^{-0.025}(12.66 + 0.00 + 8.41)/3\} = \max\{5.67, 6.85\} = 6.85$. The low estimate is the average of $b = 3$ intermediate values. In order to get the first of these three values, compare the discounted average of the payoffs at the last two nodes (0.00 and 8.41, respectively)

[2] It is unnecessary to store more than $O(bd)$ nodes at any time. See Broadie and Glasserman [1995a] for implementation details.

with the payoff from immediate exercise (5.67); if the first value is higher, the optimal decision is to continue, otherwise it is optimal to exercise. Consequently, since $e^{-0.025}(0.00 + 8.41)/2 < 5.67$, the decision to exercise immediately is inferred, and the first value is 5.67. Similarly, the second one is 0.00 because $e^{-0.025}(12.66 + 8.41)/2 > 5.67$, i.e., it is optimal to continue and a continuation value of 0.00 is extracted from the second node. The third one is $e^{-0.025}(8.41) = 8.20$ because $e^{-0.025}(12.66 + 0.00)/2 > 5.67$. Consequently, the low estimate is $(5.67 + 0.00 + 8.20)/3 = 4.62$. The low and high estimates at other nodes are obtained in the same manner.

The low and high estimates for the price of the option corresponding to this simulation run are 3.43 and 4.16 respectively. Repeating the procedure many times over by simulating new trees, we get refined values of the two estimates as their respective averages over all the simulation runs. More importantly, we can get standard errors, and hence confidence intervals, for these estimates, and a conservative confidence interval for the option price. If we let d be the number of exercise opportunities and n the number of replications, then the work required to carry this out grows like nb^{d-1}. In particular, increasing the branching parameter is typically far more costly than increasing the number of replications. However, increasing b is essential for reducing bias and thus reducing the width of the confidence interval. This difficulty motivates our investigation into techniques that help reduce the size of the confidence interval without having to increase b.

3 PRUNING

Pricing a European option is generally easier than pricing the American counterpart: there is no optimization involved in the European price. It is therefore natural to try to exploit information obtained from the European case in pricing the American option. In Broadie and Glasserman [1995a], it has been shown through examples that the European price provides a highly effective control variate. We now explain how the European price can also be used for *pruning*, i.e., reducing the number of nodes in the simulated trees. Specifically, two pruning techniques are discussed below.

Pruning at the last step. Let $T - 1$ denote the penultimate exercise opportunity. The optimal action at time $T - 1$ depends on which is greater, the immediate exercise value $h_{T-1}(S_{T-1})$ or the continuation value $g_{T-1}(S_{T-1})$. The estimators Θ and θ implicitly estimate the continuation value at each node. But at time $T - 1$ the continuation value is just the value of a European option initiated at time $T - 1$ and maturing at time T. Computing this value directly and efficiently

eliminates the need to branch at the penultimate node. In other words, if $\ell_T(S_{T-1})$ denotes the price of the European option initiated at time $T-1$ with initial stock price S_{T-1} and expiring at time T, then the low and high esimators are both set to $\max\{h_{T-1}(S_{T-1}), \ell_T(S_{T-1})\}$. Since this implies that we do not need to generate successor nodes for the nodes at $T-1$, the work required is reduced to $O(nb^{d-2})$.

Returning to the tree in Figure 1, it is not necessary to branch at time t_1. As an illustration, the low and high estimates at the topmost node at time t_1 would then become $\max\{5.67, 7.20\} = 7.20$, where 7.20 is the price of the European call option with initial stock price 105.67 and time to maturity half a year.

Intermediate pruning. The sole reason for branching (as opposed to simulating sample paths in the usual way) is to allow for consistent estimation of the optimal action at a node. Suppose that at time t there is a node corresponding to state s. If we knew that the immediate exercise value is smaller than the continuation value, i.e., $h_t(s) < g_t(s)$, we would know that the optimal action is to continue and there would be no need to branch; it would suffice to generate just one successor node. Of course, in general we do not know $g_t(s)$ since g_t is itself the value function of an optimal stopping problem. But if we can find¡ an easily computed lower bound $\ell(s) \leq$ (respectively $<$) $g_t(s)$, we can check if $h_t(s) <$ (respectively $\leq$) $\ell(s)$. If this holds, stopping is guaranteed to be suboptimal so there is no need to branch. If there is no branching out of node $i_1 \cdots i_t$, then (2) gets replaced by $\Theta_t^{\alpha_t} = \exp(-R_{t+1}^{\alpha_t 1})\Theta_{t+1}^{\alpha_t 1}$ and (5) gets replaced with $\theta_t^{\alpha_t} = \exp(-R_{t+1}^{\alpha_t 1})\theta_{t+1}^{\alpha_t 1}$. Pruning at a node in this manner considerably reduces the work required per tree because it eliminates $b-1$ successor nodes along with their progeny.

In virtually all practical examples, the value of an option remains strictly positive throughout its existence. Thus, a simple choice of bound is $\ell(s) = 0$: at any node at which the immediate exercise value is zero, there is no need to branch. This test is free because this choice of ℓ requires no computational effort. For example, in Figure 1, since the immediate exercise value at time 0 is 0, it is optimal to continue and branching is unnecessary; we should generate just one successor node instead of three.

In the case $h_t(s) > 0$, we may decide to compare the immediate exercise value with a more refined bound. Natural choices are $\ell_{t+1}(s), \ldots, \ell_T(s)$, where $\ell_k(s)$ is the value of a European option initiated in state s and maturing at time k. Each of these corresponds to a particular (suboptimal) exercise policy for the American option and thus provides a lower bound on $g_t(s)$. If $h_t(s) < \ell_k(s)$

for any $k = t+1, \ldots, T$, there is no need to branch. In the examples in Section 7, we illustrate this approach using only $\ell_T(s)$.

It may be easy to understand the idea behind pruning pictorially. Consider Figures 2 and 3. Specifically, compare the price $\ell_T(S_{t_j})$ of the European option initiated at time t_j and expiring at T with the immediate exercise value at time t_j, $h_{t_j}(S_{t_j})$. If the latter is greater, branch in the usual fashion as shown in Figure 2. If, on the other hand, the former is greater, generate only one successor node, i.e., only one branch emanates from this node;[3] see Figure 3.

4 ANTITHETIC BRANCHING

As mentioned earlier, using the European price as a control variate helps in reducing the standard error for the estimates. Another variance-reduction technique that proves useful is *antithetic branching*. Specifically, we divide the branches emanating from a node into two equal sets: the stock prices for the first half are generated in the usual manner, while the second set is obtained by changing the signs of all normal random variables in the first set. Further branching at all nodes, antithetic or not, is done in the same fashion. Figure 4 provides a simple illustration of antithetic branching. In this figure, the node marked $1'$ is the antithetic mate of the node marked 1, etc. Note that this approach is slightly different from the traditional way of using antithetic variates to reduce variance of Monte Carlo estimates for European option prices. Normally, for every simulated path, another path is generated using the corresponding antithetic variates.[4,5]

In Section 2, for each node value $S_t^{\alpha_t}$, the b samples $S_{t+1}^{\alpha_t j}$, where $j = 1, 2, \ldots, b$ and α_t denotes the sequence $i_1 i_2 \ldots i_t$, were generated independently of each other. Now we let b be even and divide the samples into two sets of size $b/2$ each. Let $S_{t+1,k}^{\alpha_t j}$ be the j^{th} sample value, $j = 1, \ldots, b/2$, in the k^{th} dimension, $k = 1, \ldots, M$, which is generated using the normal random variable $X_k^{\alpha_t j}$. In order to carry out antithetic branching, use $X_k^{\alpha_t j} = -X_k^{\alpha_t j - b/2}$ to find the sample values for $j = b/2+1, \ldots, b-1, b$.[6] In Figure 4, e.g., if $X^{(1)}$ is used to generate one of the nodes marked 1,

[3] Note that this does *not* mean that we generate only one branch from that point onwards. The decision to branch or not to branch needs to determined separately at every node.

[4] See, e.g., Chapter 14 in Hull [1993].

[5] Another approach is to generate an antithetic tree, however we need to generate a second tree using the antithetic variates from the first tree. It can be verified that doing this does not improve the estimators. In fact, it makes the estimates worse because extreme values in one tree result in extreme values in the second tree as well, although in the opposite direction. The resulting price estimates from both trees are therefore poor. Antithetic branching, on the other hand, works remarkably well as we demonstrate later in Section 7.

[6] This is equivalent to saying that if Y is the *uniform* random variable used to generate the stock price at a node, use $1 - Y$ to generate its antithetic mate.

use $-X^{(1)}$ to generate its corresponding antithetic mate which is marked by $1'$.

If the b samples at a node are generated independently of each other, as was done in Section 2, then the estimators are biased because a simulation run with finite number of branches does not replicate the distribution of stock prices perfectly. In particular, the stock prices following a node may be too high or low, thus increasing the value of the high estimate and lowering the low estimate. By using antithetic branching, this bias is reduced to a certain extent because we balance high stock prices with low stock prices and vice versa. Moreover, since new random variables need to be generated at only half of the branches, it takes less time as compared with the original approach; e.g., in the five-asset case in Example 1 of Section 5, we found that using antithetic branching reduced the computation time by approximately 25%.

Estimators. The high estimate is calculated in the same manner as in Section 2, but there is a difference in the calculation of the low estimate. Specifically, we need to do the following at each node.

1. Divide the branches into two sets consisting of 2 and $b-2$ branches, respectively. The two branches in the first set form an antithetic pair, and the continuation value is taken to be the average of discounted low estimates at these two branches. The decision to continue is based on the remaining $b-2$ branches.

2. Average the discounted low estimates at these $b-2$ branches. If this value is greater than the immediate exercise value, set the low estimate at the node to be the estimated continuation value; otherwise, set it to be the immediate exercise value.

3. Repeat these steps for all $b/2$ ways of selecting the reserved antithetic pair, and set the low estimate at the node to be the average of the $b/2$ values obtained in this manner.

4. Continue in a backward fashion through the tree until all nodes are covered.

Equation (2) from Section 2 for the high estimator, and equation (3) for $\theta_T^{\alpha_t}$, therefore remain unchanged. But the equations for $\theta_t^{\alpha_t}$, (4) and (5), are modified as follows. If

$$h_t(S_t^{\alpha_t}) \;\geq\; \frac{1}{b-2}\sum_{\substack{i=1\\ i\neq j}}^{b/2}[\exp(-R_{t+1}^{\alpha_t i})\;\theta_{t+1}^{\alpha_t i} + \exp(-R_{t+1}^{\alpha_t i+b/2})\;\theta_{t+1}^{\alpha_t i+b/2}],$$

then

$$\eta_t^{\alpha_t j} \;=\; h_t(S_t^{\alpha_t});$$

otherwise

$$\eta_t^{\alpha_t j} \;=\; \frac{1}{2}[\exp(-R_{t+1}^{\alpha_t j})\ \theta_{t+1}^{\alpha_t j} + \exp(-R_{t+1}^{\alpha_t j+b/2})\ \theta_{t+1}^{\alpha_t j+b/2}],$$

for $j = 1, \ldots, b/2$. Consequently, $\theta_t^{\alpha_t}$ can be calculated using

$$\theta_t^{\alpha_t} \;=\; \frac{2}{b}\sum_{j=1}^{b/2} \eta_t^{\alpha_t j},$$

and the low estimator is θ_0.

Combining with pruning. In some cases, further improvement can be obtained by using pruning and antithetic branching together. As described in Section 3, in order to prune the branches at a node at time t_j, we compare the immediate exercise value $h_{t_j}(S_{t_j})$ (provided $h_{t_j}(S_{t_j}) > 0$) with the price of a European option $\ell_T(S_{t_j})$ initiated at t_j with initial stock value S_{t_j} and expiring at T. If $\ell_T(S_{t_j}) > h_{t_j}(S_{t_j})$, an optimal decision to continue is inferred and branching is unnecessary. To make this technique effective when combined with antithetic branching though, if the decision is not to branch at a node, *two* successor nodes, corresponding to an antithetic pair, emanate from the node. That is, if the European price implies that the optimal decision is to continue at a node $S_t^{\alpha_t}$, then generate two successor nodes $S_{t+1}^{\alpha_t j},\ j = 1, 2$. If $X_k^{\alpha_t 1},\ k = 1, \ldots, M$, are the normal random variables used to get the first sample, use $-X_k^{\alpha_t 1}$ to generate the second one. This approach differs slightly from the pruning technique described in Section 3 where only *one* successor node was generated when an optimal decision to continue was inferred from the European price. Figure 5 provides a simple illustration of combining pruning with antithetic branching.

Estimators. If the European price implies that it is optimal not to branch at a node, the high estimator is simply

$$\Theta_t^{\alpha_t} \;=\; \frac{1}{2}[\exp(-R_{t+1}^{\alpha_t 1})\ \Theta_{t+1}^{\alpha_t 1} + \exp(-R_{t+1}^{\alpha_t 2})\ \Theta_{t+1}^{\alpha_t 2}].$$

The low estimator $\theta_t^{\alpha_t}$ can also be analogously defined in this situation: just replace Θ by θ in the previous equation.

5 LATIN HYPERCUBE SAMPLING (LHS)

As mentioned in the previous section, if stock prices at the branches of a node are generated independently of each other, then they might not replicate the true distribution very well. To improve

this replication, a useful technique is to select the branches from Latin hypercube samples.[7] Points selected in this manner have a lower sample variance than if the points were generated independently of each other.

Loosely speaking, for LHS in a single dimension, we divide the interval from 0 to 1 into b equal parts and ensure that exactly one sample is chosen from each of these subintervals. In addition, each sample point is uniformly distributed within its subinterval. In a single dimension, therefore, the methodology coincides with stratified sampling. In a multi-dimensional setting, however, the interval from 0 to 1 is divided into b equal parts in every dimension, and b points are then chosen from the resulting multi-dimensional cubes. If, for instance, the dimension is M, then b points need to be picked from the resulting b^M cubes. It is ensured that exactly one point is chosen from each stratum in every dimension. Mathematically, instead of generating the k^{th} dimension of the j^{th} sample from an ordinary uniform $[0,1]$ random variable, it is generated from $X_k^{\alpha_t j}$, where

$$X_k^{\alpha_t j} \;=\; \frac{\pi_k^{\alpha_t}(j) + U_k^{\alpha_t j}}{b}, \quad j = 1, \ldots, b,\ k = 1, \ldots, M; \tag{6}$$

$\pi_k^{\alpha_t}$ is a random permutation of $\{0, 1, \ldots, b-1\}$ uniformly distributed over $b!$ possible permutations,[8] $U_k^{\alpha_t j}$ is a random variable uniformly distributed between 0 and 1, independently generated for all k and j.

An example. Let $b = 4$ and $M = 2$, i.e., it is a two-dimensional problem with four branches per node. Dividing the interval between 0 and 1 into four equal parts in the two dimensions results in 16 possible squares from which we can pick the 4 branches; see Figure 6. Set the first permutation to be $\pi_1 = \{0, 1, 2, 3\}$ (see footnote 8). Now if a random permutation of $\{0, 1, 2, 3\}$ turns out to be $\pi_2 = \{2, 1, 3, 0\}$, then we need to generate the four branches from the colored squares in Figure 6. Observe that exactly one point is chosen from every row and column. To explain this further, let us suppose the initial uniform random variates were $\{(0.2, 0.4), (0.9, 0.1), (0.6, 0.5), (0.7, 0.7)\}$. These are four pairs of uniform $[0,1]$ random variates for the four branches in two dimensions. Using equation (6), the first tuple $(0.2, 0.4)$ changes to

$$\left(\frac{0 + 0.2}{4}, \frac{2 + 0.4}{4}\right) \;=\; (0.05, 0.6).$$

Modifying the others in the same manner, we end up with a new set of uniform random variates: $\{(0.05, 0.6), (0.475, 0.275), (0.65, 0.875), (0.925, 0.175)\}$.

[7] Latin hypercube sampling was proposed by McKay, Conover, and Beckman [1979].

[8] Actually, in our setting, only $M - 1$ random permutations are required at a node; we set the first permutation to be the identity permutation $\{0, 1, \ldots, b-1\}$.

Estimators. For the purpose of computing the estimators, we divide the branches into two equal sets and then choose each set of points from independent Latin hypercube samples. For example, if there are 50 branches per node, generate 25 successor nodes using one set of Latin hypercube samples and generate the remaining 25 from another independent set. Stated differently, at $S_t^{\alpha_t}$, generate $S_{t+1}^{\alpha_t j}$, $j = 1, \ldots, b/2$, using

$$X_k^{\alpha_t j} \;=\; \frac{2[\tilde{\pi}_k^{\alpha_t}(j) + U_k^{\alpha_t j}]}{b},$$

where $\tilde{\pi}_k^{\alpha_t}$ is a random permutation of $\{0, 1, \ldots, b/2-1\}$. Repeat this for $S_{t+1}^{\alpha_t j}$, $j = b/2+1, \ldots, b-1, b$.

The high estimate is computed in the same manner as described in Section 2. Even though the branches were separated into two sets and each set was generated using independent Latin hypercube samples, apply dynamic programming in the usual manner to the branches to get the high estimate. In order to calculate the low estimate, on the other hand, one of the two above-mentioned sets is used to make the continuation decision while the continuation value itself is taken from the other set. In other words, carry out the following steps at a node.

1. Compute the average of the discounted values of low estimators for the two sets separately.
2. Compare the continuation value from the first set with the immediate exercise value. If the immediate exercise is lower, set the node value to be the continuation value from the other set; otherwise set it to be the immediate exercise value.
3. Repeat this for the second set, and take the low estimate at the node to be the average of the two values.
4. Work backwards through the tree until all nodes are covered.

Mathematically, once again $\Theta_t^{\alpha_t}$ is as defined in equation (2), but the low estimate is formulated differently:

$$\eta_t^{\alpha_t j} \;=\; \begin{cases} h_t(S_t^{\alpha_t}) & \text{if } h_t(S_t^{\alpha_t}) \;\geq\; \dfrac{2}{b} \displaystyle\sum_{i=l_{j,1}}^{u_{j,1}} \exp(-R_{t+1}^{\alpha_t i})\; \theta_{t+1}^{\alpha_t i}, \\ \dfrac{2}{b} \displaystyle\sum_{i=l_{j,2}}^{u_{j,2}} \exp(-R_{t+1}^{\alpha_t i})\; \theta_{t+1}^{\alpha_t i} & \text{otherwise,} \end{cases}$$

for $j = 1, 2$, where $l_{1,1} = l_{2,2} = 1$, $u_{1,1} = u_{2,2} = b/2$, $l_{1,2} = l_{2,1} = b/2 + 1$ and $u_{1,2} = u_{2,1} = b$. As a result,

$$\theta_t^{\alpha_t} \;=\; \frac{1}{2}[\eta_t^{\alpha_t 1} + \eta_t^{\alpha_t 2}],$$

and, finally, the low estimator is θ_0.

6 EXTRAPOLATION

An American option can be exercised at any time before maturity. So far, however, we have allowed for exercise at only finite points in time. To overcome this shortcoming, an extrapolated estimate for the price of an option with infinite exercise opportunities can be obtained from the prices of certain options with finite exercise points. Richardson extrapolation, for instance, can be used effectively for this purpose.[9] Specifically, a series of options with increasing number of exercise opportunities are considered. Since the limit of this series is the option whose price is to be determined, extrapolating this series presents a convenient way of tackling the problem. Let C_1 be a call that can only be exercised at times 0 and T, the maturity. Allow C_2 to be exercised at 0, $T/2$ and T, C_3 at 0, $T/3$, $2T/3$ and T, and so on. Since it is easy to value options with fewer exercise opportunities, these calls can be priced using the simulation-based methodology presented in this paper. One can then use Richardson extrapolation to arrive at a reasonable estimate for the price of call C that can be exercised at any time before maturity. For illustrative purposes, let us suppose that we decide to use C_1, C_2 and C_3 to find the extrapolated estimate. Applying Richardson extrapolation to C_1, C_2 and C_3, we arrive at the following equation:[10]

$$C \;=\; C_3 + \frac{7}{2}(C_3 - C_2) - \frac{1}{2}(C_2 - C_1). \tag{7}$$

C then provides an estimate for an option with infinite exercise opportunities. The last example in the following section uses equation (7) to find the extrapolated estimates.

7 NUMERICAL RESULTS

In this section, we test the techniques discussed so far by using them to price various options. As a first example, we consider an American option on the maximum of several assets. When the number of underlying assets is two, we compare the prices obtained via simulation with those obtained using the lattice approach. Since it is not computationally feasible to price the option using the lattice approach when the number of assets is five, we consider a second example in five dimensions for which the true price can be accurately computed. Next, in order to demonstrate the advantage of

[9] Geske and Johnson [1984] used Richardson extrapolation to value an American put which can be exercised at every instant before maturity.

[10] The derivation is listed in the Appendix of Geske and Johnson [1984].

using simulation, a much harder option, which has features of quanto, basket and spread options, is considered. Finally, we illustrate with an example how extrapolation can be utilized for pricing purposes when the number of exercise opportunities is not finite.

Example 1. In order to compare and evaluate the enhancements described in this article, we picked a generic example of an American option on the maximum of M assets. Specifically, we considered two cases: $M = 2$ (Table 1) and $M = 5$ (Table 2). In these examples, the state is M-dimensional with components $S_t^{(i)}$, $i = 1, \ldots, M$. These asset prices evolve according to the rule

$$S_{t_{j+1}}^{(i)} \;=\; S_{t_j}^{(i)} \exp\{(r - \delta_i - \frac{1}{2}\sigma_i^2)(t_{j+1} - t_j) + \sqrt{t_{j+1} - t_j}\; W_j^{(i)}\}, \quad i = 1, \ldots, M;$$

where r is the (constant) interest rate, δ_i are the dividend yields, and $\{W_j^{(i)}, i = 1, \ldots, M\}$ are mean-zero normal random variates with standard deviations σ_i and correlation ρ_{ik} for $i \neq k$. For simplicity, we take $S_0^{(i)}$ to be identical for all i, and let this common initial asset price vary. The immediate exercise value at time t, before or at maturity, is given by $h_t(s) = \max\{\max\{S_t^{(1)}, \ldots, S_t^{(M)}\} - K, 0\}$, where K is the *strike price.* Of course, since we are using simulation, we are not constrained to use only lognormally distributed stock prices as we are doing in this particular example.

Tables 1 and 2 chart the results for the following parameters: the annualized interest rate $r = .05$; all assets have dividend yields $\delta_i = 0.10$ and volatilities $\sigma_i = 0.20$; their correlations are identically $\rho_{ij} = 0.30$, $i \neq j$; the strike price K is 100; the time to expiration T is 1 year; there are four exercise opportunities. In the two-asset case, we implemented pruning using the European value at the penultimate step and intermediate pruning based on first checking if $h_t(s) > 0$ and, if so, then comparing with a single European option maturing at time T.[11] Only intermediate pruning was used for the five-asset example because computing the European formula becomes tediously slow in this case. At nodes passing both these tests we generate $b = 50$ branches. For each value of the initial stock price, the first two rows in the two tables show results without and with intermediate pruning, respectively. Rows 3, 4 and 5 list the relevant results for antithetic branching, antithetic branching with pruning and Latin hypercube sampling, respectively. For the two-asset example in Table 1, all the techniques include pruning at the last step.[12] Also, the European call option price with initial stock price $S_0^{(i)}$ and time to maturity 1 year was used as a control variate with all the techniques.

The first row was allowed 100 simulation runs, and the other rows were given approximately

[11] The European price is computed using the formula provided by Johnson [1987].

[12] Henceforth, we will refer to "intermediate pruning" as just "pruning."

the same amount of CPU time as the first row.[13] The exact timing for each row on an IBM AIX Version 3 for RISC System/6000 (RS/6000, for short) is indicated in each of the tables. Confidence intervals were computed as explained in Section 2. Because these intervals are conservative, the actual coverage is usually much greater than the nominal coverage. In Table 1, the values labeled "True" were obtained using the lattice approach suggested by Boyle, Evnine and Gibbs [1989] with 800 time steps on the same option with four exercise opportunities.[14] The values labeled "Point est" are the averages of the corresponding high and low estimators. Taking the midpoint as the price estimate is a fairly arbitrary way of compromising between the two. Nevertheless, the relative error ("Rel error") was computed from this estimate.[15] Since evaluating the true price for the five-asset case using any other numerical method seems to be computationally infeasible, in Table 2 we report interval estimates and an estimated relative error ("Est rel error"), which in this case is defined to be the ratio of half the width of the confidence interval to the midpoint.

It can be seen from Tables 1 and 2 that pruning certainly reduces the width of the confidence interval, although the point estimate itself seems to be getting worse in most situations. In this regard, it should be emphasized that all the savings obtained by using pruning was devoted to increasing the number of simulation runs. These savings could very well have been used to increase b, the number of branches per node. The former approach was taken to simplify the comparison between different techniques.

Contrasting rows 3 and 4 for the two-asset example in Table 1 with rows 1 and 2, it is clear that the antithetic branching technique brings the high and low estimates closer to each other, and it also reduces their standard errors. The net effect is a significantly narrower confidence interval. Looking at the two-asset example, it is hard to say whether antithetic branching combined with pruning is better than plain antithetic branching or not. Although, for the five-asset example in Table 2, antithetic branching and pruning together seem to perform somewhat better.

Row 5 in Tables 1 and 2 list the relevant results for Latin hypercube sampling (henceforth, LHS). Clearly, the confidence intervals have shrunk by at least a factor of five in comparison with pruning, i.e., row 2. The reason being that, as previously conjectured, LHS also seems to have the dual advantage of antithetic branching: both bias and variance appear to decrease. In particular, the bias for the high estimator is dramatically reduced. It therefore seems plausible that assigning a

[13] In UNIX jargon, this is the actual *user time* as opposed to *real time*.

[14] Since the lattice approach with 800 time steps is not accurate to three decimal places, the relative errors may be slightly inaccurate.

[15] The relative error was computed using numbers with higher precision than what are reported in the tables.

higher weight to the high estimate will produce a better point estimate for LHS. Furthermore, the confidence intervals corresponding to LHS and antithetic branching (with pruning) have about the same widths, although the former seems to be marginally better.

Due to the additional time required to carry out the random permutations in LHS method, it seems that in higher dimensions, antithetic branching may dominate. Comparing rows 4 and 5 in Tables 1 and 2, for instance, we see that the advantage of LHS over antithetic branching goes down as the number of underlying assets increase from two to five.

Example 2. As stated earlier, it is computationally hard to use lattice or any other method to find the price for the option on the maximum of five assets in the preceding example. We therefore considered a second five-dimensional example to corroborate the effectiveness of the techniques presented in this paper. This is an option on the geometric average of five assets. The payoff, in other words, is determined as $\max\{(S^{(1)}S^{(2)}S^{(3)}S^{(4)}S^{(5)})^{\frac{1}{5}} - K,\ 0\}$. The results are compiled in Table 3. Values for various parameters are listed at the bottom of the table. Since the geometric mean of lognormal random variables is lognormal, this problem can be reduced to one in a single variable,[16] and this was the approach taken in order to evaluate the option prices using a lattice with 10,000 steps; see the column corresponding to "True value." Additionally, the price of the European option on the single variable was used both for pruning and as a control variate. To test the simulation methodology, however, we continued to treat this example as a five-dimensional problem. The small relative errors indicate that the three techniques, namely antithetic branching, antithetic branching combined with pruning and Latin hypercube sampling, work extremely well. There does not seem to be a clear pattern though regarding which one of the three techniques performs the best, although the smallest confidence intervals are obtained via LHS.

Example 3. Finally, we considered an example with a more involved payoff function; in particular, the payoff is given as $S^{(5)}\max\{S^{(1)} + S^{(2)} + S^{(3)} - S^{(4)} - K,\ 0\}$, which has some of the features of quanto, basket and spread options. In Table 4, $S_0^{(5)}$ was fixed at 1, while all other S_0's were initially taken to be equal and they were allowed to vary from 70 (deep out-of-money) to 130 (deep in-the-money). Other parameter values are listed below the table. Since, the European value cannot be obtained analytically, pruning was done only when the immediate exercise value was 0. Also, two control variates were used: $S^{(5)}$ and $S^{(1)} + S^{(2)} + S^{(3)} - S^{(4)}$. Despite the complicated payoff function, the three techniques prove to be quite effective as indicated by the small estimated relative

[16]The interested reader should see Boyle [1993] for complete details.

errors.

Extrapolation example. In order to illustrate the effectiveness of extrapolation, we considered the familiar problem of pricing the option on the maximum of two assets from Example 1 above with one difference: it is now permitted to be exercised at any point in time before expiration.

We used equation (7) to find the extrapolated price estimate from C_1, C_2 and C_3. Table 5 lists the final results. The two-period and three-period calls were evaluated using antithetic branching, pruning and control variate techniques. Since the one-period option is European, it was evaluated directly using its analytical formula. The high and low estimates were extrapolated separately to get an interval for the actual option price. The midpoints of these intervals were taken to be the extrapolated price estimate. These point estimates were then compared with "true" values obtained using 500 time steps with as many exercise opportunities via the lattice approach suggested by Boyle, Evnine and Gibbs [1989]. It can be seen from the table that the relative errors are less than 1% for all cases except for the deep out-of-money option (and in this case the absolute error is quite small). Higher accuracy can be achieved by increasing the number of branches and using more simulation time for evaluating each of the calls individually. Further improvement, at the expense of increased simulation time, is possible by extrapolating C_1, C_2, C_3 and C_4 instead of C_1, C_2 and C_3.

It is useful to see how effective this extrapolation technique is over many sample runs, instead of just one. In Table 6 therefore we computed the root-mean-squared (RMS) error over 100 sample runs. Specifically, each option was priced 100 times using simulation starting from different seeds for the random variable generator. The resulting 100 extrapolated price estimates were used to determine the RMS error. Once again, the relative errors are less than 1% for all options except for the deep out-of-money option.

8 SUMMARY

As the number of state variables increases, simulation becomes the only computationally feasible numerical approach for pricing options. Also, unlike other numerical methods, it allows for complex payoff functions and path dependencies. It therefore becomes important to look for possible ways of using simulation for pricing American options.

Since the approach presented in this paper relies on two estimates, one biased high and one biased low, and a confidence interval obtained from them, it is vital to reduce the bias and variance of these estimates. We find antithetic branching, both by itself and in combination with pruning,

and Latin hypercube sampling work remarkably well to this end. Including a control variate causes further enhancement.

The simulation methodology is however exponential in the number of exercise opportunities. To price an option which can be exercised at any time before maturity, therefore, it is necessary to use extrapolation techniques. We find that Richardson extrapolation works reasonably well without significantly increasing the time required.

REFERENCES

Boyle, P.P. "Options: A Monte Carlo Approach," *Journal of Financial Economics*, 4 (1977), 323–338.

Boyle, P.P. "New Life Forms on the Options Landscape," *Journal of Financial Engineering*, 2 (1993), 217–252.

Boyle, P.P., J. Evnine, and S. Gibbs. "Numerical Evaluation of Multivariate Contingent Claims," *The Review of Financial Studies*, 2 (1989), 241–250.

Broadie, M., and P. Glasserman. "Pricing American-Style Securities Using Simulation," working paper, Columbia Business School, 1995a, to appear in the *Journal of Economic Dynamics and Control.*

Broadie, M., and P. Glasserman. "A Pruned and Bootstrapped American Option Simulator," Proceedings of the Winter Simulation Conference, Society of Computer Simulation, San Diego, CA (1995b), 229–235.

Cox, J., S. Ross, and M. Rubinstein. "Option Pricing: A Simplified Approach," *Journal of Financial Economics*, 7 (1979), 229–264.

Geske, R., and H.E. Johnson. "The American Put Option Valued Analytically," *Journal of Finance*, 39 (1984), 1511–1524.

Hull, J. *Options, Futures, and Other Derivative Securities.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

Johnson, H. "Options on the Maximum or the Minimum of Several Assets," *Journal of Financial and Quantitative Analysis*, 22 (1987), 227–283.

Karatzas, I. "Optimization Problems in the Theory of Continuous Trading," *SIAM Journal of Control and Optimization*, 27 (1989), 1221–1259.

McKay, M.D., W.J. Conover, and R.J. Beckman. "A Comparison of Three Methods for Selecting

Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, 21 (1979), 239–245.

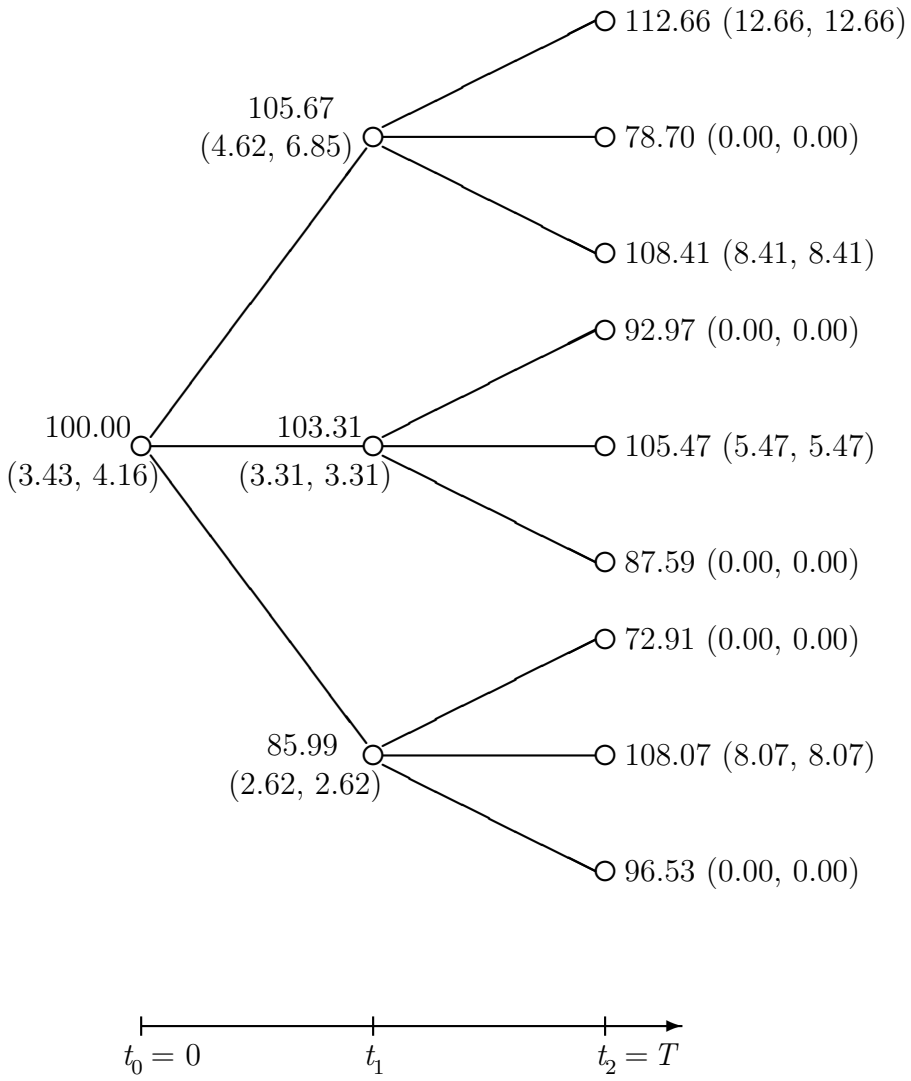

Figure 1: A tree with $b = 3$ and exercise opportunities at t_0, t_1 and t_2 for a call option; the value at each node is the stock price, and the values in parenthesis are the low and high estimates (θ, Θ).

Parameters: $M = 1$, $b = 3$, $K = 100$, $r = 5\%$, $\delta = 10\%$, $T = 1.0$, $\sigma = 20\%$, and three exercise opportunities at times 0, $T/2$, and T.

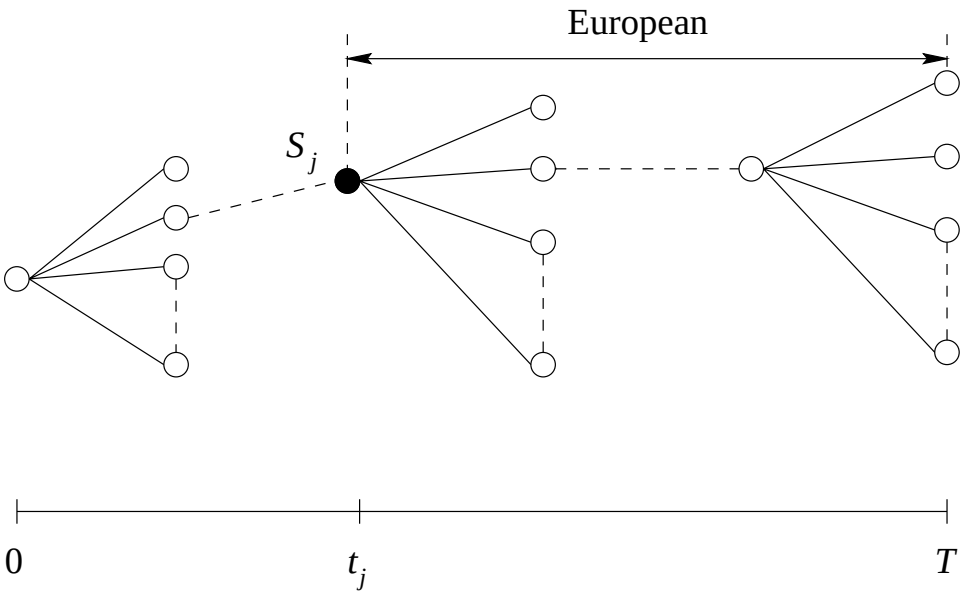

Figure 2: Pruning at time t_j: if the European value for the option initiated at time t_j with initial stock price S_{t_j} is smaller than the immediate exercise value, branch in the usual fashion.

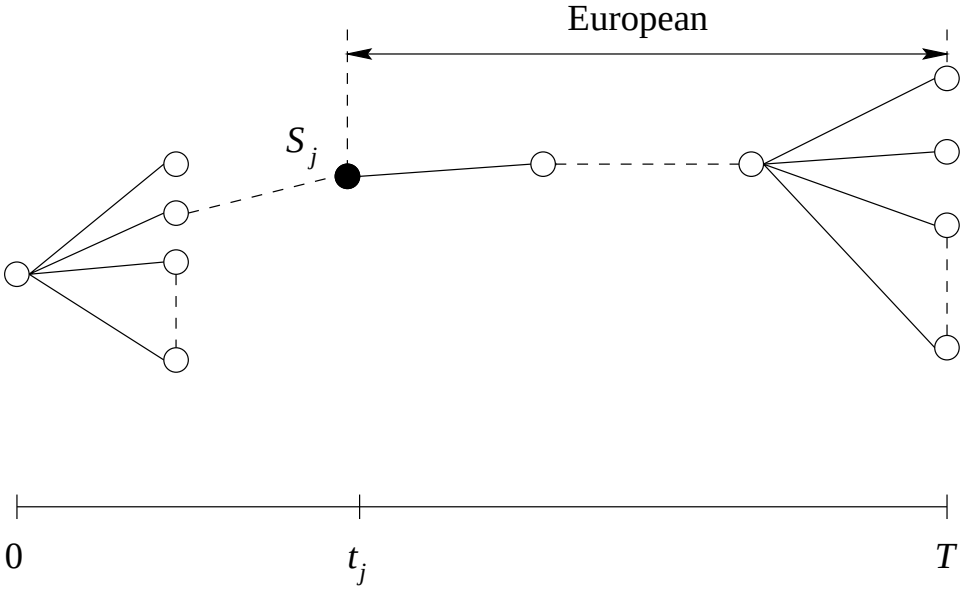

Figure 3: Pruning at time t_j: if the European value for the option initiated at time t_j with initial stock price S_{t_j} is greater than the immediate exercise value, generate only one successor node at the next time step t_{j+1}.

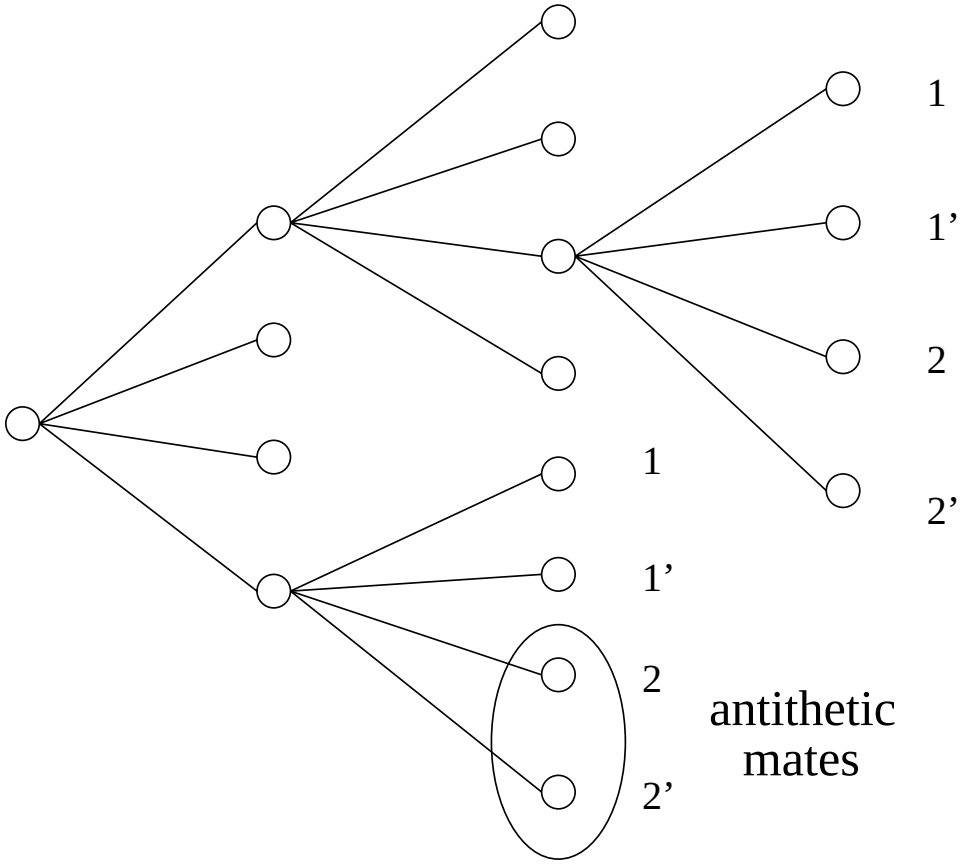

Figure 4: Illustration of antithetic branching; $b = 4$ and $1'$ is the antithetic mate of 1, etc.

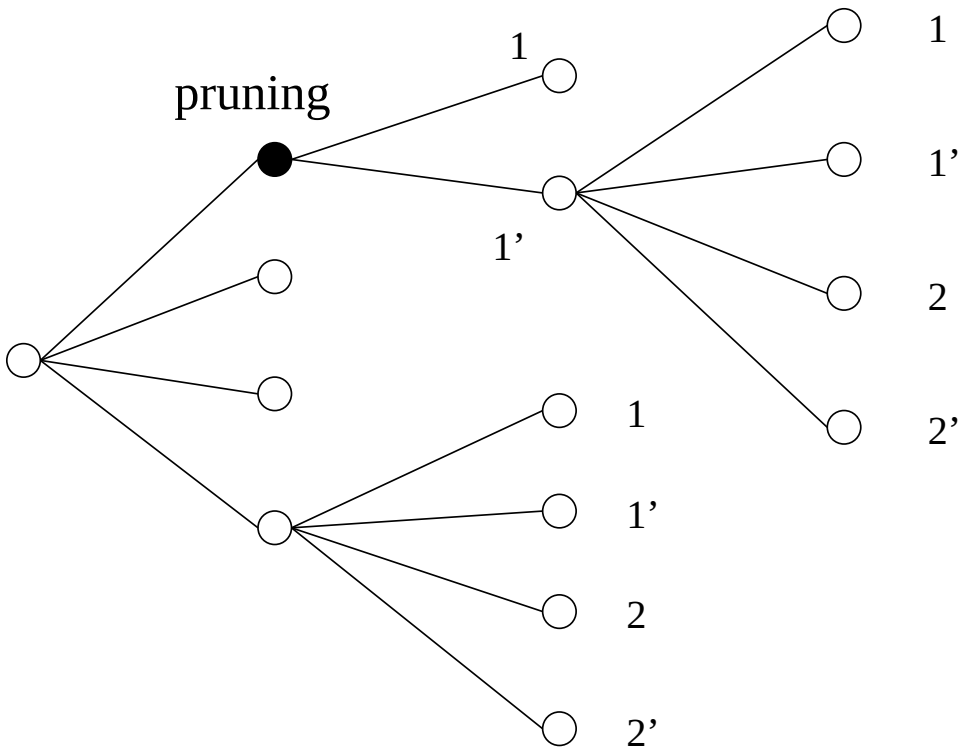

Figure 5: Illustration of antithetic branching combined with pruning; $b = 4$ and $1'$ is the antithetic mate of 1, etc.

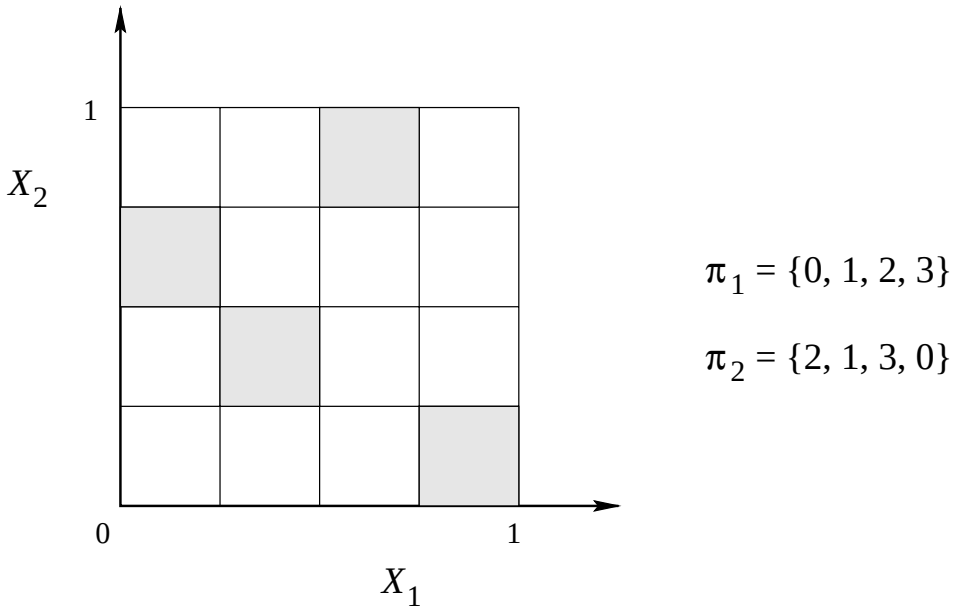

Figure 6: Illustration of picking branches from Latin hypercube samples for $b = 4$ and $M = 2$. The two random permutations, $\pi_1 = \{0, 1, 2, 3\}$ and $\pi_2 = \{2, 1, 3, 0\}$, imply that the four branches must be picked from the four colored squares.

Table 1: Option on maximum of two assets

S_0		Low est	Std error	High est	Std error	90% confidence interval	Point est	True value	Rel error
80	a	1.267	0.005	1.268	0.005	[1.259, 1.276]	1.267	1.259	0.67%
	b	1.257	0.002	1.258	0.002	[1.253, 1.262]	1.257	1.259	0.13%
	c	1.258	0.003	1.258	0.003	[1.253, 1.263]	1.258	1.259	0.07%
	d	1.257	0.002	1.258	0.002	[1.254, 1.260]	1.257	1.259	0.13%
	e	1.256	0.002	1.256	0.002	[1.253, 1.259]	1.256	1.259	0.24%
90	a	4.066	0.008	4.090	0.006	[4.052, 4.100]	4.078	4.079	0.02%
	b	4.069	0.004	4.085	0.004	[4.062, 4.092]	4.077	4.079	0.04%
	c	4.076	0.003	4.081	0.003	[4.070, 4.085]	4.078	4.079	0.01%
	d	4.076	0.003	4.080	0.003	[4.072, 4.084]	4.078	4.079	0.01%
	e	4.074	0.003	4.076	0.003	[4.069, 4.081]	4.075	4.079	0.09%
100	a	9.317	0.014	9.419	0.009	[9.295, 9.435]	9.368	9.358	0.11%
	b	9.345	0.009	9.417	0.008	[9.331, 9.431]	9.381	9.358	0.25%
	c	9.350	0.006	9.370	0.005	[9.340, 9.379]	9.360	9.358	0.02%
	d	9.361	0.005	9.378	0.005	[9.353, 9.386]	9.370	9.358	0.12%
	e	9.354	0.004	9.364	0.004	[9.348, 9.370]	9.359	9.358	0.01%
110	a	16.794	0.022	17.048	0.014	[16.758, 17.071]	16.921	16.925	0.02%
	b	16.876	0.014	17.044	0.012	[16.853, 17.064]	16.960	16.925	0.21%
	c	16.907	0.008	16.946	0.006	[16.895, 16.957]	16.927	16.925	0.01%
	d	16.904	0.008	16.941	0.008	[16.891, 16.954]	16.923	16.925	0.01%
	e	16.912	0.005	16.930	0.005	[16.904, 16.938]	16.921	16.925	0.02%
120	a	25.783	0.027	26.163	0.017	[25.738, 26.191]	25.973	25.979	0.03%
	b	25.898	0.019	26.147	0.016	[25.867, 26.173]	26.023	25.979	0.17%
	c	25.952	0.009	26.010	0.007	[25.938, 26.022]	25.981	25.979	0.01%
	d	25.941	0.011	25.996	0.010	[25.923, 26.013]	25.969	25.979	0.04%
	e	25.953	0.006	25.983	0.005	[25.943, 25.992]	25.968	25.979	0.04%

a Original approach with control variate

b Pruning and control variate

c Antithetic branching and control variate

d Antithetic branching, pruning and control variate

e Latin hypercube sampling and control variate

(All of them include pruning at last step; European price was used as control variate; Simulation time was approximately 1.7 minutes per row on an RS 6000 machine.)

Payoff: $\max\{\max\{S^{(1)}, S^{(2)}\} - K,\ 0.0\}$

Parameters: $M = 2$, $b = 50$, $K = 100$, $r = 5\%$, $\delta = 10\%$, $T = 1.0$, $\sigma = 20\%$, $\rho = 0.3$, and four exercise opportunities at times 0, $T/3$, $2T/3$, and T

Table 2: Option on maximum of five assets

S_0		Low est	Std error	High est	Std error	90% confidence interval A	90% confidence interval B	Point est C	Est rel error $\frac{B-A}{2C}$
80	a	2.686	0.002	2.729	0.001	[2.682,	2.731]	2.707	0.90%
	b	2.700	0.000	2.720	0.000	[2.699,	2.721]	2.710	0.40%
	c	2.700	0.001	2.712	0.001	[2.699,	2.713]	2.706	0.26%
	d	2.704	0.000	2.710	0.000	[2.704,	2.710]	2.707	0.12%
	e	2.702	0.001	2.709	0.001	[2.701,	2.711]	2.706	0.18%
90	a	7.731	0.005	7.890	0.003	[7.722,	7.895]	7.810	1.11%
	b	7.794	0.002	7.874	0.002	[7.791,	7.877]	7.834	0.56%
	c	7.801	0.002	7.836	0.002	[7.797,	7.840]	7.819	0.27%
	d	7.810	0.001	7.832	0.001	[7.809,	7.833]	7.821	0.16%
	e	7.811	0.003	7.831	0.003	[7.807,	7.836]	7.821	0.18%
100	a	15.733	0.010	16.032	0.007	[15.717,	16.043]	15.882	1.03%
	b	15.816	0.006	16.015	0.004	[15.806,	16.022]	15.915	0.67%
	c	15.860	0.004	15.923	0.004	[15.853,	15.929]	15.892	0.24%
	d	15.870	0.002	15.917	0.002	[15.866,	15.920]	15.893	0.17%
	e	15.869	0.005	15.908	0.004	[15.861,	15.915]	15.888	0.17%
110	a	25.558	0.016	26.004	0.010	[25.532,	26.020]	25.781	0.95%
	b	25.695	0.010	26.001	0.007	[25.678,	26.013]	25.848	0.65%
	c	25.765	0.005	25.845	0.005	[25.756,	25.853]	25.804	0.19%
	d	25.772	0.004	25.844	0.003	[25.766,	25.850]	25.808	0.16%
	e	25.767	0.006	25.823	0.006	[25.757,	25.832]	25.795	0.15%
120	a	36.180	0.017	36.740	0.012	[36.153,	36.760]	36.460	0.83%
	b	36.291	0.013	36.699	0.009	[36.268,	36.714]	36.494	0.61%
	c	36.446	0.006	36.546	0.006	[36.435,	36.555]	36.492	0.17%
	d	36.457	0.005	36.544	0.004	[36.448,	36.551]	36.502	0.14%
	e	36.445	0.008	36.518	0.007	[36.432,	36.530]	36.481	0.14%

a Original approach with control variate

b Pruning and control variate

c Antithetic branching and control variate

d Antithetic branching, pruning and control variate

e Latin hypercube sampling and control variate

(European price was used as control variate;

Simulation time was approximately 10.5 minutes per row on an RS 6000.)

Payoff: $\max\{\max\{S^{(1)}, S^{(2)}, S^{(3)}, S^{(4)}, S^{(5)}\} - K,\ 0.0\}$

Parameters: $M = 5$, $b = 50$, $K = 100$, $r = 5\%$, $\delta = 10\%$, $T = 1.0$, $\sigma = 20\%$, $\rho = 0.3$, and four exercise opportunities at times 0, $T/3$, $2T/3$, and T

Table 3: Option on geometric average of five assets

S_0		Low est	Std error	High est	Std error	90% confidence interval	Point est	True value	Rel error
70	a	0.518	0.000	0.520	0.000	[0.517, 0.521]	0.519	0.519	0.06%
	b	0.519	0.000	0.520	0.000	[0.519, 0.520]	0.519	0.519	0.01%
	c	0.519	0.000	0.519	0.000	[0.519, 0.520]	0.519	0.519	0.05%
80	a	1.661	0.001	1.672	0.001	[1.658, 1.673]	1.666	1.666	0.03%
	b	1.663	0.000	1.669	0.000	[1.663, 1.669]	1.666	1.666	0.03%
	c	1.664	0.001	1.666	0.001	[1.663, 1.667]	1.665	1.666	0.04%
90	a	3.988	0.003	4.018	0.002	[3.982, 4.022]	4.003	4.003	0.01%
	b	3.996	0.001	4.015	0.001	[3.994, 4.016]	4.005	4.003	0.06%
	c	3.996	0.002	4.003	0.001	[3.993, 4.006]	3.999	4.003	0.08%
100	a	7.831	0.006	7.897	0.004	[7.820, 7.904]	7.864	7.869	0.07%
	b	7.850	0.003	7.895	0.002	[7.845, 7.899]	7.872	7.869	0.04%
	c	7.856	0.003	7.871	0.002	[7.851, 7.875]	7.864	7.869	0.07%
110	a	13.334	0.009	13.430	0.007	[13.319, 13.442]	13.382	13.378	0.03%
	b	13.333	0.006	13.412	0.005	[13.324, 13.420]	13.373	13.378	0.04%
	c	13.357	0.005	13.384	0.004	[13.349, 13.391]	13.370	13.378	0.06%
120	a	20.227	0.058	20.572	0.018	[20.132, 20.602]	20.399	20.386	0.07%
	b	20.212	0.052	20.635	0.018	[20.127, 20.664]	20.423	20.386	0.19%
	c	20.361	0.007	20.398	0.005	[20.350, 20.407]	20.380	20.386	0.03%
130	a	30.000	0.000	30.013	0.009	[30.000, 30.027]	30.006	30.000	0.02%
	b	30.000	0.000	30.041	0.011	[30.000, 30.059]	30.021	30.000	0.07%
	c	30.000	0.000	30.000	0.000	[30.000, 30.000]	30.000	30.000	0.00%

a Antithetic branching and control variate

b Antithetic branching, pruning and control variate

c Latin hypercube sampling and control variate
(European price was used as control variate;
Simulation time was 10 minutes per row on an RS 6000 machine.)

Payoff: $\max\{(S^{(1)}S^{(2)}S^{(3)}S^{(4)}S^{(5)})^{\frac{1}{5}} - K,\ 0.0\}$

Parameters: $M = 5$, $b = 50$, $K = 100$, $r = 5\%$, $\delta_1 = \delta_2 = 6\%$, $\delta_3 = \delta_4 = \delta_5 = 8\%$, $T = 1.0$, $\sigma_1 = \sigma_2 = \sigma_3 = 40\%$, $\sigma_4 = \sigma_5 = 30\%$, $\rho_{12} = \rho_{15} = 0.8$, $\rho_{13} = \rho_{23} = 0.2$, $\rho_{14} = \rho_{24} = 0.3$, $\rho_{25} = \rho_{34} = 0.1$, $\rho_{35} = \rho_{45} = 0.7$, and four exercise opportunities at times 0, $T/3$, $2T/3$, and T

Table 4: Third example with five assets

S_0		Low est	Std error	High est	Std error	90% confidence interval A	90% confidence interval B	Point est C	Est rel error $\frac{B-A}{2C}$
70	a	0.520	0.004	0.522	0.004	[0.513,	0.530]	0.521	1.59%
	b	0.512	0.001	0.514	0.001	[0.510,	0.516]	0.513	0.54%
	c	0.520	0.006	0.521	0.006	[0.511,	0.531]	0.521	1.89%
80	a	2.309	0.011	2.323	0.011	[2.290,	2.342]	2.316	1.11%
	b	2.289	0.001	2.303	0.001	[2.287,	2.305]	2.296	0.38%
	c	2.315	0.011	2.323	0.011	[2.297,	2.341]	2.319	0.94%
90	a	6.679	0.021	6.731	0.021	[6.644,	6.766]	6.705	0.90%
	b	6.610	0.001	6.665	0.001	[6.609,	6.666]	6.637	0.43%
	c	6.693	0.017	6.729	0.017	[6.666,	6.757]	6.711	0.68%
100	a	14.517	0.031	14.647	0.031	[14.466,	14.698]	14.582	0.79%
	b	14.434	0.000	14.571	0.000	[14.433,	14.572]	14.503	0.48%
	c	14.564	0.022	14.641	0.022	[14.529,	14.677]	14.603	0.51%
110	a	26.024	0.038	26.220	0.037	[25.962,	26.282]	26.122	0.61%
	b	26.048	0.000	26.257	0.000	[26.048,	26.257]	26.153	0.40%
	c	26.025	0.025	26.171	0.024	[25.985,	26.211]	26.098	0.43%
120	a	40.289	0.053	41.087	0.035	[40.201,	41.144]	40.688	1.16%
	b	40.432	0.000	41.212	0.000	[40.432,	41.212]	40.822	0.96%
	c	40.571	0.028	40.887	0.023	[40.525,	40.925]	40.729	0.49%
130	a	60.000	0.000	60.090	0.015	[60.000,	60.115]	60.045	0.10%
	b	60.000	0.000	60.100	0.000	[60.000,	60.100]	60.050	0.08%
	c	60.000	0.000	60.006	0.002	[60.000,	60.009]	60.003	0.01%

a Antithetic branching and two control variates

b Antithetic branching, pruning and two control variates

c Latin hypercube sampling and two control variates
($S^{(5)}$ and $S^{(1)} + S^{(2)} + S^{(3)} - S^{(4)}$ were the two control variates;
Pruning was done only when immediate exercise value was 0;
Simulation time was 10 minutes per row on an RS 6000 machine.)

Payoff: $S^{(5)} \max\{S^{(1)} + S^{(2)} + S^{(3)} - S^{(4)} - K,\ 0.0\}$

Parameters: $S_0^{(5)} = 1$, $M = 5$, $b = 50$, $K = 200$, $r = 5\%$, $T = 1.0$, $\delta_1 = \delta_2 = 8\%$, $\delta_3 = \delta_4 = 10\%$, $\delta_5 = 12\%$, $\sigma_1 = \sigma_2 = \sigma_3 = 25\%$, $\sigma_4 = 20\%$, $\sigma_5 = 30\%$, $\rho_{34} = 0.4$, $\rho_{14} = \rho_{24} = 0.3$, $\rho_{12} = \rho_{13} = \rho_{23} = 0.2$, $\rho_{15} = \rho_{25} = \rho_{35} = \rho_{45} = 0.1$, and four exercise opportunities at times 0, $T/3$, $2T/3$, and T

Table 5: Extrapolation example

	1 period	2 period		3 period		Extrapolation			True	Rel
S_0		Low	High	Low	High	Low	High	Mid	value	error
70	0.234	0.236	0.236	0.237	0.237	0.242	0.242	0.242	0.245	1.29%
		(0.001)	(0.001)	(0.001)	(0.001)					
80	1.233	1.244	1.244	1.258	1.258	1.301	1.302	1.302	1.302	0.03%
		(0.002)	(0.002)	(0.001)	(0.001)					
90	3.939	4.029	4.029	4.073	4.077	4.182	4.200	4.191	4.215	0.57%
		(0.002)	(0.002)	(0.002)	(0.002)					
100	8.932	9.248	9.248	9.348	9.366	9.541	9.622	9.582	9.637	0.57%
		(0.002)	(0.002)	(0.004)	(0.004)					
110	16.029	16.726	16.726	16.903	16.937	17.176	17.327	17.251	17.349	0.56%
		(0.003)	(0.003)	(0.007)	(0.007)					
120	24.572	25.671	25.671	25.978	26.030	26.502	26.739	26.621	26.548	0.27%
		(0.003)	(0.003)	(0.010)	(0.009)					
130	33.902	35.362	35.362	35.720	35.791	36.244	36.561	36.403	36.455	0.14%
		(0.003)	(0.003)	(0.011)	(0.011)					

Antithetic branching, pruning and control variate techniques were used to evaluate the 2- and 3-period options. C_1 was evaluated using the formula derived in Johnson [1987]. App. 4 minutes were devoted for each row, and the low and high estimates were extrapolated separately. The values in parentheses are the respective standard errors. The relative error has been computed using the midpoint as the estimated option price.

Payoff: $\max\{\max\{S^{(1)}, S^{(2)}\} - K,\ 0.0\}$

Parameters: $M = 2$, $b = 50$, $K = 100$, $r = 5\%$, $\delta = 10\%$, $T = 1.0$, $\sigma = 20\%$, $\rho = 0.3$, and infinite exercise opportunities

Table 6: Effectiveness of extrapolation

S_0	RMS error
70	1.80%
80	0.48%
90	0.44%
100	0.61%
110	0.38%
120	0.18%
130	0.16%

RMS error was computed over 100 sample runs. Other parameters are the same as in Table 5.