Lattice Option Pricing By Multidimensional Interpolation

Vladislav Kargin*

October 27, 2004

Abstract

This note proposes a method for pricing high-dimensional American options based on modern methods of multidimensional interpolation. The method allows using sparse grids and thus mitigates the curse of dimensionality. A framework of the pricing algorithm and the corresponding interpolation methods are discussed, and a theorem is demonstrated that suggests that the pricing method is less vulnerable to the curse of dimensionality. The method is illustrated by an application to rainbow options and compared to Least Squares Monte Carlo and other benchmarks.

The fundamental problem of options theory is the valuation of hybrid, non-linear securities, and options theory is an ingenious but glorified method of interpolation.

Emanuel Derman "A guide for the perplexed quant"

1 Introduction

Lattice option pricing[1] is very popular among practitioners because in one-dimensional situations it is straightforward to implement and has a transparent interpretation. In multiple dimensions it has two serious drawbacks: the need to build recombining trees and the curse of dimensionality. If branches of the tree do not recombine then the number of nodes grows exponentially with the number of time steps. Similarly, the number of points in a regular grid grows exponentially with the space dimension. For example, to approximate every point in the nine-dimensional hypercube with 10% precision we need one billion points. With twenty factors, the argument runs, experiments with IBM's fastest supercomputer will quickly convince us that the lattice method of pricing is impractical with many dimensions.[2]

*Cornerstone Research, 599 Lexington Avenue floor 43, New York, NY 10022; skarguine@cornerstone.com

[1] Invented by Cox, Ross, and Rubinstein (1979).

[2] The speed of this supercomputer is around 10^{13} operations per second, so an optimistic estimate of the execution time is 10^7 seconds or about 100 days. For pricing algorithms that

This argument is usually well taken but essentially wrong. First, trees may very well have non-recombining branches and a moderate number of nodes. We only need to leave some nodes without descendants and to interpolate values on these nodes. Second, in the case of many dimensions, we can use irregular grids with a moderate number of points provided that we can interpolate the value function faithfully from its values on the irregular grid.

A suspicion may arise that this just shifts the computational burden to the interpolation problem and that the curse of dimensionality will remain as dangerous as it was before. Indeed, traditional approximation theory says that accurate approximation of a general function needs a number of grid points that is exponential in the space dimension.

Does this invalidate the idea of pricing by interpolation? No, because we deal with specific classes of functions which may be approximated better than an arbitrarily chosen function. As the simplest example, consider linear functions. They can be recovered from values on only $d+1$ points in d-dimensional space. Recall also the classical Kotelnikov-Shannon sampling theorem that says that a function with limited bandwidth of its Fourier transform can be completely recovered from the values it takes on a discrete sample of points. These examples suggest that for certain classes of functions the problem of approximation may be satisfactorily solved even in multiple dimensions. Indeed, this hope is being realized in recent and continuing work on non-linear multidimensional approximation. This paper aims to apply these new ideas to option pricing.

The essential idea is to choose the approximating functions adaptively. First, the researcher chooses a large class of functions that are easy to compute and that are suitable to the problem at hand. Second, the researcher allows the data to select the functions that are most fitting as a basis of approximation. As a result, the approximation is well adapted to the properties of the given function. The main practical and theoretical tasks are automating this procedure and exploring its convergence properties.

The idea of using approximations for option pricing is not entirely new. Recently, it was implemented by Longstaff and Schwartz (2001) in their "simple but powerful" Least Squares Monte Carlo algorithm.[3] Their method is based on the Monte Carlo pricing method and they assume that the researcher can guess good basis functions for approximations. Tsitsiklis and Roy (2001) describe a similar method and analyze its convergence properties. The cardinal distinction of the method described in this paper is that it suggests the adaptive choice of the approximating basis. This adaptive way of approximation allows construction of an algorithm which is universally applicable to a wide range of possible options.

As an additional benefit, the approximation provided by the new algorithm

generalize lattice pricing to multiple dimensions see Boyle (1988), Boyle, Evnine, and Gibbs (1989), Madan, Milne, and Shefrin (1989), Kamrad and Ritchken (1991), and McCarthy and Webber (2001).

[3]See also related work by Tilley (1993), Barraquand and Martineau (1995), Carriere (1997), Broadie and Glasserman (1997a), Broadie and Glasserman (1997b), and Raymar and Zwecher (1997).

allows easy pricing everywhere in factor space. In contrast, both the standard lattice and Monte Carlo methods produce option values for only one combination of factors. This benefit is especially useful if there is a need to visualize the dependence of option price on factors, or to compute hedge factors - sensitivities of the option value with respect to changes in factors.

The remainder of the paper is organized as follows. Section 2 briefly reviews the problem of option pricing and relates it to the problem of approximation. Section 3 outlines the framework of the algorithm, illustrates it with a simple example, and briefly describes why adaptive approximation is likely to break the curse of dimensionality. Section 4 explaines how the approximation can be used to find lower and upper bounds on the option value. Section 5 applies the method to a set of benchmark options and compare the results with the results in the literature. Section 6 concludes.

2 Option Pricing and Interpolation

The problem is as follows. Let the price of a derivative security depend on N factors that follow a specified diffusion process. The derivative is of the American type and so can be exercised at any time. Assume also that the price of the derivative is not path-dependent. Then by the dynamic replication argument, the value of the derivative satisfies the familiar[4] partial differential equation :

$$\frac{\partial f}{\partial t} + r\sum_i x_i \frac{\partial f}{\partial x_i} + \frac{1}{2}\sum_{i,k} v_{ik} x_i x_k \frac{\partial^2 f}{\partial x_i \partial x_k} = rf. \tag{1}$$

Here $f(x,t)$ is the value of the derivative at time t if the vector of factors is x.

One way to solve this equation is to write it in an integral form using a certain measure μ over the space of Brownian motion paths. This is the Feynman-Kac representation[5] of PDE (1) solution as an integral from a functional of Brownian motion paths:

$$f(x,0) = \sup_\tau \int_{x(t)\in\mathcal{W}} \pi(x(\tau),\tau) e^{-rt} d\mu(x(t)). \tag{2}$$

Here τ is a stopping time, $\pi(x(\tau),\tau)$ is the payoff at time τ if the factors are $x(\tau)$, and $\mathcal{W}$ is the space of paths of the Wiener process.

We can write the Wiener measure μ as a limit over a sequence of time discretizations with Gaussian transition probabilities. Then, each of the discrete time problems can be solved recursively through the Bellman equation that relates the current option value to the values at the next time stage:

$$f(x,t) = \max\left\{\pi(x(t),t); \int_{y\in X} f(y,t+\Delta t) e^{-r\Delta t} d\widetilde{\mu}(x,y,t)\right\}. \tag{3}$$

[4] see, for example, Hull (1999) or Wilmott and Howson (1995).

[5] see Kac (1949) or Karatzas and Shreve (1991).

Here Δt is a time interval, $d\widetilde{\mu}(x,y,t)$ is the probability of transition from x to y consistent with the Wiener measure $d\mu$, and X is the space of factors.

The next step - crucial for our analysis - is to discretize the equation over space. This means choosing a grid $G \subset X$ and a suitable approximation for $d\widetilde{\mu}$. Here is where difficulties begin. The standard tree methods use a regular – usually cubic – lattice, and specify probabilities of transitions from each lattice point to nearby lattice points so as to match the covariance matrix of the continuous process.

In multidimensional situations the number of points in regular lattices is prohibitively large. Therefore we have to use an irregular grid with large gaps. Transitions over such a grid are unlikely to approach the Wiener process uniformly.[6] So what to do? One solution is to use spatial interpolation as in the following formula:

$$\widehat{f}(x,t) = \max\left\{\pi(x(t),t), \int_{y\in\mathcal{D}(x)} \widehat{f}(y,t+\Delta t)e^{-r\Delta t}d\widehat{\mu}(x,y,t)\right\}. \qquad (4)$$

Here x belongs to grid G and y to $\mathcal{D}(x)$, which is a set of descendants of point x, which may very well lie outside of the grid. The function $\widehat{f}(y,t+\Delta t)$ is interpolated from the values of $\widehat{f}$ on the grid points that were obtained in the previous step of recursion. Measure $d\widehat{\mu}$ approximates $d\mu$.

In summary, the main idea of the method is to separate two different uses of the grid, which usually plays a crucial role in both approximating the evolution of the stochastic process and in keeping information about the option value function. We suggest using the grid only for the latter purpose and simulating the stochastic process by computing small clusters of descendant points around each grid point. The values on the descendant points are interpolated from values on the grid points. Clearly, the success of this idea crucially depends on the quality of the spatial interpolation. We will discuss modern methods of multidimensional interpolation immediately after presenting the outline of the algorithm and an example.

3 Outline of Algorithm

Here is the general outline of the algorithm:

1) Generate G, a grid – possibly irregular – in the factor space.

2) For each point $x \in G$, initialize the value function by computing the payoff at the final stage T.

3) Begin recursion over $t < T$: Compute an approximation to the value function at stage t using an approximation at stage $t+1$.

This step can be realized in different ways. Since one part of this step is applying the backward integral operator that corresponds to the factor process,

[6] See, however, Berridge and Schumacher (2004) for encouraging advances in this direction.

the algorithm becomes more precise if it can be computed analytically. If not, we can always proceed as follows:

a) For each point $x \in G$, compute a set of the states $\mathcal{D}(x)$ that follow x in a discrete approximation to the factor process.

b) For each $y \in \mathcal{D}(x)$, compute $\widehat{f}(y, t+dt)$ by interpolating.

c) Compute the continuation value function at point x as the discounted average of $\widehat{f}(y, t+dt)$ over $\mathcal{D}(x)$.

d) Compute the new value function by taking the maximum of the continuation value function and the exercise payoff.

e) Proceed to the next step of the recursion.

Example 1 ***Option on the value of a firm with debt***

The algorithm is best illustrated by a one-dimensional example. Consider an option on the value of a firm that pays fixed coupons on an outstanding debt. Although between payment dates the value of the firm follows a lognormal diffusion process, the simple binomial model is not going to have recombining branches at the payment dates. How do we apply the interpolation method in this situation?

Figure 1. One-dimensional Option Pricing by Tree with Interpolation

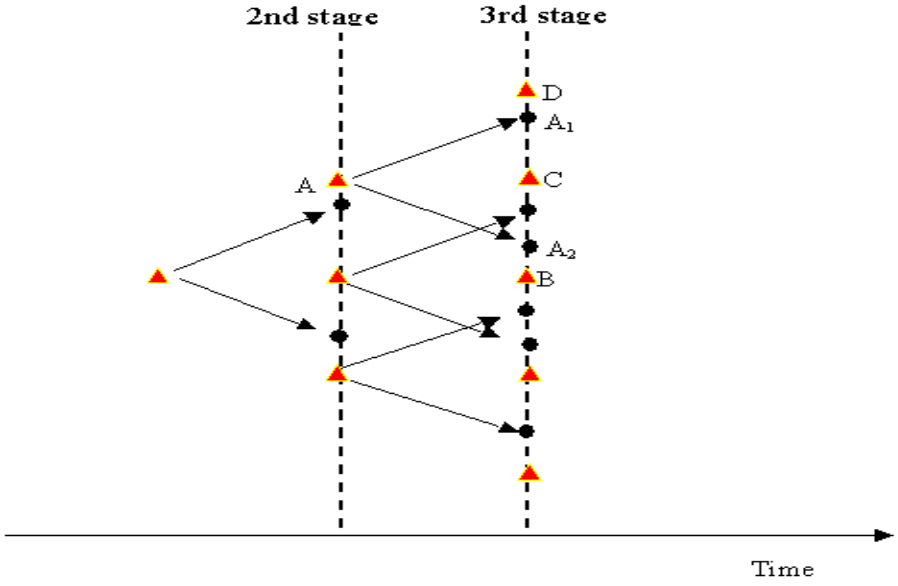

Figure 1 illustrates the organization of computations. Grid points are pictured by red triangles and their descendants by black balls. The value of the third-stage triangles is computed by definition. To compute the value of the second-stage triangles, we compute the value of their descendants, the third-stage black balls. We do it by a simple linear approximation from the two neighboring triangles. After the value of the descendants is computed, we compute the continuation value of the triangle as a discounted weighted average

using risk-neutral probabilities as weights. For example, see how we compute the value at triangle A. We find its descendants A_1 and A_2, and we compute the value of the descendants by a linear approximation from B, C, and D. Then we compute the value of A by taking a discounted average of the values of A_1 and A_2.

This example illustrates the usefulness of the interpolation method in a simple one-dimensional situation. The real strength of the method, however, is that it extends the simplicity of lattice methods to multidimensional situations. This extension uses modern methods of non-linear multidimensional approximation, which we discuss next.[7]

In multiple dimensions the most successful approach to date is by fixing an over-complete set of basis functions and then looking for an approximation recursively. The typical realization of this idea is by the relaxed greedy algorithm (RGA), which proceeds by forming a convex combination of a ridge function $\phi(b_{n+1}x + c_{n+1})$ and the previous approximation $f_n(x)$:

$$f_{n+1}(x) = \alpha\phi(b_{n+1}x + c_{n+1}) + (1 - \alpha)f_n(x), \tag{5}$$

and then estimating b_{n+1}, c_{n+1}, and α. Here $\phi(t)$ is a function of one-dimensional parameter, and $b_{n+1}x$ means the scalar product of vectors b_{n+1} and x.

Barron (1993) advocates using a sigmoidal function ϕ consistent with the neural network literature, Jones (1992) uses sinusoidal functions, and Breiman (1993) suggests using a connected pair of half-hyperplanes. The specifics of our application call for a different choice. Since the approximation serves only as a tool for solving a partial differential equation, the preferable choice seems to be Gaussian functions that can be easily propagated backward by the action of the backward integral operator associated with the factor process.

A question immediately arises: In which circumstances can recursive methods find efficient approximations by Gaussians? Below is a theorem that answers this question by generalizing a theorem from Jones (1992).

Let

$$f(x) = \sum_{i=1}^{\infty} a_i\phi(x; B_i, C_i) + u(x), \tag{6}$$

where $\phi(x; B, C)$ is a multi-dimensional Gaussian with parameters B (precision, i.e. inverse of covariance matrix) and C (shift of the center).

Assume that

$$\begin{array}{ll} 1) & \sum |a_i| = K < \infty, \\ 2) & \|\phi\| = 1, \\ 3) & \|u\| \le \varepsilon, \end{array} \tag{7}$$

where the norm is the L^2-norm.

Denote functions $\pm K\phi(x; B, C)$ that enter the expansion (6) as ϕ_i. Then f is their linear convex combination. Let us define the approximation at stage $n + 1$ as a convex combination $f_{n+1} = (1 - \lambda)f_n + \lambda\phi$ of the previous stage

[7] Two excellent reviews of the modern non-linear approximation methods are DeVore (1998) and Temlyakov (2003).

approximation and one of the functions ϕ_i. We will choose the approximation that solves the following problem

$$\inf_{\lambda\in[0,1],\phi\in\{\phi_i\}} \left\{ \|(1-\lambda)f_n + \lambda\phi - f\|^2 \right\}. \tag{8}$$

Theorem 2 *For each $\alpha \in (0,1)$ and $n \geq 1$, the approximant f_n is guaranteed to satisfy the following inequality:*

$$\varepsilon_n^2 \equiv \|f_n - f\|^2 \leq \max\left\{ \frac{[(K+1)/\alpha]^2}{n}, \frac{\varepsilon^2}{(1-\alpha)^2} \right\}. \tag{9}$$

Remark: By taking infinum over α we can obtain the following two corollaries:

Corollary 3 *If $\varepsilon = 0$, then $\varepsilon_n^2 \leq (K+1)^2/n$*

Corollary 4 *If $\varepsilon > 0$ and n is sufficiently large, then*

$$\varepsilon_n^2 \leq \varepsilon^2 + 2\frac{K+1}{\sqrt{n}}\varepsilon + \frac{(K+1)^2}{n}. \tag{10}$$

Proof of Theorem: Let f_n be the approximation obtained at step n of the recursive algorithm. Consider the next recursive step. First, we can write:

$$\begin{aligned} \|(1-\lambda)(f_n - f) + \lambda(\phi - f)\|^2 &= (1-\lambda)^2 \|f_n - f\|^2 \\ &\quad +2\lambda(1-\lambda)\left(f_n - f, \phi - f\right) + \lambda^2 \|\phi - f\|^2. \end{aligned} \tag{11}$$

Next, using assumption (7.3) and the Cauchy-Schwarz inequality, we obtain the following inequality:

$$\left(f_n - f, \sum_{i=1}^{\infty} \alpha_i \phi_i - f \right) \leq \|f_n - f\| \left\| \sum_{i=1}^{\infty} \alpha_i \phi_i - f \right\| \leq \varepsilon_n \varepsilon, \tag{12}$$

where α_i are coefficients of a convex linear combination. Therefore,

$$\sum_{i=1}^{\infty} \alpha_i \left(f_n - f, \phi_i - f\right) \leq \varepsilon_n \varepsilon$$

From the positivity of α_i it follows that we can find such a ϕ_i that

$$\left(f_n - f, \phi_i - f\right) \leq \varepsilon_n \varepsilon. \tag{13}$$

In addition, assumptions (7.1) and (7.2) imply $\|\phi - f\|^2 \leq (\|\phi\| + \|f\|)^2 \leq (K+1)^2$ for any ϕ. Consequently, (11) implies

$$\varepsilon_{n+1}^2 \equiv \inf_{\lambda\in[0,1],\phi} \|(1-\lambda)(f_n - f) + \lambda(\phi - f)\|^2 \tag{14}$$

$$\leq (1-\lambda)^2 \varepsilon_n^2 + 2\lambda(1-\lambda)\varepsilon\varepsilon_n + \lambda^2 (K+1)^2. \tag{15}$$

Choose

$$\lambda = \frac{(K+1)^2 - \varepsilon\varepsilon_n}{(K+1)^2 + \varepsilon_n^2 - 2\varepsilon\varepsilon_n}. \tag{16}$$

This is a valid choice of λ provided that $\varepsilon_n \geq \varepsilon$. For this choice we have the following bound for the error of the next step:

$$\varepsilon_{n+1}^2 \leq \varepsilon_n^2 \frac{(K+1)^2 - \varepsilon^2}{(K+1)^2 + \varepsilon_n^2 - 2\varepsilon\varepsilon_n}. \tag{17}$$

Therefore,

$$\varepsilon_{n+1}^{-2} \geq \varepsilon_n^{-2} \left[1 + \frac{(\varepsilon_n - \varepsilon)^2}{(K+1)^2 - \varepsilon^2}\right] \tag{18}$$

$$\geq \varepsilon_n^{-2} + \frac{1}{(K+1)^2}\left(1 - \frac{\varepsilon}{\varepsilon_n}\right)^2. \tag{19}$$

Consequently, if $\varepsilon_n \geq \varepsilon/(1-\alpha)$, then we have a sequence of inequalities:

$$\varepsilon_{n+1}^{-2} - \varepsilon_n^{-2} \geq \left(\frac{\alpha}{K+1}\right)^2, \tag{20}$$

$$\ldots \tag{21}$$

$$\varepsilon_1^{-2} - \varepsilon_0^{-2} \geq \left(\frac{\alpha}{K+1}\right)^2. \tag{22}$$

Summing them up we get

$$\varepsilon_{n+1}^{-2} \geq \varepsilon_0^{-2} + (n+1)\left(\frac{\alpha}{K+1}\right)^2 \geq (n+1)\left(\frac{\alpha}{K+1}\right)^2, \tag{23}$$

$$\varepsilon_{n+1}^2 \leq \frac{[(K+1)/\alpha]^2}{n+1}. \tag{24}$$

Therefore, either $\varepsilon_n \leq \varepsilon/(1-\alpha)$ and then $\varepsilon_{n+1} \leq \varepsilon/(1-\alpha)$, or $\varepsilon_{n+1}^2 \leq (n+1)^{-1}((K+1)/\alpha)^2$.

QED.

The significance of Theorem 2 is that it shows that a large class of functions can be approximated to a high precision δ by an expansion that has $O(\delta^{-2})$ terms. In particular, the rate of growth of the number of terms is independent of the space dimension. Moreover, the theorem shows that the approximation can be found by the recursive optimization method.

In practice the parameters of the expansion must be estimated from the values of the function on a discrete grid. This introduces an additional error in the approximation - the estimation error. The extent of this error is not analysed in this paper. However, Theorem 2 and results in Niyogi and Girosi (1999) suggest that the number of the gridpoints needed to bring the approximation error below a certain threshold grows only polynomially with dimension.

4 Lower and Upper Bounds

In practical applications, we are often interested not only in an estimate but also in the firm bounds on the option value. Fortunately, the Monte Carlo method and interpolations from the lattice method can be combined for the efficient calculation of these bounds.

Indeed, let $\widehat{V}(x,t)$ denote the approximation to the continuation value function. Define the following stopping rule $\widehat{\tau}$: "Stop if $\widehat{V}(x,t) \leq \pi(x,t)$". It simply tells the holder of the option to stop when the approximate continuation value of the option is smaller than the exercise value. Then the lower bound on the option value at time 0 is given by

$$V_L = E_0 e^{-r\widehat{\tau}} \pi(x_{\widehat{\tau}}, \widehat{\tau}) \leq \sup_{\tau} E_0 e^{-r\tau} \pi(x_\tau, \tau) = V. \tag{25}$$

The expectation can be easily computed using Monte Carlo simulations of possible factor paths.

The upper bound can be computed using the duality method developed in Rogers (2002) and Haugh and Kogan (2004). Let $e^{-rt}M_t$ be a supermartingale. Then

$$\begin{aligned} V &= \sup_{\tau} E_0 e^{-r\tau} \pi(x_\tau, \tau) && (26)\\ &= \sup_{\tau} \left\{ E_0 e^{-r\tau} \left[\pi(x_\tau, \tau) - M_\tau\right] + E_0 e^{-r\tau} M_\tau \right\} && (27)\\ &\leq \sup_{\tau} E_0 e^{-r\tau} \left[\pi(x_\tau, \tau) - M_\tau\right] + M_0 \text{ (because } E_0 e^{-r\tau} M_\tau \leq M_0) && (28)\\ &\leq E_0 \sup_{t} e^{-rt} \left[\pi(x_t, t) - M_t\right] + M_0 && (29) \end{aligned}$$

Consequently, if we manage to find such a supermartingale that the expectation of $e^{-r\tau}\left[\pi(x_\tau,\tau) - M_\tau\right]$ is uniformly small, then we can calculate a good upper bound on the option price.

Intuitively, supermartingale M represents a replicating portfolio that the writer of the option constructs to hedge his position. The portfolio should be designed in such a way that it covers or almost covers the funds needed in the case of the option exercise. The possible deficit in funds is measured by the difference $\pi(x_t,t) - M_t$, and the value of the option cannot be larger than the sum of the replicating portfolio value M_0 and the discounted expectation of the supremum of the deficit.

The main question is how to find a good supermartingale M. One possible way is to choose the martingale distillation of the process $\overline{V}(x_t,t) = e^{-rt}\max\left\{\widehat{V}(x_t,t), \pi(x_t,t)\right\}$. In this case we define

$$e^{-r(t+1)} M_{t+1} = e^{-rt} M_t + E_t \overline{V}(x_{t+1}, t+1) - \overline{V}(x_t, t). \tag{30}$$

The expectation in this expression can be computed numerically. The choice will provide a good upper bound provided that the approximation $\widehat{V}$ is good approximation to the option continuation value.

5 Application

The main goal of this section is to show that the method described above is a viable alternative to approximate Monte Carlo methods. It demonstrates this by computing values of rainbow options. These securities have the following payoff structure:

$$\pi = \max(f(K, S_1, ..., S_n), 0). \tag{31}$$

For example, the put option on the minimum of several assets has

$$f = K - \min(S_1, ..., S_n). \tag{32}$$

This set of securities is a convenient benchmark for testing a pricing method because it was extensively studied in the literature. For the case of European puts on the minimum or the maximum there are analytic formulas derived by Stulz (1982) and Johnson (1987). The American put on the minimum of two assets is priced in Boyle (1988) by a variant of the multinomial lattice method. Boyle, Evnine, and Gibbs (1989) give the results of a lattice method for three-dimensional European puts. Broadie and Glasserman (1997a), Broadie and Glasserman (1997b), Raymar and Zwecher (1997), Broadie, Glasserman, and Gain (1997), and Longstaff and Schwartz (2001) use two- and five-dimensional options as a benchmark for their Monte Carlo simulation methods.

The most essential choice in our algorithm was how many gridpoints to use. The grid was constructed by generating Sobol's quasi-Monte Carlo sequences in a hypercube, and then transfoming them by an appropriate Gaussian distribution. The number of grid points was determined using the cross-validation method.[8] Namely, the grid was divided into two portions: training and validation sets. The approximation was found using the training set only, and its quality was evaluated on the validation set by computing the mean squared error (MSE). The new terms in the approximative expansion were added only if they decreased the MSE criterion. The lowest value of the MSE was taken as the performance measure for a given number of data points. If it was unsatisfactory, the number of grid points was increased and the procedure repeated.

Table 1 summarizes the results of the application of the Interpolative Lattice (IL) method to European and American puts on the minimum of two assets. The results are compared to the results of the analytical formula and to the results of the lattice method from Boyle (1988). This table shows that IL gives high precision results for European options where the difference is on average less than a cent. For American puts the results of the IL and Boyle methods are slightly different but still are very close to each other.

Table 2 shows the results of IL application to valuation of various put options on three assets. The parameters are as in Boyle, Evnine, and Gibbs (1989) but since the results in Boyle, Evnine, and Gibbs (1989) apparently contain a miscalculation, the results obtained by the Monte Carlo method are used as a

[8] See Hastie, Tibshirani, and Friedman (2001) for a fuller description of the cross-validation method.

benchmark.[9] The results for European options are generally in good agreement with the Monte Carlo results, with the difference less than 5% of the option value.

Table 3 shows the results of pricing American puts on various functions of five asset prices. The results are compared with results by the Least Squares Monte Carlo, binomial and Berridge-Schumacher methods as they are reported in Berridge and Schumacher (2004). The results are in good agreement. For the put on the geometric average of the five asset prices, the dimension reduction is possible and we are able to compute the exact price. In this case , the interpolative lattice method slightly overestimates the true price but the difference is less than 3%. The interpolative lattice method gives an estimate that in three cases exceeds and in one case falls below the estimate given by Least Squares Monte Carlo. This evidence suggests that the interpolative lattice method tends to give an estimate that is biased upward.

The pricing times of the interpolative lattice method are rather long in the case of five assets – around 2 hours. Most of the time goes into the search for a good adaptive approximation.

In summary, it appears that pricing by adaptive interpolation is a viable alternative to other methods of pricing multidimensional options including Least Squares Monte Carlo.

6 Conclusion

This paper proposes a novel variant of the lattice option pricing, which is based on modern methods of adaptive interpolation. In multiple dimensions the method allows using irregular grids and thus avoids both the curse of dimensionality and the necessity to build recombining trees. The method is easy to apply to many examples of derivative securities and its practical viability is corroborated by numerical examples.

A theorem is demonstrated that suggests that the new method is less likely to be vulnerable to the curse of dimensionality. Much further research is needed, however, to determine the convergence properties of the new method.

References

Barraquand, J. and D. Martineau (1995). Numerical valuation of high-dimensional multivariate American securities. *Journal of financial and quantitative analysis 30*, 383–405.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory 39*(3), 930–945.

[9] The Monte Carlo method has been tested on 2-dimensional options from Boyle (1988), and the results have been found to be in remarkable agreement with the results of Boyle (1988).

Berridge, S. J. and J. M. Schumacher (2004). An irregular grid approach for pricing high-dimensional american options. Tilburg University/CentER Discussion Paper 2004-18.

Boyle, P. P. (1988). A lattice framework for option pricing with two state variables. *Journal of Financial and Quantitative Analysis 23*(1), 1–12.

Boyle, P. P., J. Evnine, and S. Gibbs (1989). Numerical evaluation of multivariate contingent claims. *Review of Financial Studies 2*(2), 241–250.

Breiman, L. (1993). Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory 39*(3), 999–1013.

Broadie, M. and P. Glasserman (1997a). Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control 21*, 1323–1352.

Broadie, M. and P. Glasserman (1997b). A stochastic mesh method for pricing high-dimensional American options. Working paper.

Broadie, M., P. Glasserman, and G. Gain (1997). Enhanced Monte Carlo estimates for American option prices. *Journal of Derivatives 5*(1).

Carriere, J. (1997). Valuation of early exercise price of options using simulations and nonparametric regression. *Insurance, Mathematics and Economics 19*, 19–30.

Cox, J., S. Ross, and M. Rubinstein (1979). Option pricing: A simplified approach. *Journal of Financial Economics 7*, 229–263.

DeVore, R. A. (1998). Non-linear approximation. *Acta Numerica 7*, 51–150.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Haugh, M. and L. Kogan (2004). Pricing american options: A duality approach. *Operations Research*. Forthcoming.

Hull, J. C. (1999). *Options, Futures, and Other Derivatives* (4 ed.). Prentice Hall Incorporated.

Johnson, H. (1987). Options on the minimum or the maximum of several assets. *Journal of Financial and Quantitative Analysis 22*, 277–283.

Jones, L. K. (1992). A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Annals of Statistics 20*(1), 608–613.

Kac, M. (1949). On distribution of certain Wiener functionals. *Transactions of the American Mathematical Society 65*(1), 1–13.

Kamrad, B. and P. Ritchken (1991). Multinomial approximating models for options with K state variables. *Management Science 37*(12).

Karatzas, I. and S. E. Shreve (1991). *Brownian Motion and Stochastic Calculus* (2 ed.). Springer-Verlag.

Longstaff, F. A. and E. S. Schwartz (2001). Valuing American options by simulations: A simple least-squares approach. *Review of Financial Studies 14*(1), 113–147.

Madan, D. B., F. Milne, and H. Shefrin (1989). The multinomial option pricing model and its Brownian and Poisson limits. *Review of Financial Studies 2*(2).

McCarthy, L. A. and N. J. Webber (2001). Pricing in three-factor models using icosahedral lattices. *Journal of Computational Finance 5*(2), 1–36.

Niyogi, P. and F. Girosi (1999). Generalization bounds for function approximation from scattered noisy data. *Advances in Computational Mathematics 10*, 51–80.

Raymar, S. B. and M. J. Zwecher (1997). Monte Carlo estimation of American call options on the maximum of several stocks. *Journal of Derivatives 5*, 7–23.

Rogers, L. C. G. (2002). Monte carlo valuation of american options. *Mathematical Finance 12*, 271–286.

Stulz, R. M. (1982). Options on the minimum or the maximum of two risky assets. *Journal of Financial Economics 10*, 161–185.

Temlyakov, V. N. (2003). Non-linear methods of approximation. *Foundations of Computational Mathematics 3*, 33–107.

Tilley, J. (1993). Valuing American options in a path simulation model. *Transactions of the Society of Actuaries 45*, 83–104.

Tsitsiklis, J. N. and B. V. Roy (2001). Regression methods for pricing complex american-style options. *IEEE Transactions on Neural Networks 12*(4), 694–703.

Wilmott, P. and S. Howson (1995). *The mathematics of financial derivatives: A student introduction* (1 ed.). The Press Syndicate of the University of Cambridge.

Table 1. Value of Put on Minimum of Two Assets

Strike	European Put			American Put			IL Pricing Time (min)
	IL	Exact	Difference	IL	Boyle	Difference	
35	1.391	1.387	0.004	1.433	1.423	0.01	8
40	3.811	3.798	0.013	3.894	3.892	0.002	10.4
45	7.504	7.500	0.004	7.720	7.689	0.031	10.6

Following Boyle (1988) the parameters are $S_1 = 40,\ S_2 = 40,\ \sigma_1 = 0.20,\ \sigma_2 = 0.30,$ $\rho = 0.5,\ r = 5 -$ percent per annum effective $= 0.048790$ continuously, $T = 7$ months $= 0.5833333$ years, and exercise prices = 35, 40, 45. The number of time steps used in Boyle (1988) is reported as 50.

The IL (interpolated lattice) pricing uses 3 time periods and an irregular grid with 4,000 grid points and 1,600 descendants from each grid point. The grid points were constructed by transforming a Sobol sequence in a unit square by an inverse normal distribution. The descendants were also constructed using a transformed Sobol sequence.

The number of terms in each adaptive approximation was limited to 8. Each new term in an approximation was found using function “fminu” from the Matlab optimization toolbox.

The pricing was done on a standard IBM PC with Pentium microprocessor and Windows 2000 operating system. The entire code is written in Matlab and available on request.

Table 2. Value of Puts on Various Functions of Three Assets

Put type	European			American	Pricing time (min)
	IL	MC	Difference	ILM	
Maximum	0.993	0.989[1]	0.004	2.028	13.2
Minimum	7.639	7.656[1]	-0.017	8.755	12.7
Geom. average	2.858	2.865[1]	-0.007	3.642	11.5
Arith. average	2.698	2.698[1]	0	3.823	12.5

Following Boyle, Evnine, and Gibbs (1989), the parameters are $K = 100$, $S_i = 100$, $r = 0.1$ -- percent per annum, $\sigma_i = 0.2$, $\rho_{ij} = 0.5$, $T = 1$, where $1 \le i < j \le 3$. Boyle, Evnine, and Gibbs use 80 time steps and the Richardson extrapolation.

The IL method uses 3 time steps, an irregular grid with 4,000 points, and 1600 descendents from each grid point.

[1] The values of European options given by Boyle, Evnine, and Gibbs (1989) appear to be miscalculated for the chosen set of parameters. For this reason, the numbers in the table are based on the Monte Carlo method of option pricing that used 9×10^6 iterations. The standard deviations of the estimates are about 0.001, 0.003, 0.002, and 0.002 for maximum, minimum, geometric average, and arithmetic average puts, respectively. The method has been tested on the example from Boyle (1988) (see Table 1) and its results agree with the results from Boyle (1988).

Table 3. Value of Puts on Various Functions of Five Assets

Put type	European	American/Bermudan					IL pricing time (min)
		IL	Binomial	LSM	Berridge-Schumacher	Exact	
Maximum	0.114	0.247	0.230	0.275	0.276	-	131.2
Minimum	5.67	5.950	5.841	5.815	5.847	-	134.3
Geom. average	1.158	1.379	1.340	1.348	1.350	1.342	135.0
Arith. average	1.013	1.269	1.235	1.241	1.2460	-	148.5

Following Berridge and Schumacher(2002), the parameters are $K = 40$, $S_i = 40$, $r = 1.06$ -- per annum, $\sigma_i = 0.2$, $\rho_{ij} = 0.25$, $T = 1$, where $1 \le i, j \le 5$. Berridge and Schumacher use average of runs on 50 grids of size 1000.

The results for binomial and LSM methods are as reported in Berridge and Schumacher. The IL method uses 3 time steps, an irregular grid with 5,000 points, 1600 descendants from each grid point, and up to 20 functional elements in the approximating expansion at each time step.