A Research on Pricing High-Dimensional American Options Based on Broadie-Glassermann Stochastic Mesh Algorithm

Lin Zeng(lz447), Sizhang Zhao(sz459)

Abstract

Increasingly complex and sophisticated financial products continue to be introduced and accepted in the marketplace. In particular, many of them have payoffs that depend on multiple assets, designed to isolate ever refined types of risks. American options where the owner has the right to exercise early, are especially challenging to price. We study a stochastic mesh method introduced by Mark Broadie and Paul Glasserman for pricing high-dimensional American options when there is a finite, but possibly large, number of exercise dates. Then we apply antithetic variates for variance reduction. The criticism of the model is given based on the computational evidence.

1 The Stochastic Mesh Method

The stochastic mesh method is designed to solve a general optimal stopping problem, of which the American option pricing problem with discrete exercise opportunities (Bermudan) is the focus of our study. Let $S_t = (S_t^1, ..., S_t^n)$ be the underlysing asset price which is a Markov process on R^n, with fixed initial state S_0 and discrete time parameter $t = 0, 1, ..., T$. The problem is to compute

$$Q = \underset{\tau}{max} E[h(\tau, S_\tau)] \tag{1}$$

where τ is a stopping time taking values in the finite set $\{0, 1, ..., T\}$, and $h(t, x) \geq 0$ is interpreted as a payoff from exercise at time t in state x.

More generally, the value starting at time t in state x is

$$Q(t, x) = max(h(t, x), E[Q(t + 1, S_{t+1})|S_t = x]) \tag{2}$$

for $t < T$ and $Q(T, x) = h(T, x)$. We are interested in computing $Q \equiv Q(0, S_0)$.

1.1 Mesh Estimator

The stochastic mesh method begins by generating random vectors $X_t(i)$ for $i = 1, ..., b$ and $T = 1, ..., T$. Since S_0 is given, we set $X_0(1) = S_0$. The mesh estimator is defined inductively by setting

$$\hat{Q}(T, X_T(i)) = h(T, X_T(i)) \tag{3}$$

for $i = 1, ..., b$. For times $t = T - 1, ..., 0$ and $i = 1, ..., b$, the mesh estimator is

$$\hat{Q}(t, X_t(i)) = max \left(h(t, X_t(i)), \frac{1}{b} \sum_{j=1}^{b} \hat{Q}(t + 1, X_{t+1}(j)) w(t, X_t(i), X_{t+1}(j)) \right) \tag{4}$$

where $w(t, X_t(i), X_{t+1}(j))$ is a weight attached to the arc joining $X_t(i)$ to $X_{t+1}(j)$. We use the notation $\hat{Q}(t, X_t(i))$ to indicate the algorithm's estimate of the true American price $Q(t, X_t(i))$. At time $t = 0$ only $i = 1$ is applicable in equation (4) and $\hat{Q} \equiv \hat{Q}(0, S_0)$ is the final mesh estimator of the true price Q.

In order to complete the description of the algorithm, we need to address the details of how the random vectors $X_t(i)$ are generated and how the weights on the arcs are determined.

Let $f(t, x, \cdot)$ denote the density of S_{t+1} given $S_t = x$ and let $f(t, \cdot)$ denote the marginal density of S_t (with S_0 fixed). For $t = 1, ..., T$, the vectors $X_t(i), i = 1, ..., b$ are generated as i.i.d. samples from the density function $g(t, \cdot)$. We require $g(t, u) > 0$ if $f(t - 1, x, u) > 0$ for some x. The choices of the mesh density functions $g(t, \cdot)$ are crucial to the practical success of the method, so we will discuss it in further details in the Section 1.3.

In order to motivate the weights on the arcs, recall that the American option value at time t in state $S_t = x$ is

$$Q(t, x) = max\,(h(t, x), E[Q(t + 1, S_{t+1})|S_t = x])$$

We need to approximate $Q(t, x)$ at all points $x = X_t(1), ..., X_t(b)$ using the available information from the mesh, i.e., $\hat{Q}(t+1, X_{t+1}(j))$ for $j = 1, ..., b$. To do this, we need to estimate all of the quantities $E[Q(t+1, S_{t+1})|S_t = X_t(i)], i = 1, ..., b$, using the same information $\hat{Q}(t+1, X_{t+1}(j)), j = 1, ..., b$. The main difficulty is that density of S_{t+1} given $S_t = x$ is $f(t, x, \cdot)$ while the mesh points $X_{t+1}(j), j = 1, ..., b$ were generated from the density function $g(t+1, \cdot)$. However, observe that

$$\begin{aligned} E[Q(t+1, S_{t+1})|S_t = x] &\equiv \int Q(t+1, u) f(t, x, u) du \\ &\equiv \int Q(t+1, u) \frac{f(t, x, u)}{g(t+1, u)} g(t+1, u) du \\ &\equiv E\left[Q(t+1, X_{t+1}(j)) \frac{f(t, x, X_{t+1}(j))}{g(t+1, X_{t+1}(j))}\right] \end{aligned} \tag{5}$$

The final expression allso us to approximate the expectations $E[Q(t+1, S_{t+1})|S_t = X_t(i)], i = 1, ..., b$ even though the points $X_{t+1}(j), j = 1, ..., b$ were generated according to the density $g(t+1, \cdot)$ and not according to $f(x, X_t(i), \cdot)$. [1] Define the mesh estimator as

$$\hat{Q}(t, x) = max\left(h(t, x), \frac{1}{b}\sum_{j=1}^{b} \hat{Q}(t+1, X_{t+1}(j)) w(t, x, X_{t+1}(j))\right) \tag{6}$$

where $w(t, x, X_{t+1}(j)) = f(t, x, X_{t+1}(j))/g(t+1, X_{t+1}(j))$. The mesh estimator approximates $Q(t, X_t(i))$ by $\hat{Q}(t, X_t(i))$.

There are two main properties of the mesh estimator under some smooth assumption of $f(t, x, \cdot)$ and $g(t+1, \cdot)$.

Theorem 1 (Mesh estimator bias): *The mesh estimator $\hat{Q}_b(0, S_0)$ is biased high, i.e.,*

$$E[\hat{Q}_b(0, S_0)] \geq Q(0, S_0)$$

for all b.

Theorem 2 (Mesh estimator convergence): *Let $r > p > 1$. Assume $f(t, x, \cdot)$ and $g(t+1, \cdot)$ are well-defined,*

$$||\hat{Q}_b(t, x) - Q(t, x)|| \to 0$$

as $b \to \infty$, for all x and t.

1.2 Path Estimator

Next we develop an estimator based on simulated paths which is biased low. By combining the high-biased mesh estimator with a low-biased path estimator, we can generate a valid confidence interval for the American option price.

The path estimator is defined by simulating a trajectory of the underlying process S_t until the exercise region determined by the mesh is reached. Denote the simulated path by $S = (S_0, S_1, ..., S_T)$. The path S is simulated (independent of the mesh points $X_t(i)$) according to the density function of the process S_t , i.e., the density of the simulated point

[1] Very beautiful result, isn't it?

S_{t+1} given $S_t = x$ is $f(t, x, \cdot)$. Along this path, the optimal policy exercises at $\tau^*(S) = min\{t : h(t, S_t) \geq Q(t, S_t)\}$ for a payoff of $h(\tau^*, S_{\tau^*})$. The approximate optimal policy determined by the mesh exercises at

$$\hat{\tau}(S) = min\left\{t : h(t, S_t) \geq \hat{Q}(t, S_t)\right\} \tag{7}$$

where $\hat{Q}(t, S_t)$ is given in equation (6). Define the path estimator by

$$\hat{q} = h(\hat{\tau}, S_{\hat{\tau}}) \tag{8}$$

Since the stopping time $\hat{\tau}_b$ is not necessarily an optimal stopping time due to the discrete exercise time, an immediate consequence is that the path estimator is a lower bound on the true price. There are two similar properties of the path estimator.

Theroem 3 (Path estimator bias): *The path estimator $\hat{q}_b$ is biased low, i.e.,*

$$E[\hat{q}_b] \leq Q(0, S_0)$$

for all b.

Theorem 4 (Path estimator convergence): *Suppose the conditions in Theorem 2 are in effect and that $E[h(t, S_t)^{1+\epsilon}] < \infty$ for all $t = 1, ..., T$, for some $\epsilon > 0$. Suppose also that $P(h(t, S_t) = Q(t, S_t)) = 0$ for all $t = 0, 1, ..., T-1$. Then*

$$E[\hat{q}_b] \rightarrow Q(0, S_0)$$

as $b \rightarrow \infty$, i.e., $\hat{q}_b$ is asymptotically unbiased.

1.3 Interval estimation

In order to give a confidence interval for the option price Q, generate N independent meshes with corresponding mesh estimates $\hat{Q}^{(i)} = \hat{Q}_b^{(i)}(0, S_0), i = 1, ..., N$, and then combine them to give

$$\bar{Q}(N) = \frac{1}{N}\sum_{i=1}^{N} \hat{Q}^{(i)}$$

For each mesh $i, i = 1, ..., N$, generate n_p independent paths and corresponding path estimates. Average these individual estimates to give the path estimates $\hat{q}^{(i)} = \hat{q}_b^{(i)}(0, S_0), i = 1, ..., N$. These N path estimates, each based on n_p paths, are combined to give

$$\bar{q}(N) = \frac{1}{N}\sum_{i=1}^{N} \hat{q}^{(i)}$$

With $\bar{Q}(N)$ and $\bar{q}(N)$ replacing $\hat{Q}_b$ and $\hat{q}_b$, respectively. Theorems 1-4 hold for any $N \geq 1$. Finally, form the confidence interval

$$\left[\bar{q}(N) - z_{\alpha/2}\frac{s(\hat{q})}{\sqrt{N}}, \bar{Q}(N) + z_{\alpha/2}\frac{s(\hat{q})}{\sqrt{N}}\right] \tag{9}$$

In fact, this interval (9) is conservative.

1.4 Selection of the Mesh Density

As described in the paper, for the stochastic mesh method to be practically viable, it is essential to exploit efficiencies in the computation of the estimator wherever possible. This requires, in particular, careful choice of the density used to generate the mesh. It also motivates the use of control variates which is discussed in the next section.

The paper shows that if the mesh density is not chosen carefully there is a risk an exponential growth in variance. The *average density method* defined below is significant because it eliminates this risk. Indeed, it reduces the potentially exponential variance of the $L(T, j)$ to zero!

The average density functions are defined as

$$g(t, u) = f(0, S_0, u) \qquad \text{for} \quad t = 1 \tag{10}$$

and

$$g(t, u) = \frac{1}{b} \sum_{j=1}^{b} f(t-1, X_{t-1}(j), u) \qquad \text{for} \quad t = 2, ..., T \tag{11}$$

Using the average density is equivalent to generating b independent paths of the underlying and then "forgetting" which nodes were on which paths. Taking this observation one step further leads to the following implementation: simulate b independent paths $(X_0(i), ..., X_T(i)), i = 1, ..., b$, as in an ordinary simulation and then apply the weight

$$\frac{f(t-1, X_{t-1}(i), X_t(j))}{b^{-1} \sum_{k=1}^{b} f(t-1, X_{t-1}(k), X_t(j))}$$

to the transition from $X_{t-1}(i)$ on the ith path to $X_t(j)$ on the jth path. These weights define the mesh; recall equation (6). Since this construction generates exactly one successor from each of the b transition densities $f(t-1, X_{t-1}(i), \cdot), i = 1, ..., b$, it may be viewed as a *stratified* implementation of the average mesh density. This is the construction we use in our numerical experiments.

1.5 Variance Reduction: Antithetic Variates

The use of antithetic variates with the path estimator is fairly standard. For each simulated path $S = (S_0, S_1, ..., S_T)$, we also generate an antithetic path $S' = (S'_0, S'_1, ..., S'_T)$. For example, if the original path is driven by standard normal increments, then the antithetic path is driven by the negative of the normal increments. The two option estimates, which in general involve different stopping times, are then averaged to give the path estimate. When controls are used, they are computed in the same way for the antithetic paths. More detailed discussion of the antithetic technique is given in Boyle, Broadie, and Glasserman (1997).

2 Computational Results

2.1 Max Option Pricing results

In this section, we give numerical results with the stochastic mesh method based on max-option. The payoff upon exercise of this option is

$$h(t, S_t) = (max(a_1 S_t^1, ..., a_n S_t^n) - K)^+$$

If the S_t^i are prices of discount bonds of various maturities (in, e.g., a Gaussian model of interest rates), then the payoff given above for a basket option becomes the payoff of an option on a coupon-paying bond.

Table 2.1.1 show five asset max-option results with, respectively, $T = 3, 6,$ and 9 (and thus $4, 7$ and 10 exercise dates including time zero). The parameteres are $r = 5\%$, $\delta = 10\%$, $\sigma = 20\%$, $\rho = 0$, $K = 100$. The initial vector is $S_0 = (S, ..., S)$ with $S = 90, 100$ or 110 as indicated in the table. Equal time steps are used, with exercise opportunities at $t = 0, 1, ..., T$ years. For each panel, the parameters (b, n_p) are $(50, 500)$ and $(100, 1000)$ for each two rows, respectively.

Since the pricing results provided in the paper are obtained with the greatest computational effort which are not described in details, we compare our pricing with Premia (Premia is a software designed for option pricing, hedging and financial model calibration). We use the antithetic variates to reduce the variance of the path estimate, so we find our confidence intervals are narrower than that provided by Premia. However, this pricing results are far from practical viability and accuracy.

S	Mesh Replication	T=3							
		Path Est	Path std err	Mesh Est	Mesh std err	90% confidence interval		Premia L	Premia U
90	50, 500	14.07	8.58	19.46	2.33	13.914849	19.730392	13.77	23.22
	100, 1000	14.31	8.32	18.93	1.67	14.200288	19.121743	14.43	21.79
100	50, 500	22.21	9.73	31.22	2.90	21.3567773	31.4743715	23	36.49
	100, 1000	23.24	10.28	30.57	2.07	22.6030869	30.6982999	23.34	33.79
110	50, 500	32.15	9.46	44.27	3.40	31.85426	45.095609	33.12	50.99
	100, 1000	33.32	11.34	42.91	2.36	32.6171395	43.0562743	33.49	47.19
S	Mesh Replication	T=6							
		Path Est	Path std err	Mesh Est	Mesh std err	90% confidence interval		Premia L	Premia U
90	50, 500	16.26	8.56	29.63	3.69	16.112435	29.902387	14.78	40.33
	100, 1000	16.60	8.95	28.91	2.60	16.045149	29.076052	14.53	40.81
100	50, 500	24.37	10.06	43.24	4.41	23.4882021	43.6265535	24.36	58.31
	100, 1000	25.16	10.93	42.12	3.06	24.4825516	42.3096608	24.22	57.18
110	50, 500	34.46	11.81	57.22	4.62	34.056135	58.256261	34.66	78.11
	100, 1000	34.37	12.96	56.40	3.41	33.5667309	56.611354	33.15	74.42
S	Mesh Replication	T=9							
		Path Est	Path std err	Mesh Est	Mesh std err	90% confidence interval		Premia L	Premia U
90	50, 500	16.21	8.38	34.38	4.24	15.476575	34.749027	14.8	56.92
	100, 1000	16.61	9.28	34.35	3.06	16.032395	34.53558	15.38	52.13
100	50, 500	25.32	10.36	48.92	5.28	24.4115553	49.3823741	23.47	78.78
	100, 1000	26.32	11.72	48.04	3.60	25.5935869	48.2631303	23.25	71.4
110	50, 500	34.51	11.89	64.41	5.68	34.006474	65.454209	34.2	102.64
	100, 1000	36.29	13.64	63.11	4.11	36.037325	63.960866	32.43	92.01

Table 2.1.1 Max Option Pricing results

In addition, we demonstrate the necessarity for using antithetic variance reduction method for path estimator. Table 2.1.2 shows that using the antithetic method, the variance of path estimator can be reduced more than twice.

K	No Antithetic Variate	Antithetic Variate
90	153.35	70.08
100	224.47	76.55
110	302.88	117.19

Table 2.1.2 Comparison for the Variance for Applying Antithetic Variate Method

2.2 Numerical Difficulties

The computational effort in generating the mesh is proportional to $n \times b \times T$. The effort in the recursive pricing of equation (6) is proportional to $n \times b^2 \times T$. Hence the overall effort for mesh estimator is polynomial in the problem dimension (n), the mesh parameter (b) and the number of exercise opportunities ($T + 1$).

Equation (9) shows that the mesh estimator must be computed before the path estimator. Once the mesh estimator has been computed, the additional effort to generate the path estimator is $n \times b \times T$. In our numerical implementation, we average the results from n_p independent paths to give the final path estimator for each mesh. For the path and mesh estimators to have comparable variances, we take n_p proportional to b. Hence, the overall work associated with the path estimator is proportional to $n \times b^2 \times T$, the same as the mesh estimator.

In Table 2.1.1, the mesh parameter b and path parameter n_p doubles from one row to the next within each panel. Hence the computational effort increases by roughly a factor of four from one row to the next. We use R to run our algorithm. The CPU time for the first row in each panel is about 4.3 minutes and for the successive row is 23.7 minutes. Although we can see that with larger b and n_p, the lower and upper bound would converge as shown in the paper. It takes a long time for us to replicate the results. And surely, by using the package for compute mesh density in R, it slows down the algorithm in a sense.

3 Criticism of the Model

In this study, we analyze and test the stochastic mesh method for pricing a general class of high-dimensional pricing problems with a finite number of exercise dates. The computational effort increases quadratically with the number of mesh points, linearly with the number of exercise opportunities, and linearly with the problem dimension. The paper show that the method converges as the computational effort increases. Numerical results illustrate this convergence and demonstrate this viability of the method. That being said, there are some evident limitation of this method.

1. Expensive computation

   Due to the reason that we can only approximate a continous process via Monte Carlo Simution, it is understandable that we can never price an American option through Monte Carlo simulation. According to Theorem 2 and Theorem 4, if number of states $b \to \infty$, the high-biased estimator and the low-biased estimator would converge to the true continuation value and finally leads to the true value of the option. Thus, the number of states b plays an important role to price American Option. However, as we mentioned in Part 2.2, the time complexity of our algorithms is $n \times b^2 \times T$ for both mesh estimator and path estimator. Thus, if we increase the number of the states, our algorithm would take intolerably long time. Thus, there exists a tradeoff between the accuracy and speed for our algorithms, which is one of the most important constraints for the algorithms. Besides the number of the states, we cannot decrease the time steps also due to the computational constraints, which also made our model far from reality.

2. Wide range for the price

   From our numerical results, we can easily see the difference between the high-biased estimator and the low-estimator is so large that the price is not meaningful. However, due to the high computational costs, we cannot enlarge the number of states that much. Thus the low accuracy make our algorithms less practical. As mentioned in the paper, practical success of this method depends critically on the use of effective variance reduction techniques.

3. Strong assumptions

   We have made some assumptions to make sure our algorithms work.

   (a) We assume the Black-Scholes model is correct, which means that we assume the return is log-normally distributed. And we assume the constant volatility, constant interest rate and constant dividend rate. Based on the assumptions, we construct the weight matrix with the conditional density functions. If the assumptions of black-scholes model is violated, our model would also collapse.

   (b) We made a lots of technical assumptions(due to page constraint, we have not mentioned here) to make sure our estimator would converge to the true value as we increase the number of states. If any of the assumptions is violated, the two estimator would not converge, which would make our model useless.

4. The reliance on explicit knowledge of the transition density of the underlying state variables

In many cases, the transition density is unknown or may even fail to exist. In such setting, one may consider using a normal or lognormal density as an approximation. An alternative strategy for selecting mesh weights is proposed and tested in Broadie, Glasserman, and Ha (2000). That method does not use a transition density but instead uses information about moments of the underlying state variables or the prices of easily computed European options.

5. Truly high dimention?

   Although we have five underlying assets for the model, the assumptions for them are really strong. They can only have one common volatility, start with the same price and the correlation between the assets are all the same. And it can hardly be the reality. However, this problem can be easily solved. We implement one improved version of the model with a more general distribution function for high dimentianal log-normal distribution. However, this algorithm would take much longer time and thus we did not research deeply into it.