Probabilistic Constrained Optimization: Methodology and Applications (S. P. Uryasev, Editor), pp. 32-50

Pricing American Options by Simulation Using a Stochastic Mesh with Optimized Weights

Mark Broadie (mnb2@columbia.edu)
Columbia Business School
New York, NY 10027

Paul Glasserman (pg20@columbia.edu)
Columbia Business School
New York, NY 10027

Zachary Ha (zachary.ha@gs.com)
Goldman Sachs & Co.
New York, NY 10005

Abstract

This paper develops a simulation method for pricing path-dependent American options, and American options on a large number of underlying assets, such as basket options. Standard numerical procedures (lattice methods and finite difference methods) are generally inapplicable to such high-dimensional problems, and this has motivated research into simulation-based methods. The optimal stopping problem embedded in the pricing of American options makes this a nonstandard problem for simulation.

This paper extends the *stochastic mesh* introduced in Broadie and Glasserman [5]. In its original form, the stochastic mesh method required knowledge of the transition density of the underlying process of asset prices and other state variables. This paper extends the method to settings in which the transition density is either unknown or fails to exist. We avoid the need for a transition density by choosing mesh weights through a constrained optimization problem. If the weights are constrained to correctly price sufficiently many simple instruments, they can be expected to work well in pricing a more complex American option. We investigate two criteria for use in the optimization — maximum entropy and least squares. The methods are illustrated through numerical examples.

1 Introduction

Computational methods for pricing derivative securities can be broadly divided into deterministic methods and simulation-based methods. The first type generally involves discretizing time and discretizing the possible levels of the underlying asset prices; the discrete approximation is then solved exactly. Well-known examples of this approach include binomial and trinomial lattices, and finite difference methods. These methods are widely used, particularly in valuing relatively simple derivative securities in relatively simple models (see, e.g., [13] for background).

Deterministic methods can be very fast and effective if the dimension of the state vector representing the underlying model is 1, 2, or perhaps 3. But the time and space requirements of these methods typically grow exponentially in the dimension, rendering these methods inapplicable to high-dimensional problems.

Simulation methods are based on stochastic sampling of paths of the underlying state vector. Their space requirements generally grow linearly in the dimension of the state vector. They typically converge in proportion to the square root of the number of paths generated, a convergence rate independent of the dimension of the problem. This makes simulation-based methods attractive for valuing path-dependent and multi-asset derivatives.

A complication arises, however, with simulation techniques in pricing option contracts with American-style features—i.e., contracts in which the holder can choose the time of exercise. In this case, an optimal exercise boundary has to be determined through some type of dynamic programming procedure. The difficulty arises in combining the forward evolution of simulation with backward induction of dynamic programming. Recently, several methods have been proposed to address this issue; see [1, 3, 4, 5, 6, 7, 9, 10, 12] and references there.

In this paper we further develop the *stochastic mesh* method introduced in Broadie and Glasserman [5]. This method simulates multiple paths in parallel and uses information from all paths to estimate the continuation value (the value of holding an option rather than exercising) at each node along each path. The continuation value at each node is estimated as a discounted weighted average of the option values at the next time step across all paths. In the original mesh method, the weights were computed from the transition density of the underlying process. For complex models the transition density may be unknown, and in *singular* problems the transition density fails to exist altogether.

An example of a singular problem is an American Asian option, an option whose payoff at exercise depends on the time-average price of the underlying asset. The singularity arises from the fact that the running average is a deterministic transformation of the path of the underlying asset. Singularities also arise in any model in which the number of driving factors is smaller than the number of state variables, as is typical in term structure models. For example, a term structure model may represent 80 interest rates (quarterly rates over 20 years) and yet be driven by just a

three-dimensional Brownian motion.

To address these issues, we develop a strategy for selecting weights that does not rely on the knowledge or existence of a transition density. We choose the weights through optimization subject to constraints such as matching moments of the underlying process. The goal is to choose the weights so that the mesh correctly prices simple instruments and then use those weights to price complex instruments. Because the number of constraints is typically much smaller than the number of weights, the problem is underdetermined and we have to impose an optimization criterion to choose a particular set of weights. We investigate two criteria in particular — maximum entropy and least squares.

The rest paper of this paper is organized as follow. We give a general formulation of the problem in Section 2. Section 3 reviews the stochastic mesh method and explains the crucial step of calculating the weights. Optimization problems are then formulated in Section 4 for the new approach to choosing weights. In Section 5 we give various numerical examples to illustrate the methods.

2 Problem Formulation

We denote by $S_t = (S_t^1, \ldots, S_t^n)$ the vector of underlying state variables at time t and we assume that S_t is a Markov process. The Markov property can in most cases be enforced by introducing additional variables in the state vector, if necessary. The payoff (or in some cases the discounted payoff) from exercise at time t in state S_t is given by $h(t, S_t)$ for some function h. Path-dependent payoffs can again be accommodated by introducing additional state variables, if necessary. For example, if the payoff depends on the time-average, the maximum, or the minimum of one of the state variables, we can include the running average, the running maximum, or the running minimum in the state vector.

Perhaps the simplest interesting case of a model of this type is a multivariate lognormal process. In this case, $\log S_t$ is an n-dimensional Brownian motion process with a fixed covariance matrix Σ. If the process is simulated under the risk-neutral measure, each component $\log S_t^i$ has drift r, where r is the risk-free interest rate, assumed deterministic and constant. We return to this special case later to illustrate various methods.

We assume that exercise of the option is restricted to a finite set of dates and, for simplicitly, we assume that these dates are equally spaced Δt time units apart. (Options with a finite set of exercise opportunities are sometimes called "Bermudan" and the term "American" reserved for continuous exercise opportunities. We interpret "American" to refer to either case.) Letting $T = d\Delta t$ denote the option expiration date, the problem we seek to solve is finding

$$V(0, S_0) = \max_{\tau} E[h(\tau, S_\tau)],$$

with S_0 a given initial state and τ restricted to be a stopping time (with respect to the Markov process $\{S_{k\Delta t}, k = 0, 1, \ldots\}$) taking values in $\{0, \Delta t, \ldots, d\Delta t\}$. In an important special case of this problem, the expectation is with respect to the risk-neutral measure and h denotes a payoff discounted at the risk-free rate. More generally, the expectation could be with respect to some other martingale measure and h could be discounted by the corresponding numeraire asset, which would then be one of the state variables or possibly a transformation of the state variables.

We assume that the function h is explicitly available. Thus, the value $h(T, S_T)$ from exercise at expiration is available as a function of S_T. This leads to the following backward induction for determining the value at time 0:

$$\begin{aligned} V(T,s) &= h(T,s) \\ V(k\Delta t, s) &= \max\{h(k\Delta t, s), E[V((k+1)\Delta t, S_{(k+1)\Delta t})|S_{k\Delta t} = s]\}, \\ &\qquad k = d-1, d-2, \ldots, 1, 0. \end{aligned} \tag{1}$$

The first term inside the max operator is the immediate exercise value of the option and the second term (the conditional expectation) is the continuation value. Essentially all methods for pricing American options approximate this dynamic programming representation in some way. Simulation can be useful in estimating the continuation value since simulation is particularly well suited to estimating expectations.

3 Stochastic Mesh Method

The stochastic mesh method may be viewed as a particular way of choosing the points at which the immediate exercise and continuation values will be calculated and a particular way of estimating the continuation value. The method begins by generating nodes for the mesh. The next step is to construct weights for the transitions between nodes. Finally, the weights are used for estimating prices in the mesh. We give a general description of the method and illustrate it in the case of a lognormal process.

3.1 Mesh Construction

Construction of the mesh begins with simulation of multiple independent copies of the underlying process S_t, all from a common initial state S_0. A schematic representation of these paths is shown in Figure 1. In the figure, node i, for example, corresponds to the n-dimensional state vector $(S^1_{3\Delta t}, S^2_{3\Delta t}, \ldots, S^n_{3\Delta t})$. The state vector $(S^1_{4\Delta t}, S^2_{4\Delta t}, \ldots, S^n_{4\Delta t})$ at node j is generated from node i using whatever method would ordinarily be used to simulate the underlying state vector over a time period Δt. It must be stressed that the figure is purely schematic, with each path representing an

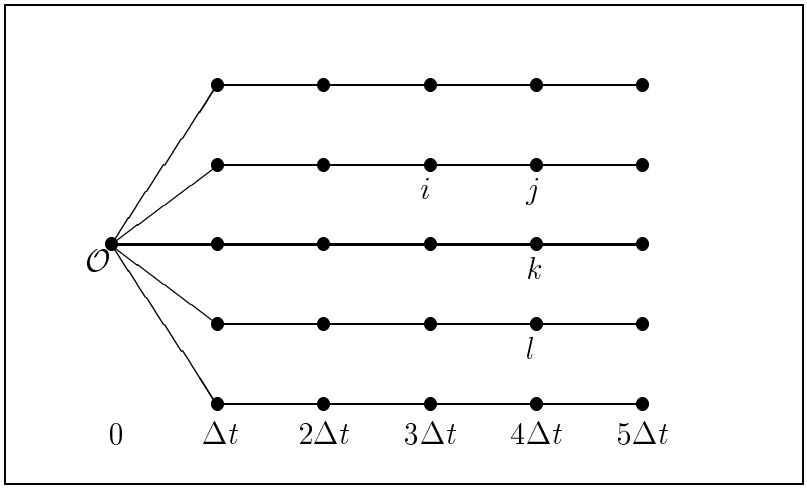

Figure 1: Stochastic mesh. Node i contains the vector $(S^1_{3\Delta t}, S^2_{3\Delta t}, \ldots, S^n_{3\Delta t})$.

independent simulated trajectory and all paths simulated from the same transition law. There is no sense in which node j is "higher" than node k, for example.

To be more concrete consider the following n-dimensional, m-factor lognormal processes

$$\frac{dS^i_t}{S^i_t} = r\,dt + \sum_{j=1}^{m} L_{ij}\,dW^j_t, \quad i = 1, \ldots, n.$$

Here, r is the risk-free interest rate, the W^j_t are independent standard Brownian motions, and L is an $n \times m$ matrix. The law of this process depends on L only through the instantaneous covariance matrix $\Sigma = LL^\top$. Paths of this process can be simulated using

$$S^i_{t+\Delta t} = S^i_t \exp\left((r - \frac{1}{2}\Sigma_{ii})\Delta t + \sqrt{\Delta t}\sum_{j=1}^{m} L_{ij}X_j\right), \quad i = 1, \ldots, n, \tag{2}$$

with the X_j's sampled from the standard normal distribution. Repeated use of the recursive relation in (2) (with independent X_j's) produces a single path for the mesh. Repeating this path generation procedure for multiple sets of independent X_j's produces a set of paths from which to construct the mesh.

3.2 Weights from a Transition Density

Once we have the paths for the mesh, we choose a set of weights w_{ij}, with w_{ij} denoting the weight attached to the transition from node i at one time slice to node j at the next time slice. The key feature of the mesh is that i and j need not be on the same path for w_{ij} to be nonzero. In fact, once we have generated the paths we deliberately "forget" which nodes were on the same paths and treat every node at time $(k+1)\Delta t$ as a potential successor of every node at time $k\Delta t$.

Given N nodes at each time slice and a set of weights w_{ij}, we approximate (1) using

$$V_i(k\Delta t) = \max\left(h_i(k\Delta t), \sum_{j=1}^{N} w_{ij} V_j((k+1)\Delta t)\right), \tag{3}$$

or

$$V_i(k\Delta t) = \max\left(h_i(k\Delta t), D_{\Delta t}^{-1} \sum_{j=1}^{N} w_{ij} V_j((k+1)\Delta t)\right),$$

with $D_{\Delta t}^{-1}$ a discount factor, depending on how the discounting is handled. Here, $V_i(k\Delta t)$ denotes the estimated value of the option at node i and time $k\Delta t$, and $h_i(k\Delta t)$ is the (explicitly available) immediate exercise value at node i and time $k\Delta t$. The weights w_{ij} will in general depend on the time index k; we suppress the time argument to simplify notation.

In the original mesh method of Broadie and Glasserman [5], the process S_t is assumed to have a known transition density and the weights are calculated from this transition density. To make this more explicit, suppose S_t satisfies

$$P(S_{t+\Delta t} \in A | S_t = x) = \int_A f(t, x, y)\, dy,$$

for some probability density $f(t, x, \cdot)$, all x, all t, and all (measurable) A. In the multivariate lognormal case (2), a transition density exists and is easily expressed in terms of the standard normal density provided the covariance matrix Σ has full rank.

For some fixed time t, let p_{ij} denote the value of the transition density from node i at time t to node j at time $t + \Delta t$. In [5], the weights are defined as

$$w_{ij} = \frac{p_{ij}}{\sum_k p_{kj}}. \tag{4}$$

This choice implies that

$$\sum_{i=1}^{N} w_{ij} = 1, \tag{5}$$

where N is the total number of mesh paths. Notice that here the destination node j is fixed and the sum is over the possible source nodes i. Because of (5), every node at time $t + \Delta t$ is assigned the same total weight; the nodes differ in how this total weight is distributed among the possible source nodes one time step earlier.

An important feature of (5) becomes evident in checking the price of a *European* option as estimated by the mesh. Of course, there is no need to use a mesh in the European case, but it is instructive to examine this case. For a European option, we replace (3) with $V_j(T) = h_j(T)$ and

$$V_i(k\Delta t) = \sum_{j=1}^{N} w_{ij} V_j((k+1)\Delta t);$$

we omit the max because in the European case the holder of the option no longer has the right to exercise early. A simple induction argument shows that in this case (5) implies

$$V(0) = \frac{1}{N} \sum_{j=1}^{N} h_j(T).$$

In other words, the mesh price telescopes to the average over the payoffs at the terminal nodes—precisely the same estimate that would be obtained from the original N paths using standard simulation rather than the mesh. The same is true if the payoff is discounted over each time step.

A natural alternative to (5) is the condition

$$\sum_{j=1}^{N} w_{ij} = 1. \tag{6}$$

Numerical experiments suggest that choosing the weights to enforce this constraint is less effective than enforcing (5), when the weights are defined from a transition density. In Section 4, where we choose weights through optimization procedures, (6) is a more convenient constraint.

3.3 High-Biased Estimator

Once the weights are determined it is straightforward to calculate the mesh price using (3). By repeating this step one obtains an estimate $V_{\mathcal{O}}(0)$ of the option value at the root node $\mathcal{O}$ of the mesh. In fact, through this procedure one also obtains an estimate of the option value and continuation value at every node in the mesh. This in turn implicitly yields an estimate of the exercise and continuation regions. At a node where the maximum in (3) is attained by the first term, the mesh estimates that it is optimal to exercise the option; wherever the maximum is attained by the second term, the mesh estimates that it is optimal to continue.

The option values estimated by the mesh through (3) tend to overestimate the true option price. This is a consequence of the convexity of the max function and Jensen's inequality. Taking conditional expectations E_i in (3) with respect to the state at node i yields

$$\begin{aligned} E_i[V_i(k\Delta t)] &= E_i\left[\max\left(h_i(k\Delta t), \sum_{j=1}^{N} w_{ij} V_j((k+1)\Delta t)\right)\right] \\ &\geq \max\left(h_i(k\Delta t), \sum_{j=1}^{N} w_{ij} E_i[V_j((k+1)\Delta t)]\right). \end{aligned}$$

This effect propagates backwards through the mesh and results in an estimate at time 0 that is biased high. For a more complete investigation of the properties of the mesh estimator, see Broadie and Glasserman [5].

3.4 Low-Biased Estimator

As in [5], the high-biased estimator can be combined with a low-biased estimator to produce an interval estimate for the option price. To generate the low-biased estimator, we generate additional independent paths through the mesh. These additional paths use the mesh only to determine *when* to exercise; the payoff assigned to each of these paths is the exact payoff upon exercise and not a value estimated from the mesh.

From the initial node $\mathcal{O}$ we begin simulation of a new path of S_t over time steps $\Delta t, 2\Delta t, \ldots, T$. At each step, to determine whether or not to stop, we calculate weights with respect to the new node on the path, use (3) to estimate the option value at that node and repeat this procedure until the estimated option value is equal to the exercise value, i.e. the estimated exercise region is reached. Upon exercise, we record a payoff by evaluating h. Figure 2 shows three paths a, b and c that are simulated and stopped when the exercise boundary (as estimated by the mesh) is reached.

The key observation here is that the original mesh determines the exercise decision at all possible states and not simply those corresponding to nodes in the mesh. Suppose for example that we have simulated a path to a state s at time $k\Delta t$. From state s (which is generally not a node in the mesh) we evaluate the transition density to each node j at time $(k+1)\Delta t$; call these weights w_{sj}. Using these weights we estimate the continuation value from state s as

$$\sum_{j=1}^{N} w_{sj} V_j((k+1)\Delta t)$$

using values V_j already calculated in the mesh. We also evaluate the immediate exercise value $h(s, k\Delta t)$ in state s. If the estimated continuation value exceeds the immediate exercise value, we continue simulating the path by generating a transition out of state s; if the immediate exercise value is greater, we stop and record a payoff of $h(s, k\Delta t)$. We repeat this procedure over many paths (all on the original mesh) and then average.

The exercise region determined by the mesh (and then used by the independent paths) is not in general the optimal exercise region. We know, however, that it cannot be better than the optimal exercise region. Thus, the average payoff generated in this way cannot be greater than the average payoff under the optimal policy. We may therefore conclude that the estimate produce by this second pass through the mesh is biased low. See [5] for a more complete investigation of this method and its properties.

The construction of the weights and the high- and low-biased estimators explained thus far apply in the case of a known transition density. For the special case of (2), this requires that the covariance matrix have full rank. To address settings in which the transition density is either unknown or fails to exist, we next develop a new approach to constructing weights.

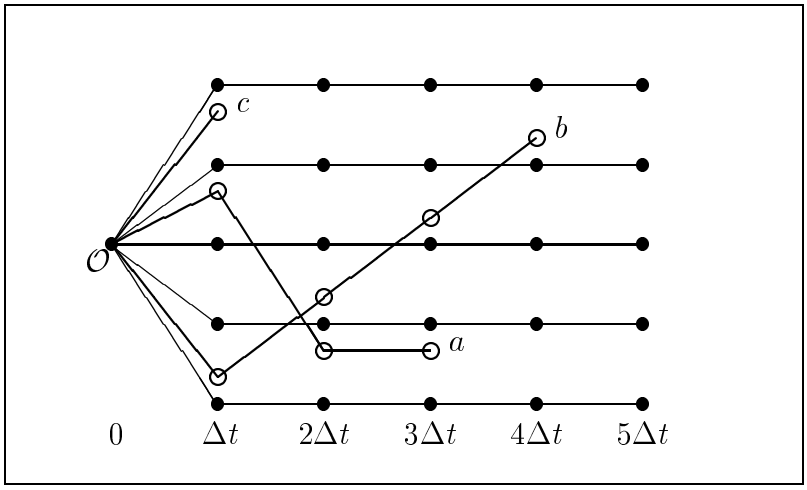

Figure 2: Schematic diagram of three randomly generated paths a, b and c for the low-biased estimator. The paths terminate upon entry into the exercise region estimated by the mesh.

4 Weights via Optimization

To obviate the need for a probability density, we instead formulate the problem of choosing "good" weights as a constrained optimization problem. The constraints we impose ensure that the mesh values of certain basic quantities — for example, low-order moments of the state variables — coincide with their theoretical values. In general, the number of constraints will be much smaller than the number of weights and the problem is underdetermined. We therefore impose an optimization criterion to choose a particular set of weights among all those satisfying the constraints. We investigate two criteria in particular: maximum entropy and least squares. Both of these criteria give preference to uniformity in the weights (subject to the constraints), which is attractive in view of the symmetry in the construction of the mesh. The maximum entropy criterion ensures nonnegativity of the weights; least squares does not but it is computationally easier to work with.

4.1 Maximum Entropy Weights

For a fixed node i, the entropy criterion is

$$L_0 = -\sum_{j=1}^{N} w_{ij} \log(w_{ij}). \tag{7}$$

This objective is maximized (subject to the sum constraint (6)) by the uniform distribution, i.e. $w_{ij} = 1/N$. However, to obtain "good" weights we impose further constraints, such as matching the first order and higher order moments for the underlying processes. The maximum entropy solution then corresponds to the "most

uniform" distribution satisfying the constraints. For a different application of entropy weights in pricing derivative securities, see Avellaneda et al. [2].

As an illustration of the types of constraints we use, consider, for example, the case of a single underlying asset with value $S_t(k)$ on path k at time t. Suppose $E[S_{t+\Delta t}|S_t] = e^{r\Delta t}S_t$, as in the lognormal case (2). Then we might impose the constraint that the w_{kj} satisfy

$$S_t(k) = e^{-r\Delta t}\sum_{j=1}^{N} w_{kj}S_{t+\Delta t}(j),$$

the sum taken over all nodes at time $t + \Delta t$. This ensures that the weights w_{kj} correctly "price" the underlying asset itself at node k.

Observe that in this example the constraint is linear in the weights. Suppose more generally that at each node i there are K linear constraints given by

$$\sum_{j=1}^{N} B_{kj}w_{ij} = b_k, \quad k = 1, \ldots, K, \tag{8}$$

where B is a $K \times N$ matrix and b a K-dimensional vector. (The matrix B and vector b will in general depend on the node i.) We incorporate these constraints in the optimization criterion at node i by setting

$$L = L_0 + \lambda_0(\sum_j w_{ij} - 1) + \sum_{k,j} \lambda_k(B_{kj} - b_k)w_{ij}, \tag{9}$$

where the λ_k's are Lagrange multipliers. Notice that we have also imposed (6) as a constraint.

By explicitly solving the following equations

$$\frac{\partial L}{\partial w_{ij}} = 0, \tag{10}$$

$$\frac{\partial L}{\partial \lambda_0} = 0, \tag{11}$$

we obtain w_{ij} in terms of the Lagrange multipliers and parameters for the constraints as

$$w_{ij} = \frac{\exp(\sum_k \lambda_k(B_{kj} - b_k))}{\sum_\ell \exp(\sum_k \lambda_k(B_{k\ell} - b_k))}. \tag{12}$$

Solving the rest of the constraints given by (8), therefore, amounts to minimizing the following function with respect to λ's ([8]):

$$\log\left(\sum_\ell \exp\left(\sum_k \lambda_k(B_{k\ell} - b_k)\right)\right). \tag{13}$$

We use the Newton-Raphson method to obtain the Lagrange multipliers that minimize this function and solve for the weights using (12). The number of constraints is usually much smaller than the number of weights N and thus the numerical optimization is viable. It should be clear from the presence of the logarithm in (7) that the maximum entropy weights are always positive.

4.2 Least Squares Method

The Taylor approximation $-\log(w) \approx 1 - w$ near $w = 1$ leads to the approximation $\sum_j w_{ij}(1 - w_{ij})$ for L_0 in (9). Because we also impose the constraint that $\sum_j w_{ij} = 1$, maximizing this approximation is equivalent to choosing weights through the least squares criterion of minimizing $\sum_j w_{ij}^2$. The least squares problem has the advantage that it can be solved explicitly, without the need for numerical optimization. By solving $\partial L/\partial w_{ij} = 0$ we find that the vector $w_i = (w_{i1}, \ldots, w_{iN})$ of weights out of node i satisfy

$$w_i = \frac{1}{2}\lambda B, \tag{14}$$

with $\lambda = (\lambda_1, \ldots, \lambda_k)$. By plugging this expression into (8) we obtain the weights in terms of the known parameters as

$$w_i = B^\top (BB^\top)^{-1} b. \tag{15}$$

Intuitively, the solution for the weights is simply obtained from (8) by inverting the retangular matrix B, and $B^\top(BB^\top)^{-1}$ may be viewed as the pseudo-inverse of B.

An advantage of this method is the improvement in speed. But, there is a drawback that the weights produced are not guaranteed to be nonnegative. If we imposed nonnegativity as a constraint, solving the least squares problem would again require numerical optimization and would therefore have little or no advantage over the maximum entropy criterion.

Longstaff and Schwartz [9] also use a least squares method in pricing American options by simulation. However, they use least squares in regressing an option's continuation value against a set of basis functions rather than to price simple instruments exactly.

5 Some Numerical Examples

In this section we give various examples to illustrate the methods presented in the previous sections.

5.1 One Dimensional Examples

In order to show how the mesh method works on cases with known solutions we first price a one-dimensional American put option using the original weights from

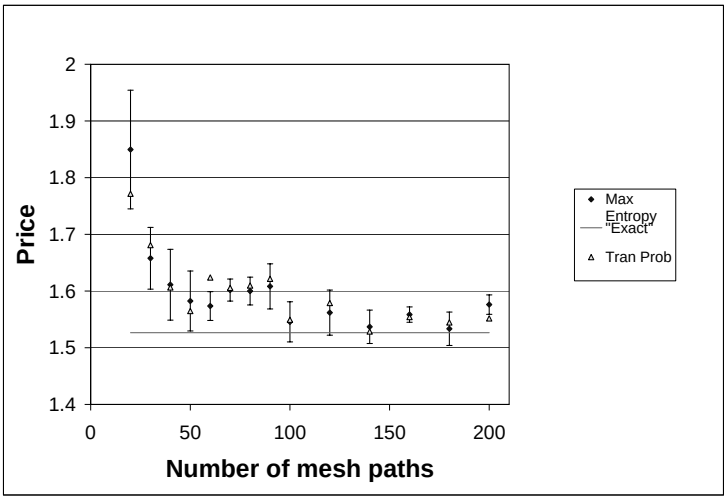

Figure 3: American put price on single asset. Exact price and the prices from weights estimated from the transition densities and the maximum entropy method are plotted. $S_0 = 40$, the strike is 40, the risk-free rate is 10%, the volatility is 20%, and the time to expiration is 5 years. There are five equally spaced exercise opportunities. Number of exercise dates = 5.

the transition probability densities and from the maximum entropy method and plot the high-biased estimators in Figure 3. The figure plots estimated prices against the number of mesh paths for an option with five exercise dates. The error bars show 95% confidence intervals around the maximum entropy estimate. The two methods are hard to distinguish from each other and they approach the true price (obtained from a binomial lattice) as the mesh size increases.

Next, we plot high-biased and low-biased estimates for the same option using the maximum entropy and the least squares methods and compare them with the exact price in Figure 4. The high-biased estimates obtained by the least squares method (LS-High) are higher than those obtained by the maximum entropy method (ME-High). The low-biased estimators are less distinguishable.

The large standand error in the two figures can be reduced if we use a larger mesh size. We use the least squares method to achieve this and plot the low-biased estimator in Figure 5. The reduction in standard error is clear. Although this method comes very close to the true value, it appears to slightly underestimate the true value. The estimated exercise region implicit in the mesh is thus slightly suboptimal.

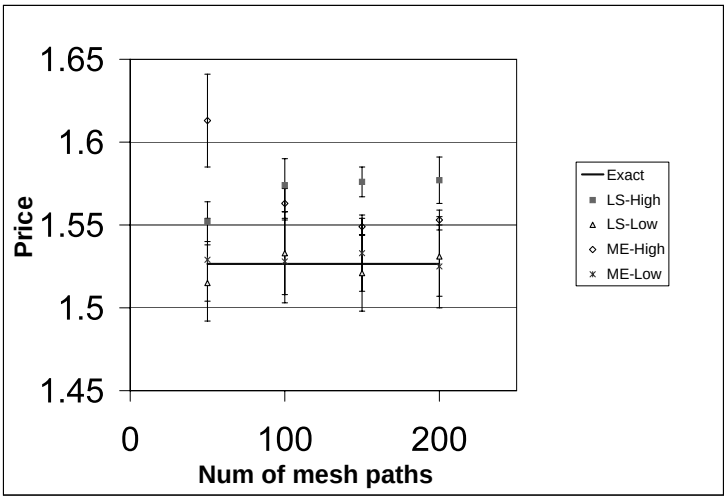

Figure 4: High- and low-biased estimators for the American put price estimated using the maximum entropy and the least squares methods.

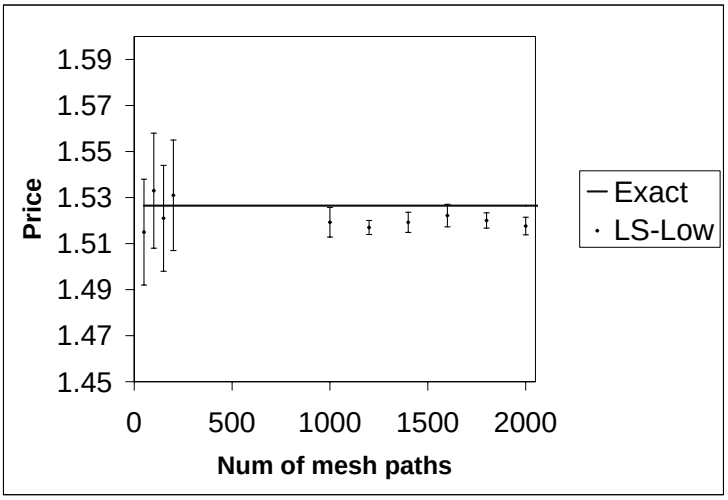

Figure 5: Low-biased estimator for the American put at large mesh sizes.

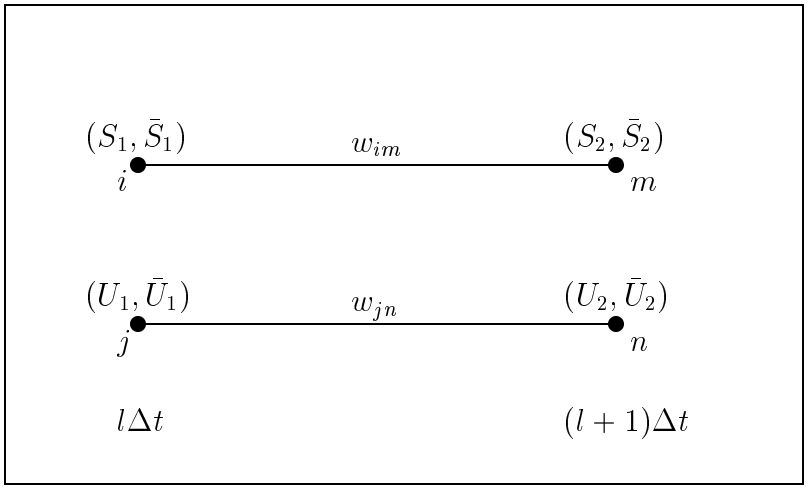

Figure 6: Illustration of singularity in American Asian option.

5.2 American Asian option

The payoff function for an American Asian option depends on the average price over the discrete exercise dates. For example, a possible payoff function for the American Asian put option is given by $\max(X - \bar{S}, 0)$ where X is the strike price and $\bar{S}$ is the average of $S_{k\Delta t}$ up to the current time. This option corresponds to a two-dimensional singular case since the underlying asset price and its average are perfectly correlated over a single time step. The problem is illustrated in Figure 6. In the mesh construction, $(S_2, \bar{S}_2)$ is simulated from $(S_1, \bar{S}_1)$ and similarly $(U_2, \bar{U}_2)$ is simulated from $(U_1, \bar{U}_1)$. When we attempt to interconnect the nodes we encounter a difficulty: if the underlying asset moves from S_1 to U_2, it is generally *not* the case that the running average moves from $\bar{S}_1$ to $\bar{U}_2$; indeed, this would happen only if $\bar{U}_2$ happens to equal $(l\bar{S}_1 + U_2)/(l+1)$, which occurs with probability zero. There is therefore no way to assign a weight w_{in} based on the probability density of moving from i to n.

Figure 7 shows prices obtained from the least-squares mesh method with two types of constraints and from a non-recombining bushy tree (used here as the benchmark). On the left the American and European prices are plotted against the number of time steps. The first pair of prices on the right denoted by a filled triangle and filled circle are Richardson extrapolated prices (see, e.g., [13] for background on Richardson extrapolation). The other two pairs on the right correspond to prices with the following two types of constraints: (i) first three moments for both the underlying and average stock prices and their first order cross moments (B3N3NB1), and (ii) first three moments for the average prices (B3). The two sets of prices are close to each other and the low-biased estimators from the mesh are approximately within 3% of the price from the Richardson extrapolated bushy tree price.

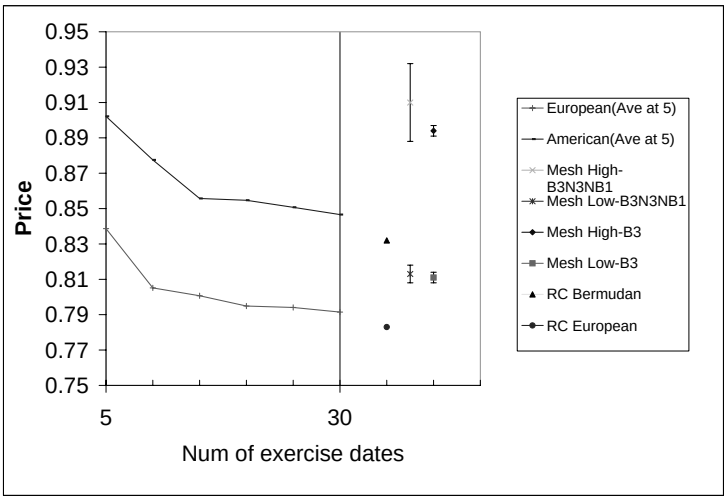

Figure 7: American Asian put with the payoff function $\max(X - \bar{S}, \cdot)$.

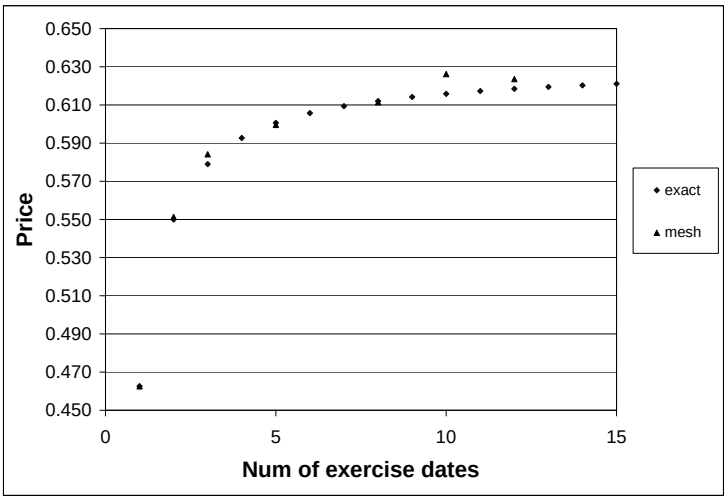

Figure 8: Four-dimensional American vs number of exercise dates

Table 1: Full-rank, multi-dimensional test cases using least squares weights. Here, r and T are the risk-free interest rate and the maturity, respectively. High estimates use replications of a 500-path mesh; Low estimates use 2000 independent paths through the mesh. High and Low interval estimates are ± 1 standard error. The covariance matrices for two cases are Σ_{2D} =((0.04,0.01),(0.01,0.04)) and Σ_{4D} = ((0.04,0.01,0.005,0.001),(0.01,0.02,0.01,0.005), (0.005,0.01,0.1,0.05),(0.001,0.005,0.05,0.08)). ;

Spot	Strike	r	T	High	Low	Exact	European
2D-2factor							
$(40, 40)$	40	0.1	0.5	1.176±0.007	1.126±0.009	1.137	0.982
$(38, 42)$	43	0.12	1.0	3.050±0.000	3.050±0.000	3.050	1.810
$(37, 45)$	40	0.15	1.0	0.809±0.010	0.741±0.007	0.762	0.514
4D-4factor							
$(40, 40, 40, 40)$	40	0.1	0.5	1.225±0.007	1.183±0.009	1.191	0.857
$(40, 38, 35, 45)$	42	0.12	1.0	2.669±0.004	2.603±0.001	2.665	1.496

5.3 Multi-Dimensional Options on a Geometric Average

If the payoff function depends on a vector of lognormally distributed asset prices only through their geometric average, then the option can be reduced to a one-dimensional problem, because the geometric average is again lognormal. Thus, this option provides nice test cases for our methods: we can solve it in the mesh as a multi-dimensional problem and compare with results obtained from a binomial lattice applied to the equivalent one-dimensional problem. If the multi-dimensional asset process is given by (2), then its geometric average process is given by the one-dimensional process

$$\bar{S}_t = \bar{S}_0 \exp\left((\tilde{r} - \frac{1}{2}\tilde{\sigma}^2)t + \sqrt{t}\tilde{\sigma}X\right), \tag{16}$$

where $\bar{S}_0 = (\prod_j S_0^j)^{1/n}$ and

$$\tilde{\sigma} = \frac{1}{n}\sqrt{\sum_{j,k}\Sigma_{jk}}, \tag{17}$$

$$\tilde{r} = r + \frac{1}{2n^2}\sum_{j,k}\Sigma_{j,k} - \frac{1}{2n}\sum_j \Sigma_{jj}. \tag{18}$$

In Figure 8 we plot results for an American put on the geometric average of four assets. The covariance matrix in this case has full rank and the weights are constructed from the transition probability density. Notice the rapid convergence to the continuous-exercise price as the number of exercise dates increases.

Table 1 lists American put prices for two-dimensional (2D) and four-dimensional (4D) cases having full rank. The mesh estimates (with weights constrained to match

Table 2: Multi-dimensional, singular test cases using least squares weights. All cases have a strike of 40, a risk-free rate of 10%, an expiration of 0.5 years. Under the Mesh column, the first number gives the number of paths in the mesh and the second number gives the number of independent paths simulated through the mesh to generate the Low estimate. High and Low interval estimates are ± 1 standard error.

Spot	Mesh	High	Low	Exact	European
2D-1factor					
(40, 40)	100/1000	1.215±0.103	0.976±0.016	1.027	0.863
	200/2000	1.065±0.128	0.982±0.013		
	200/10000	1.220±0.208	0.984±0.007		
	500/2000	1.241±0.130	1.004±0.012		
	2000/20000	1.103±0.083	1.010±0.004		
	3000/20000	1.023±0.072	1.013±0.002		
4D-2factor					
(40,40,40,40)	1000/10000	1.665±0.006	1.510±0.005	1.521	1.359
	2000/10000	1.663±0.003	1.511±0.004		
4D-1factor					
(40,40,40,40)	1000/10000	1.387±0.004	1.267±0.005	1.279	1.115
	2000/10000	1.399±0.002	1.268±0.007		
8D-3factor					
(...40...)	2000/10000	3.961±0.010	3.488±0.012	3.529	3.417
12D-3factor					
(...40...)	2000/10000	3.059±0.008	2.710±0.008	2.749	2.637
16D-3factor					
(...40...)	2000/10000	2.761±0.007	2.436±0.008	2.473	2.352
20D-3factor					
(...40...)	2000/10000	2.578±0.008	2.269±0.007	2.305	2.176

all means and covariances) are compared both with exact values (computed in a binomial lattice) and with the corresponding European prices. Including the European price helps illustrate how much of the early-exercise value is found by the mesh. Clearly, the mesh finds most of this value in these examples. As might be expected, the estimates for the longer maturity put prices are less accurate than those for the shorter maturities.

Table 2 shows results for singular cases ranging from a two-dimensional, one-factor model to a 20-dimensional three-factor model. (Details of the covariance matrices used in these examples are available from the authors.) Several of these cases are extremely singular so that constructing weights based on (nonexistent) transition densities would seem to hopeless. These are clearly harder problems, but notice that in most cases the Low estimate is accurate to the first digit at which the Exact and European prices differ.

6 Conclusion

This paper expands the scope of the stochastic mesh method for pricing multidimensional American options to address models in which a transition density for the underlying state variables is unknown or fails to exist. This includes multivariate lognormal processes with a singular covariance matrix. We avoid the need for a transition density by choosing mesh weights through an optimization problem. We choose weights using either a maximum entropy or least squares criterion subject to constraints that ensure the mesh correctly prices simple instruments. Numerical examples illustrate the method. Important directions for future work include improvements in speed and methods for pruning negative weights.

Acknowledgments Research support from NSF grant DMI9457189, from the Center for Applied Probability at Columbia University, from an IBM University Partnership Award, and from Goldman Sachs & Co. is gratefully acknowledged.

References

[1] Andersen, L., 1999, "A Simple Approach to the Pricing of Bermudan Swaptions in the Multi-Factor Libor Market Model," working paper, General Re Financial Products, NY.

[2] Avellaneda, M., C. Friedman, R. Holmes, D. Samperi, 1997, "Calibrating Volatility Surfaces via Relative-Entropy Minimization," *Applied Mathematical Finance*, Vol.4, No.1, 37–64.

[3] Barraquand, J., and D. Martineau, 1995, "Numerical Valuation of High Dimensional Multivariate American Securities," *Journal of Financial and Quantitative Analysis*, Vol. 30, No. 3, 383–405.

[4] Broadie, M., and P. Glasserman, 1997, "Pricing American-Style Securities Using Simulation," *Journal of Economic Dynamics and Control*, Vol. 21, Nos. 8-9, 1323–1352.

[5] Broadie, M., and P. Glasserman, 1997, "A Stochastic Mesh Method for Pricing High-Dimensional American Options," working paper, Columbia University.

[6] Broadie, M., P. Glasserman, and G. Jain, 1997, "Enhanced Monte Carlo Estimates for American Option Prices," *Journal of Derivatives*, Vol. 5, No. 1 (Fall), 25–44.

[7] Carriere, J.F., 1996, "Valuation of the Early-Exercise Price for Derivative Securities using Simulations and Splines," *Insurance: Mathematics and Economics*, Vol. 19, 19–30.

[8] Golan, A., G. Judge and D. Miller, 1996, *Maximum Entropy Econometrics: Robust Estimation with Limited Data*, Wiley, New York.

[9] Longstaff, F., and E. Schwartz, 1998, "Valuing American Options By Simulation: A Simple Least Squares Approach," working paper, UCLA Anderson Graduate School of Management.

[10] Pedersen, M., 1999, "Bermudan Swaptions in the LIBOR Market Model," working paper, SimCorp A/S, Copenhagen, Denmark.

[11] Tilley, J.A., 1993, "Valuing American Options in a Path Simulation Model," *Transactions of the Society of Actuaries*, Vol. 45, 83–104.

[12] Tsitsiklis, J., and B. Van Roy, 1997, "Optimal Stopping of Markov Processes: Hilbert Space Theory, Approximation Algorithms, and an Application to Pricing High-Dimensional Financial Derivatives," working paper, Laboratory for Information and Decision Sciences, MIT, Cambridge, MA.

[13] Wilmott, P., *Derivatives*, Wiley, Chichester, England, 1998.