An efficient grid lattice algorithm for pricing American-style options

Zhongkai Liu
Department of Statistics,
North Carolina State University,
Raleigh, NC 27695, USA
Email: zliu15@ncsu.edu

Tao Pang*
Department of Mathematics,
North Carolina State University,
Raleigh, NC 27695, USA
Email: tpang@ncsu.edu
*Corresponding author

Abstract: Option pricing is an important area of research in the finance community. In this paper, we develop a computationally feasible and efficient lattice algorithm in pricing American-style options. The key idea is to build a time adjusted grid lattice model and afterwards implement backward induction to price options. The time adjusted grid lattice guarantees high accuracy in relatively few discrete finite nodes. To illustrate the performance of the lattice algorithm, European and American options are priced separately, and results are compared to other popular methods in terms of both accuracy and efficiency. All suggest that the proposed lattice algorithm does a better job. Moreover, the fast convergence behaviours of the lattice algorithm as well as the relationship between the converged option price and the number of determination dates are studied as well.

Keywords: lattice algorithm; time adjusted grid lattice; European options; American options.


Biographical notes: Zhongkai Liu holds a Bachelor of Science degree in Statistics with a minor in finance from Shanghai University of Finance and Economics (2011) and a Master of Science degree in Statistics from North Carolina State University (2013). He is currently a PhD candidate in Department of Statistics at North Carolina State University and is expected to graduate in May, 2016.

Tao Pang obtained his PhD in Applied Mathematics from Brown University (2002). He obtained his MS in Mathematics (1997) and BS in Mathematics (1994) both from University of Science and Technology of China. He is currently an Associate Professor of Financial Mathematics at North Carolina State University. He is a Chartered Financial Analyst (CFA) holder and a Financial Risk Manager (FRM) holder. He is also a co-Director of the Raleigh,
1 Introduction

The owner of a put (call) option has the right but no obligation to sell (buy) an underlying asset at the exercise price. European and American options are the two major types of options, while the names are not based on where they are traded but based on the exercise options that come with them. European options can only be exercised on the maturity date, while American options can be exercised any time up to maturity date. The pricing of options plays an important role both in theory and real derivative markets.

European options have been valued by the celebrated Black-Scholes (BS) formula, a closed-form solution derived by Black and Scholes (1973) and Merton (1973). However, in the case of American options, because of the early exercise possibility, the pricing problem leads to complications for analytic calculations, lacking closed-form solutions. McKean (1965) and Moerbeke (1976) showed that the price of an American option satisfies a partial differential equation (PDE) with a boundary condition changing in time to maturity, called an optimal exercise boundary. Various numerical and approximation schemes have been studied to price American options, which can be divided into three groups including numerical techniques, analytical approximation methods, and Monte Carlo simulations.

Numerical techniques were initiated by a finite difference (FD) scheme (Merton et al., 1977) with its convergence proved by Jaillet et al. (1990). The key idea is to transform the PDE into a system of linear equations and thus the solutions to the equation system provide the option price for all times and underlying asset prices. As an extension, Dempster and Hutton (1999) used a FD approximation and derived a linear programming problem at each time step and each of these problems is solved by the simplex method, which is comparable to the projected successive over-relaxation technique, another numerical pricing method posed in Cryer (1971). Other related works and extensions using FD method can be found in Hull and White (1990), Wilmott (1995), Tavella and Randall (2000), Forsyth and Vetzal (2002), Duffy (2006), Tangman et al. (2008), Zhu and Chen (2011) and Kim et al. (2013). Another popular method in this group is the binomial tree (Cox et al., 1979) and its variant trinomial tree method (Boyle, 1986). They are widely used because of their simplicity and ease to adapt to any kind of options. The key idea is to build a lattice which discretises the time space of the asset price and then discounts, using risk neutral valuation, the cash flows backwards from maturity until the beginning of the contract. The extensions of the binomial tree method include the accelerated binomial model (Breen, 1991) which uses Richardson’s extrapolation (Marchuk and Shaidurov, 1983) to reduce the number of steps, and the binomial BS method (Broadie and Detemple, 1996) which is based on using the BS formula one step before expiration. There is another subgroup of numerical methods using integral representations of the optimal exercise boundary and the price function developed in Kim (1990), Jacka (1991), Carr et al. (1992), including Huang et al. (1996) as an example of such a method and Ju (1998) as an extension. As a technique originated in physics problems and based on Landau (1950) transform, front fixing method was proposed in
Wu and Kwok (1997) for solving free boundary problem in the field of option pricing. Other relevant methods related to the fixed domain transformation include Nielsen et al. (2002), Sevcovic (2007), Zhang and Zhu (2009) and Egorova et al. (2014). Furthermore, Muthuraman (2008) presented a moving boundary approach to convert the free boundary problem into a sequence of fixed boundary problems. And recently Chockalingam and Muthuraman (2015) proposed a variant of this moving boundary approach, which is more efficient and flexible.

The second group of methods use analytical approximations to represent the price of an option. One of the first was the quadratic approximation (Barone-Adesi and Whaley, 1987) based on the approach in MacMillan (1986), which seeks an approximate solution to the BS PDE by neglecting a quadratic term for the exercise premium. Afterwards, Ju and Zhong (1999) refined the quadratic formula, pricing long maturity options more accurately. However, these methods are not convergent. By viewing an American option as a sequence of Bermudan options, Geske and Johnson (1984) proposed an approximation method and Bunch and Johnson (1992) gave a modified version. By regarding put options as the limit to a sequence of perpetual option values, Carr and Faguet (1996) presented an analytical approximation method. More recently, a semi-closed form of solution for the option price based on a Taylor series expansion was derived in Zhu (2006).

Due to the complexity of the underlying dynamics and multi-factors situation, Monte Carlo simulation methods, simulating paths for asset prices, become popular in financial theory and practice. After Boyle (1977) first introduced Monte Carlo simulation to study European option pricing and Tilley (1993) first applied simulation to American option pricing, Fu and Hu (1995), Carriere (1996), Grant et al. (1997), Broadie and Glasserman (1997), Tsitsiklis and Van Roy (1999), Longstaff and Schwartz (2001) and Ibáñez and Zapatero (2004) priced options based on Monte Carlo simulation. Among them, the least squares Monte Carlo (LSM) approach (Longstaff and Schwartz, 2001) gains most popularity because of its simplicity and accuracy. They applied least-squares regressions in which the explanatory variables are certain polynomial functions and estimated the continuation values of a number of derivatives, where only in-the-money paths in the regressions are used in order to increase efficiency. In addition, Clément et al. (2002) gave more derivation results as well as the convergence rate of LSM method. Jia (2009) gave a summary of all the Monte Carlo methods in pricing American options and made a comparison, concluding that the LSM approach is more suitable for problems in higher dimensions than other comparable Monte Carlo methods. As pricing multi-asset options poses great computational challenges, high-performance option pricing algorithms based on parallel simulation comes into being. Wan et al. (2006) proposed a parallel quasi-Monte Carlo approach to price multidimensional American options. A larger scale simulation model can be found in Chang et al. (2012) and Hu et al. (2015).

Although there are various approaches so far in pricing American options, expensive computational cost and slow convergence problem are complained a lot. In this paper, we develop an efficient grid lattice approach for pricing American-style options. Not only does the proposed algorithm gain similar accuracy to the current popular pricing models in the market, but it runs much faster comparatively. Motivated by Xiao (2011) that designed a lattice algorithm for the LIBOR market model (LMM), we build a time adjusted grid lattice that uses a variable substitution to shift the centre of an underlying asset price distribution to zero, which ensures that the distribution is symmetric and can be represented by a relatively small number of discrete points. The shift transformation is
the key to achieve high accuracy in relatively few discrete finite nodes. To illustrate the performance of the lattice algorithm, we price European and American options separately, and compare the results with current popular methods in terms of accuracy and efficiency. Here we restrict our focus to only put options since pricing call options is conceptually the same with identical equations, just different boundary conditions.

The paper is organised as follows. In Section 2, some preliminaries and the background about pricing American options are given. Section 3 is devoted to elaborating the lattice algorithm. The numerical implementations are presented in Section 4. For European options, we compare the lattice algorithm with the closed-form BS formula, while the FD method and LSM approach are compared to the lattice algorithm in pricing American options. Section 5 gives concluding remarks.

2 Preliminaries and backgrounds

We consider a complete filtered probability space \((\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{0 \leq t \leq T}, \mathbb{P})\) over a finite time horizon \([0, T]\), where the state space \(\Omega\) is the set of all possible realisations of the stochastic economy between time 0 and T with element \(\omega\) as a sample path. \(\mathcal{F}\) is the \(\sigma\)-field of distinguishable events at time T and \(\{\mathcal{F}_t\}_{0 \leq t \leq T}\) is the natural filtration generated by the underlying asset price process. \(\mathbb{P}\) is a probability measure defined on the elements of \(\mathcal{F}\).

Assume that the underlying asset price follows a geometric Brownian motion under the market probability measure \(\mathbb{P}\):

\[
dS_t = \mu S_t dt + \sigma S_t dW_t,
\]

where \(W_t\) is a standard Brownian motion under \(\mathbb{P}\). As a result of no-arbitrage theory, we assume the existence of an equivalent martingale measure \(\mathbb{Q}\) for this economy. Then under the risk-neutral measure \(\mathbb{Q}\), the asset price process follows the stochastic differential equation (SDE) as follows.

\[
dS_t = rS_t dt + \sigma S_t d\tilde{W}_t, \tag{2.1}
\]

where \(r\) and \(\sigma\) represent the risk-free interest rate and volatility of the underlying asset price separately and \(\tilde{W}_t\) is a standard Brownian motion under \(\mathbb{Q}\). If there is any dividend with continuous dividend \(q\), the drift term in equation (2.1) will be \((r - q)S_t\) instead of \(rS_t\). Without loss of generality, we assume that the asset does not pay dividends.

We study an American-style put option on an underlying asset (stock) with exercise price \(K\) and expiration date \(T\). For a large positive integer \(N > 0\), define

\[
\Delta t = \frac{T}{N}, \quad t_i = i\Delta t, \quad i = 0, 1, 2, \ldots, N.
\]

We focus on discrete time steps \(0 = t_0 < t_1 < \ldots < t_N = T\) and decide the optimal stopping policy at each of the \(N + 1\) exercise dates. In practice, American options are continuously exercisable before maturity and thus we can take \(N\) to be sufficiently large. The valuation of the option is denoted as \(P(\omega; t_i)\), where \(\omega \in \Omega\) is a sample path and \(i = 0, \ldots, N\). The payoff function of the put option at maturity date \(T\) is

\[
P(\omega; T) = (K - S_T (\omega))^+, \quad \max \{ K - S_T (\omega), 0 \}. \tag{2.2}
\]
The value of an American option equals the maximised value of the discounted cash flows from the option, where the maximum is taken over all stopping times with respect to the filtration \( \mathcal{F}_t \) (Hull, 2011). Specifically, at time \( t_i \), the value of immediate exercise equals the cash flow from immediately exercising the option, which is \((K - S_{t_i})_+\). On the other hand, by the no-arbitrage theory, the value of continuation is given by taking the expectation of the remaining discounted cash flows with respect to the risk-neutral measure \( Q \). Consequently, with the boundary condition in equation (2.2), the value of the option at \( t_i \) is the maximum of immediate exercise value and continuation value, which can be mathematically expressed as

\[
P(\omega; t_i) = \max \left\{ (K - S_{t_i}(\omega))_+, \mathbb{E}_Q \left[ e^{-r(t_i-t)} P(\omega; t_{i+1}) \mid \mathcal{F}_{t_i} \right] \right\}.
\]  

(2.3)

This representation shows that the value of the option is maximised pathwise, and thus unconditionally the investor will exercise the option as long as the immediate exercise value is greater than or equal to the value of continuation.

Among various numerical techniques in pricing American options, FD method, which creates a mathematical relationship which links together every point on the solution domain like a chain, stands out because all options satisfy the BS PDE

\[
\frac{\partial P}{\partial t} + rS \frac{\partial P}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} - rS = 0,
\]

(2.4)

denote the first and second derivative of the option price over stock price \( S \) separately. In particular, the price \( P(t, S) \) of an American put option with maturity \( T \) and strike price \( K \) satisfies the following free boundary problem (see Etheridge, 2002):

\[
\begin{align*}
\left\{ \frac{\partial P}{\partial t} + rS \frac{\partial P}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} - rS & < 0, \quad \forall t \in [0, T], S \leq S_f(t); \\
P(t, S) & = K - S, 
\end{align*}
\]

(2.5)

\[
\begin{align*}
\left\{ \frac{\partial P}{\partial t} + rS \frac{\partial P}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} - rS & > 0, \quad \forall t \in [0, T], S > S_f(t); \\
P(t, S) & > (K - S)_+, 
\end{align*}
\]

(2.6)

\[
\frac{\partial P}{\partial S}(t, S_f(t)) = -1, \quad P(t, S_f(t)) = (K - S_f(t))_+ \;
\]

(2.7)

\[
P(T, S) = (K - S)_+, \quad \text{where } S_f(t), 0 \leq t \leq T \text{ is the free boundary that is part of the solution.}
\]

(2.8)
An efficient grid lattice algorithm for pricing American-style options

Based on the asset value discretisation and time discretisation, we can solve equation (2.4) backwards in time from expiry to the present by approximating the three derivative terms. The popular approximation methods include explicit Euler, implicit Euler and Crank-Nicolson method (Morton and Mayers, 2005). Then the rest is to solve a linear system at each time step. The key advantage of FD is that we can automatically determine the option values at every mesh point in the domain. If we require an option value for a particular value of $S$ and $t$, all we need is to extract that point from the solution matrix. Although we do not have a continuous solution as a result of using a finite discretisation, we can refer to some kind of interpolation techniques to extract an approximation if we want to require an option value which does not coincide with one of our mesh points.

As an alternative, the LSM approach, which uses least-squares regression to estimate the conditional expected payoff to the option holder from continuation, is also widely used on behalf of the Monte Carlo simulation because of its applicability in dealing with path-dependent and multi-factor situations where traditional FD techniques cannot be used. Details as well as the comparison with FD method can be found in Longstaff and Schwartz (2001), showing a high accuracy and flexibility of the LSM method.

3 Lattice algorithm

3.1 Time adjusted grid lattice

For convenience, we assume the constant volatility:

Assumption 1: The underlying asset (stock) price has constant instantaneous volatility regardless of time $t$.

Otherwise, we should replace $\sigma$ by $\sigma(t)$, a function of time $t$, in the SDE (2.1) and our approach will work the same way.

Let $\tilde{S}_t$ be the discounted stock price given by $\tilde{S}_t = e^{-\sigma t}S_t$, then by Itô’s Lemma, we get the SDE for $\tilde{S}_t$ as follows.

$$d\tilde{S}_t = \sigma \tilde{S}_t d\tilde{W}_t,$$

where $\tilde{W}_t$ is a Brownian motion under the risk-neutral measure $Q$ by Girsanov’s theorem in Girsanov (1960). The discounted stock price $\tilde{S}_t$ is a martingale under the risk-neutral measure $Q$. It is easy to find the solution of (3.1) as the follows:

$$\tilde{S}_t = S_0 \exp\left(-\frac{\sigma^2}{2} t + \sigma \tilde{W}_t\right).$$

Taking the expectation of the above equation conditioned on the information at $t = 0$ gives
\[
\mathbb{E}_Q[\tilde{S}_t] = \mathbb{E}_Q\left[ S_0 \exp\left( -\frac{\sigma^2}{2} t + \sigma \tilde{W}_t \right) \right] \\
= S_0 \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{w^2}{2t}} d\sigma \\
= S_0 \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{(w-\sigma)^2}{2t}} d\sigma \\
= S_0 \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{y^2}{2t}} dy \quad \text{(set } y = w - \sigma t) \\
= S_0.
\]

Typically, to numerically evaluate the integral \( \int_{-\infty}^{\infty} e^{-\frac{(w-\sigma)^2}{2t}} d\sigma \), we would use \( \int_{-A}^{A} e^{-\frac{y^2}{2t}} dy \) for some given \( A > 0 \). Due to the fact that the function \( e^{-\frac{(w-\sigma)^2}{2t}} \) is centred at \( \sigma t \) and is symmetric about \( \sigma t \), the approximation \( \int_{-A}^{A} e^{-\frac{y^2}{2t}} dy \) will not be as good as \( \int_{-\sigma t}^{\sigma t} e^{-\frac{(w-\sigma)^2}{2t}} d\sigma \), which is equivalent to \( \int_{-\sigma t}^{\sigma t} e^{-\frac{y^2}{2t}} dy \).

Therefore, we will build our lattice based on the process \( Y_t = \tilde{W}_t - \sigma t \), a Brownian motion with drift under \( \mathbb{Q} \), instead of the Brownian motion \( \tilde{W}_t \). In other words, for a given \( A > 0 \) that is large enough, we will build a grid lattice for \( Y_t \in [-A, A] \) instead of a lattice for \( \tilde{W}_t \). Actually, since \( Y_t = \tilde{W}_t - \sigma t \), we have that

\[
Y_t \in [-A, A] \Leftrightarrow \tilde{W}_t \in [-A - \sigma t, A - \sigma t].
\]

So we call this grid lattice as the time-adjusted grid lattice. Compared to the grid lattice for \( \tilde{W}_t \in [-A, A] \), this time-adjusted grid lattice will improve the accuracy for estimating expectations for functions of \( S_t \) numerically.

The variable substitution used for derivation \( y = w - \sigma t \), which corresponds to \( Y_t = \tilde{W}_t - \sigma t \), is the key idea of the lattice algorithm as explained in Xiao (2011). This variable transformation ensures that the distribution is centered on zero and symmetric, which achieves high accuracy when we express the stock price in discrete finite form and use numerical integration to calculate the expectation. In fact, without this linear transformation, the lattice method in pricing American options either does not exist or introduces too much error for longer maturities.

After applying the variable substitution, equation (3.2) can be expressed as

\[
\tilde{S}_t = S_0 \exp\left( -\frac{\sigma^2}{2} t + \sigma \tilde{W}_t \right) = S_0 \exp\left( \frac{\sigma^2}{2} t + \sigma Y_t \right),
\]
An efficient grid lattice algorithm for pricing American-style options

where \( Y_t = \tilde{W}_t - \sigma t \) is a Brownian motion with drift under the risk neutral measure \( Q \).

Consequently, based on equation (3.4) as well as the relationship \( \tilde{S}_t = e^{-\sigma t} S_t \), the stock price is given by

\[
S_t = S_0 \exp \left( rt + \frac{\sigma^2}{2} t + \sigma Y_t \right).
\]  

(3.5)

3.2 Lattice procedure

As a generic term, lattice is a layered graph that attempts to transform a continuous-time and continuous-space underlying process into a discrete-time and discrete-space process, where the nodes at each level represent the possible values of the underlying process in that period. In pricing financial products, there are mainly two types of lattices, including tree lattices and grid lattices. Traditional binomial tree (Cox et al., 1979) belongs to the tree lattices, assuming that the underlying process has two possible outcomes at each stage. On the other hand, grid lattices, also called rectangular lattices, are built in a rectangular FD grid, permitting the underlying process to change by multiple states. Examples include Amin (1993), Gandhi and Hunt (1997), Martzoukos and Trigeorgis (2002) and Das (2011), indicating that the grid lattices are more realistic and convenient for the implementation of a Markov chain solution.

As an instance of the grid lattice, Figure 1 shows the state space for the underlying process \( y_t \) over two discrete time periods. The starting state \( y_0 \) at valuation date \( t_0 \) is the single root of the lattice. At each date \( t_i \), the underlying process \( y_{ti} \) is discretised into a number of vertical nodes, where the number of nodes depends on the convergence rate. The node \( y_{n,j} \) can evolve to any discrete state in next time point with certain transition probability, which will be explained later. Lattice approaches are ideal for pricing early exercise products given their backward-in-time nature described in Xiao (2011). Our proposed lattice algorithm as introduced below, a grid lattice model for pricing options, also uses backward induction but exploits the Gaussian structure to gain extra efficiencies as well.

Figure 1 Grid lattice for the underlying process \( y_t \) over two discrete time periods
To begin with, we need to build the lattice framework, which is used to model the random process in equation (3.5). Unlike traditional trees, we only position nodes at the $N$ discretised determination dates $0 = t_0 < t_1 < \ldots < t_N = T$. Such discretised schemes basically convert the Brownian motion into discrete variables. Specifically, at any determination date $t_i$, we discretise the Brownian motion to be equally spaced as a grid of nodes $y_{i,j}$, for $j = 1, \ldots, M_i$. The number of nodes $M_i$ and the space between nodes $\delta_{i,j} = y_{i,j} - y_{i,j-1}$ at each determination date can vary depending on the length of time and accuracy requirement. In order to guarantee a certain level of accuracy, the nodes should cover a certain number of standard deviations of the Gaussian distribution. Consequently, the discrete form of the stock price, conditional on being in the state $y_{i,j}$ at time $t_i$, can be expressed as

$$S_{i,j} = S_0 \exp\left(\frac{r t_i + \sigma^2 t_i}{2} + \sigma y_{i,j}\right).$$

(3.6)

Another useful information in performing valuation for the underlying instrument is the transition probability density function. Since the underlying state process $\tilde{W}_t$ in equation (3.2) is a standard Brownian motion, the transition probability density from state $(\tilde{w}_{i,j}, t)$ to state $(\tilde{w}_{i',j'}, T)$ is given by

$$p\left(\tilde{w}_{i,j}, t; \tilde{w}_{i',j'}, T\right) = \frac{1}{\sqrt{2\pi(T-t)}} \exp\left(-\frac{(\tilde{w}_{i',j'} - \tilde{w}_{i,j})^2}{2(T-t)}\right).$$

(3.7)

Applying the variable substitution $Y_t = \tilde{W}_t - at$, equation (3.7) will be expressed as

$$p\left(y_{i,j}, t; y_{i',j'}, T\right) = \frac{1}{\sqrt{2\pi(T-t)}} \exp\left(-\frac{(y_{i',j'} - y_{i,j} + \sigma T - at)^2}{2(T-t)}\right).$$

(3.8)

According to (2.3), the American put option value at time $t_i$ is the maximum of immediate exercise value $(K - S_i(\omega))$, and continuation value $\mathbb{E}_Q[e^{-r(t_i+\delta)}P(\omega; t_{i+1}) | F_i]$. To calculate the continuation value, we use the proposed time adjusted grid lattice. Consequently, the American put option value at time $t_i$ and state $y_j$ can be expressed as

$$P(t_i; y_j) = \max\left\{(K - S_i(\omega))_+, P^c(t_i; y_j)\right\},$$

(3.9)

where $P^c(t_i; y_j)$, the continuation value in state $y_j$ at time $t_i$, is in the form of

$$P^c(t_i; y) = \frac{e^{-r(t_i+\delta)}}{\sqrt{2\pi(t_{i+1} - t_i)}} \int_{-\infty}^{\infty} P(t_{i+1}; y) e^{-\frac{(y - y_j + \sigma(t_{i+1} - t_i))^2}{2(t_{i+1} - t_i)}} dy.$$  

(3.10)

Finally, in the process of computing the value of continuation by taking the expectation of the remaining discounted cash flows with respect to the risk neutral measure $Q$, numerical fast integration algorithms are needed. For ease of illustration, in the paper we refer to the trapezoidal rule integration (Atkinson, 2008), while cubic spline integration (Bartels et al., 1995) and fast Fourier transform (Van Loan, 1992) are also good candidates.
4 Numerical implementation

In this section, we implement the lattice algorithm on pricing European options and American options respectively. We focus on put options with strike price \( K = 40 \) and try a couple of settings with different initial stock prices \( S_0 \), volatility \( \sigma \), maturity time \( T \), and discretised exercise times \( N \). The \( N \) determination dates, \( 0 = t_0 < t_1 < \ldots < t_N = T \), are evenly spaced with \( \Delta t = t_i - t_{i-1} \) for \( i = 1, \ldots, N \). The short term interest rate is fixed at \( r = 0.06 \). Comparisons between the lattice algorithm and other popular pricing approaches are illustrated in terms of accuracy and efficiency. The valuation procedure of the lattice algorithm is as follows.

Step 1 Create the lattice in terms of determining the number of nodes \( M_t \) and the grid space \( \delta y = y_j - y_{j-1} \) (space between nodes) for \( j = 1, \ldots, M_t \) at each determination date \( t \), which depends on the length of time and the accuracy requirement. In order to simplify the illustration, we choose the same settings across the lattice. In other words, the nodes are equally spaced and symmetric with fixed number of nodes \( M_t \) and common \( \delta y \) at each determination date. In addition, the nodes should be enough to cover a certain number of standard deviations of the standard Gaussian distribution to guarantee a certain level of accuracy. In particular, we set \( \delta y = \frac{2A}{M-1} \) and \( y_{ij} = -A + j\delta y, j = 1, 2, \ldots, M \). It is easy to check that \( y_{ij}, j = 1, 2, \ldots, M \) covers the interval \( [-A, A] \).

Step 2 Find the option value at each final node. At the final maturity date \( t_N = T \), the payoff of the option, either European or American type, in any state \( y_j \) is given by

\[
P(T; y_j) = (K - S_T) \cdot \max \{K - S_T, 0\},
\]

where \( j = 1, \ldots, M \).

Step 3 Find the option value at the penultimate determination date \( t_{N-1} \). According to equation (3.9), the American option value \( P_{Am} \) in any state \( y_j \) can be expressed by

\[
P_{Am} (t_{N-1} ; y_j) = \max \{ (K - S_{t_{N-1}}) \cdot \max \{K - S_{t_{N-1}}, 0\}, P (t_{N-1} ; y_j) \},
\]

where \( P (t_{N-1} ; y_j) \) denotes the discounted continuation value in state \( y_j \) at time \( t_{N-1} \). Based on the transition probability density in equation (3.8) and the trapezoidal rule integration, the continuous value can be given by

\[
P (t_{N-1} ; y_j) = e^{-\frac{\gamma M}{\sqrt{2\pi\Delta t}}} \int_{-\infty}^{\infty} P(T; y) \exp \left( -\frac{(y - y_j + \sigma\Delta t)^2}{2\Delta t} \right) dy
\]

\[
\approx e^{-\frac{\gamma M}{\sqrt{2\pi\Delta t}}} \frac{\delta y}{2} \sum_{k=2}^{M} \left[ P(T; y_k) \exp \left( -\frac{(y_k - y_j + \sigma\Delta t)^2}{2\Delta t} \right) \right] + P(T; y_{k-1}) \exp \left( -\frac{(y_{k-1} - y_j + \sigma\Delta t)^2}{2\Delta t} \right).
\]
Step 4 Find the option value at earlier nodes. Let us go to the determination date $t_i$. According to equation (3.9), the American option value $P_{Am}$ in any state $y_j$ can be expressed by

$$P_{Am}(t_i; y_j) = \max\{K - S_{y_j}, P^c(t_i; y_j)\}, \quad (4.4)$$

where $P^c(t_i; y_j)$ denotes the discounted continuation value in state $y_j$ at time $t_i$. Based on the transition probability density in equation (3.8) and the trapezoidal rule integration, the continuous value can be given by

$$P^c(t_i; y_j) = e^{-r\Delta t} \int_{-\infty}^{\infty} P_{Am}(t_{i+1}; y) \exp\left(-\frac{(y - y_j + \sigma \Delta t)^2}{2\Delta t}\right) dy$$

$$\approx e^{-r\Delta t} \sum_{k=2}^{M} \frac{\delta_k}{2} \left[\int P_{Am}(t_{i+1}; y_k) \exp\left(-\frac{(y_k - y_j + \sigma \Delta t)^2}{2\Delta t}\right) dy + P_{Am}(t_{i+1}; y_{k-1}) \exp\left(-\frac{(y_{k-1} - y_j + \sigma \Delta t)^2}{2\Delta t}\right)\right], \quad (4.5)$$

Similarly, we can implement backward induction to compute the discounted continuation value in each state at every time by setting $i = N - 2, \ldots, 1$.

Step 5 Compute the final integration. Once we have obtained $P_{Am}(t_i; y_j)$ in any state $y_j$ at time $t_i$ in the last step, we can find the American option value at time $t_0 = 0$ as follows.

$$P_{Am}(0) = \max\{K - S_0, P_c(0)\}, \quad (4.6)$$

where the reduced value at time zero is given by

$$P^c(0) = e^{-r\Delta t} \int_{-\infty}^{\infty} P_{Am}(t_1; y) \exp\left(-\frac{(y + \sigma \Delta t)^2}{2\Delta t}\right) dy$$

$$\approx e^{-r\Delta t} \sum_{k=2}^{M} \frac{\delta_k}{2} \left[\int P_{Am}(t_1; y_k) \exp\left(-\frac{(y_k + \sigma \Delta t)^2}{2\Delta t}\right) dy + P_{Am}(t_1; y_{k-1}) \exp\left(-\frac{(y_{k-1} + \sigma \Delta t)^2}{2\Delta t}\right)\right], \quad (4.7)$$

In fact, our method can be used for pricing European options as well. For the European put option, we just need to replace (4.2), (4.4) (4.6) by

$$P_{Eu}(t_{N-1}; y_j) = P^c(t_{N-1}; y_j), \quad (4.8)$$

$$P_{Eu}(t_i; y_j) = P^c(t_i; y_j), \quad (4.9)$$

$$P_{Eu}(0) = P^c(0), \quad (4.10)$$

respectively, and replace $P_{Am}$ by $P_{Eu}$ in (4.5) and (4.7).
4.1 Accuracy test for European options

European options are valued by a closed-form solution called BS formula (Black and Scholes, 1973). In order to show the accuracy of our proposed lattice algorithm, it is worthy to compare the result by the lattice algorithm with that from the BS formula. The criterion we use is the difference rate defined as

$$\text{Diff} = \left| \frac{P_{BS} - P_{lat}}{P_{BS}} \right| \times 100\%,$$

where $P_{BS}$ denotes the BS price which is regarded as the true value, and $P_{lat}$ is the option price valued by our proposed lattice algorithm.

To illustrate the convergence of the lattice algorithm, we focus on a European put option with strike price $K = 40$ and maturity date $T = 1$ on a share of stock with its initial price $S_0 = 36$ and volatility $\sigma = 0.2$. First we consider $N = 50$ as the number of determination dates. Figure 2 concludes that the rate of convergence (number of nodes $M$) slows down as the grid space $\phi$ is reduced. The converging prices by different grid spaces are quite close to the BS price 3.8443, indicating that the European option price converges when the lattice covers approximately ten standard deviations, i.e. $\phi \times (M - 1) \approx 10$. Table 1 gives a summarised result.

Figure 2  Convergence of a European put option using different grid spaces in the lattice algorithm (see online version for colours)

<table>
<thead>
<tr>
<th>Convergence (number of nodes)</th>
<th>Grid space</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 95</td>
<td>0.1</td>
<td>3.8447</td>
</tr>
<tr>
<td>&gt; 195</td>
<td>0.05</td>
<td>3.8446</td>
</tr>
<tr>
<td>&gt; 367</td>
<td>0.025</td>
<td>3.8441</td>
</tr>
</tbody>
</table>
Furthermore, we fix the number of nodes at $M = 501$ in order to guarantee the convergence of the lattice algorithm by each grid space, and then study the relationship between the European option price and the number of determination dates $N$, and the results are displayed in Figure 3. The converged European option price remains constant over the number of determination dates, ranging from 1 to 100, under the lattice with different grid spaces $\phi = 0.1, 0.05$ or 0.025. It makes sense because of the non-early exercise property of European options. Although it shows a trend that the lattice with a larger grid space brings a slightly larger European option price, all the converging prices by different grid spaces are quite close to the BS price 3.8443.

**Figure 3** Relationship between European put option price and number of determination dates using different grid spaces in the lattice algorithm (see online version for colours)

Considering the convergence rate, we use 151 nodes with 0.1 as the grid space in the lattice algorithm, and comparison results under various settings are summarised in Table 2. As shown, the differences between the lattice algorithm and BS formula in pricing European options are typically very small across all the 20 scenarios. Comparatively, the lattice algorithm performs better when options are in the money at time zero because of controlling the difference rate below 0.02%, while the difference rate increases for options that are at the money or out of the money although no higher than 0.06%. Overall, the lattice algorithm is able to approximate closely the BS price for European options.

**Table 2** Comparison in pricing European put options with strike price $K = 40$ between BS formula and lattice algorithm

<table>
<thead>
<tr>
<th>$S_0$</th>
<th>$\sigma$</th>
<th>$T$</th>
<th>$N$</th>
<th>$BS$</th>
<th>$Lattice$</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0.10</td>
<td>1</td>
<td>50</td>
<td>5.7454</td>
<td>5.7454</td>
<td>0.000%</td>
</tr>
<tr>
<td>32</td>
<td>0.10</td>
<td>2</td>
<td>100</td>
<td>4.1235</td>
<td>4.1237</td>
<td>0.005%</td>
</tr>
<tr>
<td>32</td>
<td>0.50</td>
<td>1</td>
<td>50</td>
<td>10.0666</td>
<td>10.0671</td>
<td>0.005%</td>
</tr>
<tr>
<td>32</td>
<td>0.50</td>
<td>2</td>
<td>100</td>
<td>11.1561</td>
<td>11.1536</td>
<td>0.02%</td>
</tr>
</tbody>
</table>
Table 2  Comparison in pricing European put options with strike price $K = 40$ between BS formula and lattice algorithm (continued)

<table>
<thead>
<tr>
<th>$S_0$</th>
<th>$\sigma$</th>
<th>$T$</th>
<th>$N$</th>
<th>BS</th>
<th>Lattice</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>0.20</td>
<td>1</td>
<td>50</td>
<td>3.8443</td>
<td>3.8447</td>
<td>0.01%</td>
</tr>
<tr>
<td>36</td>
<td>0.20</td>
<td>2</td>
<td>100</td>
<td>3.7630</td>
<td>3.7633</td>
<td>0.008%</td>
</tr>
<tr>
<td>36</td>
<td>0.60</td>
<td>1</td>
<td>50</td>
<td>9.5460</td>
<td>9.5467</td>
<td>0.007%</td>
</tr>
<tr>
<td>36</td>
<td>0.60</td>
<td>2</td>
<td>100</td>
<td>11.4847</td>
<td>11.4849</td>
<td>0.002%</td>
</tr>
<tr>
<td>40</td>
<td>0.30</td>
<td>1</td>
<td>50</td>
<td>3.5574</td>
<td>3.5593</td>
<td>0.05%</td>
</tr>
<tr>
<td>40</td>
<td>0.30</td>
<td>2</td>
<td>100</td>
<td>4.3261</td>
<td>4.3236</td>
<td>0.05%</td>
</tr>
<tr>
<td>40</td>
<td>0.70</td>
<td>1</td>
<td>50</td>
<td>9.5004</td>
<td>9.5036</td>
<td>0.03%</td>
</tr>
<tr>
<td>40</td>
<td>0.70</td>
<td>2</td>
<td>100</td>
<td>12.1523</td>
<td>12.1531</td>
<td>0.006%</td>
</tr>
<tr>
<td>44</td>
<td>0.40</td>
<td>1</td>
<td>50</td>
<td>3.7828</td>
<td>3.7810</td>
<td>0.05%</td>
</tr>
<tr>
<td>44</td>
<td>0.40</td>
<td>2</td>
<td>100</td>
<td>5.2020</td>
<td>5.2033</td>
<td>0.02%</td>
</tr>
<tr>
<td>44</td>
<td>0.80</td>
<td>1</td>
<td>50</td>
<td>9.7541</td>
<td>9.7476</td>
<td>0.06%</td>
</tr>
<tr>
<td>44</td>
<td>0.80</td>
<td>2</td>
<td>100</td>
<td>13.0367</td>
<td>13.0377</td>
<td>0.007%</td>
</tr>
<tr>
<td>48</td>
<td>0.50</td>
<td>1</td>
<td>50</td>
<td>4.2397</td>
<td>4.2419</td>
<td>0.05%</td>
</tr>
<tr>
<td>48</td>
<td>0.50</td>
<td>2</td>
<td>100</td>
<td>6.2583</td>
<td>6.2551</td>
<td>0.05%</td>
</tr>
<tr>
<td>48</td>
<td>0.90</td>
<td>1</td>
<td>50</td>
<td>10.2093</td>
<td>10.2085</td>
<td>0.008%</td>
</tr>
<tr>
<td>48</td>
<td>0.90</td>
<td>2</td>
<td>100</td>
<td>14.0657</td>
<td>14.0669</td>
<td>0.008%</td>
</tr>
</tbody>
</table>

4.2 Accuracy test for American options

In this section, we price American options by the proposed lattice algorithm, the FD method, and LSM approach. The efficiency of each method can be gauged in terms of running time. In order to measure the accuracy, we regard the FD method as the benchmark and compute the difference between the price valued by least squares Monte Carlo $P_{LSM}$ and that by FD method $P_{FD}$ through

$$\text{Diff}_{LSM} = \frac{|P_{FD} - P_{LSM}|}{P_{FD}} \times 100\%.$$  \hspace{1cm} (4.12)

Similarly, we can compare the price given by the lattice algorithm, denoted as $P_{lat}$, with $P_{FD}$ as follows.

$$\text{Diff}_{lat} = \frac{|P_{FD} - P_{lat}|}{P_{FD}} \times 100\%.$$  \hspace{1cm} (4.13)

In order to learn the convergence of the lattice algorithm, we focus on an American put option with strike price $K = 40$ and maturity date $T = 1$ on a share of stock with its initial price $S_0 = 36$ and volatility $\sigma = 0.2$. The FD method gives 4.4643 as the price of the option. With fixed $N = 50$, similar to pricing European options, the rate of convergence slows down as the grid space $\phi$ becomes smaller, shown in Figure 4. A summarised result can be found in Table 3, indicating that the converging prices by different grid spaces are quite close to the FD price. In other words, the American option price converges as soon as the lattice covers approximately five standard deviations, i.e. $\phi \times (M - 1) \approx 5$. 


Figure 4  Convergence result of an American put option using different grid spaces in the lattice algorithm (see online version for colours)

Table 3  Convergence result of an American put option using different grid spaces in the lattice algorithm

<table>
<thead>
<tr>
<th>Convergence (number of nodes)</th>
<th>Grid space</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 47</td>
<td>0.1</td>
<td>4.4774</td>
</tr>
<tr>
<td>&gt; 81</td>
<td>0.05</td>
<td>4.4779</td>
</tr>
<tr>
<td>&gt; 143</td>
<td>0.025</td>
<td>4.4776</td>
</tr>
</tbody>
</table>

Figure 5  Relationship between American put option price and number of determination dates using different grid spaces in the lattice algorithm (see online version for colours)
An efficient grid lattice algorithm for pricing American-style options

Furthermore, Figure 5 describes the relationship between the American option price and the number of determination dates $N$ with fixed number of nodes $M = 151$ to guarantee the convergence of the lattice algorithm by each grid space. First the option price increases with the number of determination dates, and afterwards it becomes stable when the number of determination dates reaches a certain point, around $N = 50$. As a result, during the lattice algorithm, assuming that the option is exercisable 50 times per year is sensible in practice.

Table 4  Comparison in pricing American put options with strike price $K = 40$ among FD method, LSM simulation, and lattice algorithm

<table>
<thead>
<tr>
<th>$S_0$</th>
<th>$\sigma$</th>
<th>$T$</th>
<th>$N$</th>
<th>FD Time$^1$</th>
<th>LSM Time$^1$</th>
<th>Diff$^2$</th>
<th>Lattice Time$^1$</th>
<th>Diff$^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0.10</td>
<td>1</td>
<td>50</td>
<td>8.0001</td>
<td>7.9530</td>
<td>0.58%</td>
<td>8</td>
<td>0.3124</td>
</tr>
<tr>
<td>32</td>
<td>0.10</td>
<td>2</td>
<td>100</td>
<td>8.0006</td>
<td>7.9540</td>
<td>0.58%</td>
<td>8</td>
<td>0.5807</td>
</tr>
<tr>
<td>32</td>
<td>0.50</td>
<td>1</td>
<td>50</td>
<td>10.5537</td>
<td>10.5703</td>
<td>0.16%</td>
<td>10.5817</td>
<td>0.2928</td>
</tr>
<tr>
<td>32</td>
<td>0.50</td>
<td>2</td>
<td>100</td>
<td>12.1212</td>
<td>12.2122</td>
<td>0.75%</td>
<td>12.1635</td>
<td>0.5807</td>
</tr>
<tr>
<td>36</td>
<td>0.20</td>
<td>1</td>
<td>50</td>
<td>4.4643</td>
<td>4.4781</td>
<td>0.31%</td>
<td>4.4776</td>
<td>0.2753</td>
</tr>
<tr>
<td>36</td>
<td>0.20</td>
<td>2</td>
<td>100</td>
<td>4.8287</td>
<td>4.8241</td>
<td>0.09%</td>
<td>4.8352</td>
<td>0.5681</td>
</tr>
<tr>
<td>36</td>
<td>0.40</td>
<td>1</td>
<td>50</td>
<td>7.0762</td>
<td>7.0601</td>
<td>0.23%</td>
<td>7.0103</td>
<td>0.2999</td>
</tr>
<tr>
<td>36</td>
<td>0.40</td>
<td>2</td>
<td>100</td>
<td>8.4808</td>
<td>8.5259</td>
<td>0.53%</td>
<td>8.4997</td>
<td>0.5786</td>
</tr>
<tr>
<td>40</td>
<td>0.30</td>
<td>1</td>
<td>50</td>
<td>3.7866</td>
<td>3.7988</td>
<td>0.32%</td>
<td>3.8062</td>
<td>0.3116</td>
</tr>
<tr>
<td>40</td>
<td>0.30</td>
<td>2</td>
<td>100</td>
<td>4.8721</td>
<td>4.8806</td>
<td>0.17%</td>
<td>4.8841</td>
<td>0.5845</td>
</tr>
<tr>
<td>40</td>
<td>0.70</td>
<td>1</td>
<td>50</td>
<td>9.5877</td>
<td>9.7834</td>
<td>2.0%</td>
<td>9.7753</td>
<td>0.3067</td>
</tr>
<tr>
<td>40</td>
<td>0.70</td>
<td>2</td>
<td>100</td>
<td>12.1516</td>
<td>12.8420</td>
<td>5.7%</td>
<td>12.7170</td>
<td>0.6028</td>
</tr>
<tr>
<td>44</td>
<td>0.40</td>
<td>1</td>
<td>50</td>
<td>3.9240</td>
<td>3.9149</td>
<td>0.23%</td>
<td>3.9476</td>
<td>0.2926</td>
</tr>
<tr>
<td>44</td>
<td>0.40</td>
<td>2</td>
<td>100</td>
<td>5.6151</td>
<td>5.6166</td>
<td>0.03%</td>
<td>5.5687</td>
<td>0.5847</td>
</tr>
<tr>
<td>44</td>
<td>0.80</td>
<td>1</td>
<td>50</td>
<td>9.6885</td>
<td>9.9573</td>
<td>2.8%</td>
<td>9.9876</td>
<td>0.2969</td>
</tr>
<tr>
<td>44</td>
<td>0.80</td>
<td>2</td>
<td>100</td>
<td>12.6217</td>
<td>13.7416</td>
<td>8.9%</td>
<td>13.3210</td>
<td>0.5798</td>
</tr>
<tr>
<td>48</td>
<td>0.50</td>
<td>1</td>
<td>50</td>
<td>4.3462</td>
<td>4.3691</td>
<td>0.53%</td>
<td>4.3701</td>
<td>0.2905</td>
</tr>
<tr>
<td>48</td>
<td>0.50</td>
<td>2</td>
<td>100</td>
<td>6.5939</td>
<td>6.6670</td>
<td>1.1%</td>
<td>6.4054</td>
<td>0.5925</td>
</tr>
<tr>
<td>48</td>
<td>0.90</td>
<td>1</td>
<td>50</td>
<td>9.9674</td>
<td>10.4593</td>
<td>4.9%</td>
<td>10.4104</td>
<td>0.2974</td>
</tr>
<tr>
<td>48</td>
<td>0.90</td>
<td>2</td>
<td>100</td>
<td>13.1822</td>
<td>14.7601</td>
<td>11.9%</td>
<td>13.9830</td>
<td>0.5802</td>
</tr>
</tbody>
</table>

Notes:  
1Run time is measured in the unit of second.  
2Differences are computed according to equation (4.12).  
3Differences are computed according to equation (4.13).

Considering the convergence, we use 151 nodes with 0.025 as the grid space in the lattice algorithm, and FD and LSM estimates are based on 100,000 simulation paths, corresponding to Longstaff and Schwartz (2001). Comparison results under various settings, including prices and computation time, are summarised in Table 4. The LSM approach and lattice algorithm both give close results to FD, which is the benchmark method, across all the scenarios except for those stocks with high volatility ($\sigma > 0.5$). However, in pricing American options on highly volatile stocks, not only is the lattice algorithm comparable to the LSM in terms of accuracy, the lattice algorithm is much more efficient with less time needed as well. In addition, when options are deeply in the money at time zero, the lattice algorithm outperforms LSM to a large extent based on
both accuracy and efficiency. In general, considering efficiency, our proposed lattice algorithm, with the least computation time, stands out among the three approaches. Consequently, in practice, while controlling the accuracy at an acceptable level, we prefer the lattice algorithm to LSM in terms of the high accuracy and distinguished computation time.

5 Concluding remarks

It is of practical importance to develop computationally feasible algorithms in pricing American-style options. In this paper, we present an efficient grid lattice approach and afterwards uses backward induction to price options. We build a time adjusted grid lattice that shifts as time changes. The lattice algorithm achieves high accuracy as well as fast computational speed. Comparison results between the lattice algorithm and current popular pricing models in the market, including BS formula in pricing European options and FD method as well as LSM approach in pricing American options, are shown in previous sections. All suggests that our lattice algorithm does a better job in terms of accuracy and efficiency. Finally, the fast convergence behaviours of the lattice algorithm as well as the relationship between the converged option price and the number of determination dates are illustrated as well.

As a framework for valuing American-style options, the grid lattice algorithm has several advantages. Besides the accuracy and fast convergence, the implementation of the model is simple and straightforward. In addition, the applicability of the lattice algorithm is much broader and more promising in markets with multiple factors by extending one factor lattice algorithm to multi-factor version, which can capture more complicated and realistic structures. For ease of illustration, we build the grid lattice model based on the trapezoidal rule integration. As mentioned in Xiao (2011), a better but more complicated solution is to spline the payoff functions because the cubic spline of the option payoffs is able to achieve higher accuracy, especially for Greeks calculations, and higher speed. In general, the spline method can provide a speedup factor around three to five times. The extension to multi-factor lattice models and the implementation of more numerical fast integration algorithms, are beyond the scope of the current paper and are interesting topics for future research.

References

An efficient grid lattice algorithm for pricing American-style options


An efficient grid lattice algorithm for pricing American-style options


