University of Toronto

CSC494 Project Report

An Empirical Evaluation of the Numerical Techiques on American Put Option Valuation

Author:
Jiwoong Im

Supervisor:
Kenneth Jackson

September 2, 2013

1 Introduction

In finance, options get traded world-wide and the popularity of options has been surging in demand since it has been introduced. The option is a financial instrument that leverages various of beneficial aspects to investors and traders. By possessing options, one has less risks than just holding equities, and perhaps, one maybe able to come up with various tactical alternatives. Indeed, trading options have higher chances of making revenues, and it can also be cost efficiency, which one can buy the stocks at a cheaper price by exercising the call option. Thus, profound understanding of options and sophisticated evaluation of options are almost compulsory.

The put option is a contract that is sold by one party (the option writer) to another party (the option holder). The contract has a prescribed *maturity date*, a prescribed *underlying asset*, and a prescribed *strike price*. The option holder has the right, but not the obligation, to sell the underlying asset to the option writer at the strike price during the certain period of its life time. The simplest option is an European put option, which the option can be only exercised at its maturity date. The American option is one style of the options that takes up a large portion of the option exchanges, and this particular option can be exercised any time up to and including the maturity date of the option.

Pricing an option using the explicit form or solving analytically is ideal. For example, the European option price can be solved using the Black-Scholes partial differential equation (PDE). However, unfortunately, most of the options must be evaluated by appling numerical techniques. Such problem of determining the fair market price of an American put option can be formulated mathematically as a free-boundary-value problem for a partial differential equation (PDE), and this problem can only be solved by numerical methods.

Another standard example of a free-boundary-value problem for a PDE is the melting ice problem. Given ice in water, we can determine the ice and water temperatures by solving the *heat equation* (diffusion equation). However, as the ice melts, the boundary between the ice and the water changes. Determining this free boundary is a key part in solving this problem. Similarly, to solve for the fair market price for an American put option, we must decide when is the optimal time to exercise the option. For example, the option holder might exercise the option to sell the stock at time t_1, but what if the price of the stock decreases further after time t_1? Therefore, the problem of determining the fair market price of an American put option includes determining the optimal time to exercise the option.

The PDE with free boundary problem can be formulated to a *continuous linear complementary problem* (LCP). This re-formulated problem can be discretized, and discretized LCP is commonly solved by projected successive over-relaxation (PSOR), and improved version of PSOR. The details of LCP is described in section 6. In our experiment, we tried to analyze two methods, and also tried to compare improved version of PSOR against the penalty method, which is another technique for pricing the American put option. The detail about the penalty method is described in section 7.

In our project, we assessed numerical techniques for the European option valutation problem and American option valuation problem. We employed finite-difference methods (FDM) to obtain numerical solutions to PDEs and LCPs, and most of numerical techniques that we examined require iterative approaches. We first tackled by assessing the performance on the European option valuation, and then evaluated the performance on the American option valuation. Numerical methods that we went over were explicit FD, fully implicit FD, Crank-Nicolson, and Rannacher-Smoothing. For pricing an European option, we extensively looked at LU factorization and succesive over-relaxation (SOR), and for the American option, we examined projected-SOR, the penalty method, and improved version of the projected SOR.

2 European Option Valuation

2.1 Constructing Black-Scholes Formula

Let $V(S,t)$ be the price of the option, where S is the current stock price and t is the current time. Note that $t \leq T$, where T is the maturity date. Also, let E be the strike price and $P(S) = max(E-S, 0)$ be the payoff value when the option is exercised.

Let Π be a portfolio, such that $\Pi = V - \eta S$. Then we have a portfolio consisting of a single option and n number of the underlying asset. Then value of the porfolio for one time step will be $d\Pi = dV - \eta dS$. The return of stock price can be constructed using stochastic differential equation (SDE),

$$\frac{dS}{S} = \sigma dX + \mu dt,$$

where σdX is the Wiener process, and μdt is the drift for a single time step. Substituting SDE into $d\Pi = dV - \eta dS$ and applying *Ito's Lemma* to dV term gives

$$d\Pi = \sigma S(\frac{\partial V}{\partial S} - \eta)dX + (\mu S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} - \mu\eta S)dt$$

Setting $\eta = \frac{\partial V}{\partial S}$, it cancels out the undeterministic part of the equation, and only the determinstic part of the equation is left.

$$d\Pi = (\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2})dt$$

Assuming that the transaction cost is free, one can expect the return of the portfolio can grow at $r\Pi dt$ in a time dt. Then substituting $r\Pi dt$ for $d\Pi$, we get

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0,$$

which is *Black Scholes partial differential equation* [1] . Notice that if $r\Pi dt$ is less than $d\Pi$, then we have an arbitrage, since we can borrow more money from the bank and invest more in the portfolio. If $r\Pi dt$ is greater, then we are short in porfolio, so it will be a loss for the investor. Hence, the Black Scholes equation must be equal to zero a the European option.

2.2 European Put Option Problem

The European option is an option that can be exercised only at the end of its maturity date. The payoff for the put option in the end of the maturity date will return $(E-S)_+$. This means that the price of the option at maturity date has to be equal to the payoff value. Moreover, as time gets near the expiry date, the price of option should converge to the payoff value. For the option valuation, we will assume that arbitrage does not exist, which means there is no opportunity to make free profit. Formulating all constraints above, we get

$$\begin{aligned}
&\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0,\\
&V(S,T) = max(E-S, 0),\\
&V(0,t) = 0\\
&V(S,t) = S, \text{ as } S \to \infty
\end{aligned}$$

[1] Scholes and Merton won the Nobel Prize in 1997 for developing their option pricing methodology based on this PDE. Unfortunately, Black died before 1997 and so could not share in the Nobel Prize.

The first equation is the PDE, the second constraint is the final condtion, and last two constraints are boundary conditions. This problem can be simplified to a diffusion equation by transforming V space into u space.

$$S = Ee^x,\ t = T - \tau/\frac{1}{2}\sigma^2,\ V = Ee^{\alpha x+\beta\tau}u(x,\tau),\ \alpha = \frac{-1}{2}(k-1),\ \beta = \frac{-1}{4}(k+1)^2,\ k = \frac{2r}{\sigma^2}$$

After applying change of variables and substitutions, the problem transforms to

$$\begin{aligned}
&\frac{\partial u}{\partial \tau} + \frac{\partial^2 u}{\partial x^2} = 0,\\
&u(x,0) = max(e^{\frac{1}{2}(k-1)x} - e^{\frac{1}{2}(k+1)x}, 0)\\
&u(0,\tau) = 0\\
&u(x,\tau) = x, \text{ as } x \to \infty
\end{aligned}$$

Notice that $\frac{\partial u}{\partial \tau} + \frac{\partial^2 u}{\partial x^2}$ is an operator associated with the well known *heat equation.* Moreover, t was moving forward in time until T, but τ is moving backword in time. This transform the final condition problem into the initial condition problem. Then the transformed formulation is

$$u(x,\tau) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} u_0(s)e^{-(x-s)^2/4\tau}ds,$$

which is a solution to the diffusion equations. And then, the closed form equation is obtained:

$$\begin{aligned}
P(S,t) &= Ee^{-r(T-t)}N(-d_2) - SN(-d_1)\\
N(d_1) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d_1} e^{-\frac{1}{2}s^2}ds\\
N(d) + N(-d) &= 1
\end{aligned}$$

3 Finite-Difference Methods to the Option Valuation Problem

Finite-difference methods are commonly used to obtain numerical approximations to option valuation problems. Let $\mathcal{L}$ be the differential operator, such that

$$\mathcal{L} = \frac{\partial}{\partial \tau} + \frac{\partial^2}{\partial x^2}.$$

A finite-difference method approximates u on a finite grid, $\{(n, m) \in [-N, N] \times [0, M]\}$, where $2N+1$ is the total number of points along the x-dimension and $M+1$ is the total number of points along the τ-dimension. Thus, $-\delta x N \leq x \leq \delta x N$ and $0 \leq \tau \leq \delta \tau M$. Observe that x is bounded in this problem, even though stock price, S, is in $[0, \infty)$. The truncation of domain along x was necessary, because we need to specify the finite domain to create the finite grid. This illustrates that choosing the upper bound and the lower bound of x can affect the accuracy. Typically the maximum is chosen between 3 to 5, which is approximately 271 dollars given a strike price of hundred dollars. In an ordinary case, this is more than enough to be our upper bound.

For the value of $u(x, \tau)$ at the mesh point, $u(n\delta x, m\delta\tau) \approx u_n^m$. Additionally, the initial condition and boundary conditions are represented as

$$\begin{aligned} u(x, 0) &= u(\ n\delta x, \ \ 0) = u_n^0 \ , \ n \in \mathbb{N} \ , \ -N \leq n \leq N \\ u(x_{min}, \tau) &= u(-N\delta x, \tau) = u_{-N}^m \ , \ m \in \mathbb{N} \ , \ 0 \leq m \leq M \\ u(x_{max}, \tau) &= u(\ N\delta x, \ \tau) = u_N^m \ , \ m \in \mathbb{N} \ , \ 0 \leq m \leq M \end{aligned}$$

Now, we have $(2N-1)M$ many u_n^m unknowns. In order to solve unknowns, u_n^m, we need to either solve it directly from some explicit formula or by constructing a linear system. There are several methods that can be applied to finite-difference approximations.

3.1 Explicit Finite-difference Method

The explicit finite-difference method uses the forward difference approximation for $\frac{\partial u}{\partial \tau}$ and the central difference approimation for $\frac{\partial^2 u}{\partial x^2}$ to compute u_n^m for all n and m. Using the forward difference with respect to the time implies that it computes u_n^m explicitly given u with previous time, $m-1$, for all n. Applying a Taylor expansion on each term in the diffusion equation, $\mathcal{L}u$, we get

$$\begin{aligned} \frac{\partial u}{\partial \tau} &= \frac{u(x, \tau + \delta\tau) - u(x, \tau)}{\delta\tau} + O(\delta\tau) \\ \frac{\partial^2 u}{\partial x^2} &= \frac{u(x, \tau + \delta\tau) - 2u(x, \tau) + u(x, \tau - \delta\tau)}{(\delta x)^2} + O((\delta x)^2) \end{aligned}$$

After restricting our domain to the finite grid, the approximation can be represented as

$$\frac{u_n^m - u_n^{m-1}}{\delta\tau} + O(\delta\tau) = \frac{u_{n+1}^{m-1} - 2u_n^{m-1} + u_{n-1}^{m-1}}{\delta x^2} + O((\delta x)^2)$$

We drop the $O(\cdot)$ truncation error terms and rearrange the equation above

$$\lambda u_{n-1}^m - (1 + 2\lambda) u_n^m + \lambda u_{n+1}^m = u_n^{m+1} \qquad (*)$$

where $\lambda = \frac{\delta\tau}{\delta x^2}$. Starting from $m = 0$ and until $m = M$, we can calculate u_n^{m+1} for all n, since we know u_n^m.

The major disadvantage of the explicit finite difference method is that it suffers from the stability problem. The choice of $\delta\tau$ and δx, directly affect the accuracy of solution. Depending on the ratio,

$\lambda = \frac{\delta\tau}{(\delta x)^2}$, we may face the computational precision problem while executing the method. The error arises on arthimetic precision, where rounding errors from the equation $(*)$ get magnified at each iteration. The method is only stable if $0 < \lambda \leq \frac{1}{2}$ and unstable if $\frac{1}{2} < \lambda$. The proof is shown on appendix A.1.

However, as we want to discretize finer along x and still maintain the accuracy, we must make the grid coarser along τ. For example, suppose λ is within a stable range and we decide to break the grid twice more along x, then the step size in time must be quartered in order to maintain the accuracy.

3.2 Fully Implicit Finite-difference Method

The fully implicit finite-difference method uses the backward difference approximation for $\frac{\partial u}{\partial \tau}$ and the central difference approximation for $\frac{\partial^2 u}{\partial x^2}$ to compute u_n^m for all n and m. The derivation of the implicit method is very similar to the explicit finite-difference method, but has intrinsically different properties and makes a big difference on stability.

$$\frac{\partial u}{\partial \tau} = \frac{u(x,\tau) - u(x,\tau-\delta\tau)}{\delta\tau} + O(\delta\tau)$$
$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x,\tau+\delta\tau) - 2u(x,\tau) + u(x,\tau-\delta\tau)}{(\delta x)^2} + O((\delta x)^2)$$

Similarly, restricting our domain to the finite grid, we get

$$\frac{u_n^{m+1} - u_n^m}{\delta\tau} + O(\delta\tau) = \frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{(\delta\tau)^2} + O(\delta x^2)$$

After dropping $O(\cdot)$ and the rearrangement, the equation becomes

$$-\lambda u_{n-1}^{m+1} + (1+2\lambda)u_n^{m+1} - \lambda u_{n+1}^{m+1} = u_n^m$$

In the implicit method, we are solving for u_n^{m+1} for all n given u_n^m. In contrast to the explicit method, we are implicitly solving u. Observe that we have a linear system, since there are $(2N-1)M$ many similar equations above and $(2N-1)M$ many unknowns. We can write the linear system as

$$\mathbf{M}\mathbf{u}^{m+1} = \mathbf{b}^m$$

Indeed, the matrix $\mathbf{M}$ turned out to be the symmetric tridiagonal matrix. This is because we are only using three-point stencil, where only u_n^{m+1}, u_{n+1}^{m+1}, and u_{n-1}^{m+1} depend on u_n^m. Moreover $\mathbf{b}^m$ is u_n^m plus boundary cases, where λu_{-N}^{m+1} and λu_N^{m+1}. Therefore, the linear system appears to be

$$\begin{pmatrix} 1+2\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & 1+2\lambda & -\lambda & & 0 \\ 0 & -\lambda & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\lambda \\ 0 & 0 & & -\lambda & 1+2\lambda \end{pmatrix} \begin{pmatrix} u_1^m \\ u_2^m \\ \vdots \\ u_9^m \\ u_{10}^m \end{pmatrix} = \begin{pmatrix} b_1^m \\ b_2^m \\ \vdots \\ b_9^m \\ b_{10}^m \end{pmatrix}$$

where $\mathbf{b}^m = \mathbf{u}^m + \lambda(u_{-N}^{m+1}, 0, 0, \ldots, u_N^{m+1})^T$

Although, the explicit finite-difference method might have an advantage on the runtime efficiency over the implicit method, the difference is very small. Indeed, with a small time step, implicit methods are slightly faster than the explicit finite-difference method. One general advantage of the implicit finite-difference method over the explicit finite-difference method is that it does not suffer from the stability problem. This illustrates that there is no constraint of λ being between $(0, \frac{1}{2}]$. Therefore, in contrast to the explicit method, it can maintain a finer grid along x without having coarse grid along τ.

3.3 Crank-Nicolson Method

Crank-Nicolson method is preferable to the explicit or implicit finite-difference method, because it takes advantage of the forward and backward difference by averaging the two methods for $\frac{\partial u}{\delta \tau}$. The central difference approximation is still used for $\frac{\partial^2 u}{\delta x^2}$.

$$\text{forward difference: } \frac{u_n^{m+1} - u_n^m}{\delta \tau} + O(\delta\tau) = \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\delta x)^2} + O((\delta x)^2)$$

$$\text{backward difference: } \frac{u_n^{m+1} - u_n^m}{\delta \tau} + O(\delta\tau) = \frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{(\delta x)^2} + O((\delta x)^2)$$

$$\text{average of two eq'ns: } \frac{u_n^{m+1} - u_n^m}{\delta \tau} + O(\delta\tau^2) = \frac{1}{2}\left(\frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{(\delta x)^2} + \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\delta x)^2}\right) + O((\delta x)^2)$$

Rearranging the equation above we get,

$$Z_n^m = (1-\lambda)u_n^m + \frac{\lambda}{2}(u_{n-1}^m + u_{n+1}^m)$$

$$(1+\lambda)u_n^{m+1} - \frac{\lambda}{2}(u_{n-1}^{m+1} + u_{n+1}^{m+1}) = Z_n^m \qquad (*)$$

Since it adopts the implicit method, it also needs to solve the system of linear equations,

$$\mathbf{C}\mathbf{u}^{m+1} = \mathbf{b}^m$$

Even though the method uses a six-point stencil, the matrix $\mathbf{C}$ still has the symmetric tridiagonal property. This is because, only three nodes are accounted for constructing the matrix, and the rest of three nodes are accumulated for $\mathbf{b}^m$. $\mathbf{b}^m$ is $\mathbf{Z}^m$ plus boundary cases, where $\frac{1}{2}\lambda(u_{-N}^{m+1}, 0, \cdots, 0, u_N^{m+1})^T$ is brought to the right hand side of equation $(*)$. Thus, the linear system is

$$\begin{pmatrix} 1+\lambda & -\frac{\lambda}{2} & 0 & \cdots & 0 \\ -\frac{\lambda}{2} & 1+\lambda & -\frac{\lambda}{2} & & 0 \\ 0 & -\frac{\lambda}{2} & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\frac{\lambda}{2} \\ 0 & 0 & & -\frac{\lambda}{2} & 1+\lambda \end{pmatrix} \begin{pmatrix} u_1^m \\ u_2^m \\ \vdots \\ u_9^m \\ u_{10}^m \end{pmatrix} = \begin{pmatrix} b_1^m \\ b_2^m \\ \vdots \\ b_9^m \\ b_{10}^m \end{pmatrix}$$

where $\mathbf{b}^m = \mathbf{Z}^m + \frac{\lambda}{2}(u_{-N}^{m+1}, 0, 0, \ldots, u_N^{m+1})^T$

The explicit finite-difference method and fully implicit finite-difference method have $O(\delta\tau) + O((\delta x)^2)$. The main benefit that we gain from Crank-Nicolson is the accuracy. The Crank-Nicolson method has an accuracy of $O((\delta\tau)^2) + O((\delta x)^2)$. The proof is shown in Appendix A.2. Moreover, the Crank-Nicolson method satifies the stability and convergence for all $\lambda > 0$.

3.4 Rannacher Smoothing Method

One of the problem that arises in practice when using the Crank-Nicolson method is that you do not exactly get $O(\delta x^2) + O(\delta\tau^2)$. This is because Crank-Nicolson method try to approximate the payoff function using Talyor expansion, that is not a smooth curve. But, noticing that the implicit finite-difference method is actually better at converging closer to the payoff function than Crank-Nicolson finite-difference method. Rannacher Smoothing method uses the advantage of both, the implicit and the Crank-Nicolson method, by running the implicit method for first few iterations, then switch back to the Crank-Nicolson method. This estimates much better than just using Crank-Nicolson in practice, and also achieves $O(\delta x^2) + O(\delta\tau^2)$

4 Solving Linear System

For solving system of linear equations from section 3, we attempted solving them by using LU factorization and Successive over-Relaxation (SOR) method.

4.1 LU factorization

In linear algebra, LU factorization is used to decompose a matrix into a lower triangular matrix and an upper triangular matrix,

$$\mathbf{M} = \mathbf{LU}$$

Let the linear system be $\mathbf{Mx} = \mathbf{LUx} = \mathbf{b}$. Then the linear system can be solved in two steps.

1. Let $\mathbf{y} = \mathbf{Ux}$ and solve for $\mathbf{Ly} = \mathbf{b}$.
2. Solve $\mathbf{y} = \mathbf{Ux}$.

Fortunately, the matrices that we deal with are all symmetric and tridiagonal matrices. Following by this property, a lower triangular matrix, $\mathbf{L}$, and an upper triangular matrix, $\mathbf{U}$, are also tridiagonal if matrix, $\mathbf{M}$, is tridiagonal matrix. Suppose that $\mathbf{M}$ is n x n matrix is

$$\begin{pmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & & 0 \\ 0 & a_{32} & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & a_{n-1n} \\ 0 & 0 & & a_{nn-1} & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & & 0 \\ 0 & l_{32} & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & l_{nn-1} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & a_{12} & 0 & \cdots & 0 \\ 0 & u_{22} & a_{23} & & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & & \ddots & u_{n-1n-1} & a_{n-1n} \\ 0 & \cdots & 0 & 0 & u_{nn} \end{pmatrix}$$

Then, each term in the lower triangular and upper triangular matrices can be determined, and we can observe that they are tridiagonal as well:

$$\begin{aligned} u_{11} &= a_{11} \\ u_{ii} &= a_{ii} - l_{i,i-1}a_{i-1,i} \ , \forall i = 2, \cdots, n \\ l_{ii-1} &= \frac{a_{i-1,i}}{u_{i-1,i-1}} \end{aligned}$$

This allow us to solve the linear system very efficiently, because solving in two steps described above are single operations each.

$$\begin{aligned} y_i &= b_i + l_{ii-1}y_{i-1} \ , \forall i = 2, \cdots, n \\ x_i &= \frac{y_i + u_{ii}x_{i-1}}{a_{ii+1}} \ , \forall i = 1, \cdots, n-1 \end{aligned}$$

Here is the LU matrix for the fully-implicit method:

$$\begin{pmatrix} 1+2\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & 1+2\lambda & -\lambda & & 0 \\ 0 & -\lambda & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\lambda \\ 0 & 0 & & -\lambda & 1+2\lambda \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{-\lambda}{y_{-N+1}} & 1 & 0 & & 0 \\ 0 & \frac{-\lambda}{y_{-N+2}} & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{-\lambda}{y_{N-2}} & 1 \end{pmatrix} \begin{pmatrix} y_{-N+1} & -\lambda & 0 & \cdots & 0 \\ 0 & y_{-N+2} & -\lambda & & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & & \ddots & y_{N-2} & -\lambda \\ 0 & \cdots & 0 & 0 & y_{N-1} \end{pmatrix}$$

such that

$$y_{-N+1} = 1 + 2\lambda$$
$$y_n = 1 + 2\lambda - \frac{\lambda^2}{y_{n-1}}$$

Here is the LU matrix for the Crank-Nicolson method:

$$\begin{pmatrix} 1+\lambda & -\frac{\lambda}{2} & 0 & \cdots & 0 \\ -\frac{\lambda}{2} & 1+\lambda & -\frac{\lambda}{2} & & 0 \\ 0 & -\frac{\lambda}{2} & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\frac{\lambda}{2} \\ 0 & 0 & & -\frac{\lambda}{2} & 1+\lambda \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{-\lambda}{y_{-N+1}} & 1 & 0 & & 0 \\ 0 & \frac{-\lambda}{y_{-N+2}} & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{-\lambda}{y_{N-2}} & 1 \end{pmatrix} \begin{pmatrix} y_{-N+1} & \frac{-\lambda}{2} & 0 & \cdots & 0 \\ 0 & y_{-N+2} & \frac{-\lambda}{2} & & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & & \ddots & y_{N-2} & \frac{-\lambda}{2} \\ 0 & \cdots & 0 & 0 & y_{N-1} \end{pmatrix}$$

such that

$$y_{-N+1} = 1 + \lambda$$
$$y_n = 1 + \lambda - \frac{\lambda^2}{4y_{n-1}}$$

4.2 Successive Over-Relaxation

Alternatively, the SOR method is an iterative method, which starts with an initial guess and improves the solution until it converges to the exact solution. Its advantage over direct calculation is that there is an elegant way for this method to adapt to the American put option pricing problem. The backward difference equation can be rearranged and written as

$$u_n^{m+1} = \frac{1}{1+2\lambda}(b_n^m + \lambda(u_{n-1}^{m+1} + u_{n+1}^{m+1})),$$

which is for the fully implicit method. The average of the backward and forward difference equations can be rearranged and written as

$$u_n^{m+1} = \frac{1}{1+\lambda}(b_n^m + \frac{\lambda}{2}(u_{n-1}^{m+1} + u_{n+1}^{m+1})),$$

which is for the Crank-Nicolson method. The procedure goes by first making an initial guess of the solution, $u_n^{m+1,0}$, we then iteratively calculate $u_n^{m+1,k+1}$ for all n until $u_n^{m+1,k+1}$ converges to u_n^{m+1} by repeating process below:

$$\text{Fully Implicit FDM: } y_n^{m+1,k+1} = \frac{1}{1+2\lambda}(b_n^m + \lambda(u_{n-1}^{m+1,k+1} + u_{n+1}^{m+1,k})),\ -N < n < N$$
$$\text{Crank-Nicolson FDM: } y_n^{m+1,k+1} = \frac{1}{1+\lambda}(b_n^m + \frac{\lambda}{2}(u_{n-1}^{m+1,k+1} + u_{n+1}^{m+1,k})),\ -N < n < N$$
$$u_n^{m+1,k+1} = u_n^{m+1,k} + \omega(y_n^{m+1,k+1} - u_n^{m+1,k})$$
$$||\mathbf{u}^{m+1,k+1} - \mathbf{u}^{m+1,k}||^2 < \epsilon$$

where ϵ is the tolerance and ω is parameter typically between 0 to 2. When $0 < \omega < 1$, then the method is called succesive under-relaxation, and when $1 < \omega < 2$, then the method is called successive over-relaxation.

The reason why this is called a successive over-relaxation is because we have a hyper parameter, ω, that controls how much change we want to make from the previous iteration step to the next. Most of times, we make changes more than how much we need to. Furthermore, the method is gauranteed to be converge, because $\varrho(\mathbf{G}) < 1$ where $\mathbf{G}$ is iteration matrix of SOR and ϱ is spectral radius.

5 European Put Option Valuation Experiments

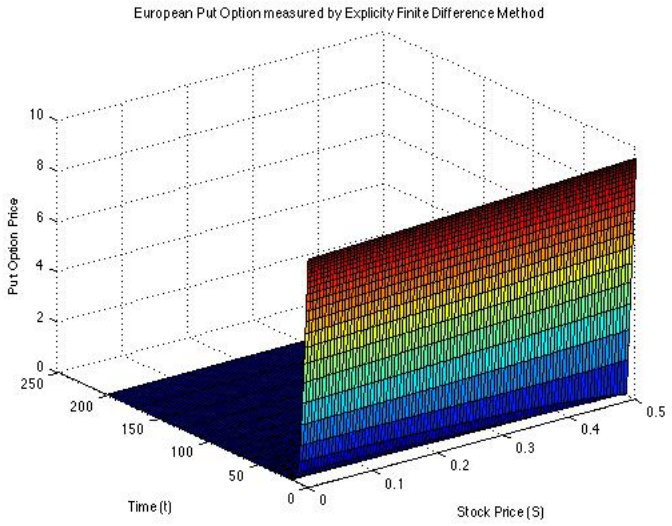

Figure 1: European Option Price Calculated by Explicit FDM over time

Three different finite-difference method techniques were explored to solve the European put option pricing problem; explicit, fully-implicit, and Crank-Nicolson methods. The implementation and performance experiment was done by evaluating each method with an exact put option price as a benchmark. The exact put option price was calculated by using the Black-Scholes formula.

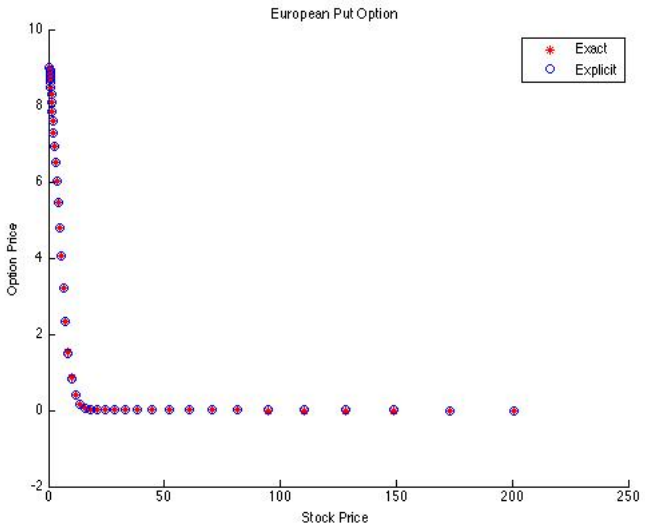

Figure 2: European Option Price Calculated by Explicit FDM at time $t = 0$

For the experiment, set the interest rate, $r = 0.05$, option exercise price, $K = 10$, volatility, $\sigma = 0.2$, and maturity of the option T to be 0.5 yr. several trials were performed by fixing the size of grid respect to τ and changing the size of the grid with respect to x, and visa versa.

Here is the chart for price of European put option evaluated by different methods at $t = 0$. The parameters for the chart below is $M = 100$, $N = 25$, $\delta\tau = 0.1$, $\delta x = 0.0004$, $\lambda = 0.04$, x is bounded by ± 3 :

Price	Exact (BS)	Explicit	Implicit-LU	Implicit-SOR	CN-LU	CN-SOR
0.4979	9.0144	9.0139	9.0139	9.0139	9.0139	9.0139
0.5613	8.9509	8.9506	8.9530	8.9505	8.9506	8.9506
0.6329	8.8794	8.8792	8.8844	8.8790	8.8792	8.8792
0.7136	8.7987	8.7986	8.8014	8.7983	8.7986	8.7986
0.8046	8.7077	8.7076	8.7090	8.7073	8.7076	8.7076
0.9072	8.6051	8.6050	8.6056	8.6047	8.6050	8.6050
1.0228	8.4895	8.4894	8.4896	8.4890	8.4894	8.4894
1.1533	8.3590	8.3590	8.3590	8.3586	8.3590	8.3589
1.3003	8.2120	8.2119	8.2119	8.2116	8.2119	8.2119
1.4661	8.0462	8.0461	8.0461	8.0458	8.0461	8.0461
1.6530	7.8593	7.8592	7.8592	7.8588	7.8592	7.8592
1.8637	7.6486	7.6484	7.6484	7.6481	7.6484	7.6484
2.1014	7.4109	7.4108	7.4107	7.4104	7.4108	7.4108
2.3693	7.1430	7.1429	7.1428	7.1425	7.1428	7.1428
2.6714	6.8409	6.8408	6.8407	6.8404	6.8407	6.8407
3.0119	6.5004	6.5002	6.5001	6.4999	6.5002	6.5001
3.3960	6.1164	6.1162	6.1162	6.1160	6.1162	6.1162
3.8289	5.6837	5.6836	5.6837	5.6834	5.6836	5.6836
4.3171	5.1966	5.1969	5.1971	5.1969	5.1970	5.1970
4.8675	4.6505	4.6513	4.6519	4.6517	4.6516	4.6516
5.4881	4.0441	4.0453	4.0464	4.0462	4.0458	4.0458
6.1878	3.3847	3.3849	3.3865	3.3864	3.3857	3.3857
6.9768	2.6939	2.6901	2.6916	2.6915	2.6908	2.6908
7.8663	2.0106	1.9995	1.9998	1.9997	1.9997	1.9997
8.8692	1.3864	1.3673	1.3658	1.3657	1.3665	1.3665
10.0000	0.8703	0.8473	0.8446	0.8445	0.8459	0.8459
11.2750	0.4906	0.4703	0.4681	0.4681	0.4692	0.4692
12.7125	0.2453	0.2324	0.2317	0.2317	0.2321	0.2321
14.3333	0.1077	0.1021	0.1025	0.1025	0.1023	0.1023
16.1607	0.0412	0.0399	0.0407	0.0407	0.0403	0.0403
18.2212	0.0136	0.0139	0.0146	0.0146	0.0143	0.0143
20.5443	0.0039	0.0044	0.0048	0.0048	0.0046	0.0046
23.1637	0.0009	0.0012	0.0014	0.0014	0.0013	0.0013
26.1170	0.0002	0.0003	0.0004	0.0004	0.0004	0.0004
29.4468	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001
33.2012	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 1: Price of European Put Option evaluated by different methods at $t = 0$

From the Table 1, we can observe that most of methods produce reasonably accurate values compared to exact values that were computed by solving the Black-Scholes equation. However, the largest errors are generally emanate from near the stock price at 10, which is the strike price. The reason behind this is because the payoff function is a piecewise linear function that has slanted "L" shape. Therefore, the payoff function is not smooth, nor is the derivative at the strike price. And yet, the finite-difference methods use the Taylor series to approximate the payoff function, and requires the function to be analytic or smooth unto a certain degree . This causes the biggest error at the strike price. The values from Table 1, is measured with when $M = 100$ and $N = 40$. However, as we discretize the grid to be even finer, which

means increasing the size of N, our finite-difference method becomes identical to the actual option curve.

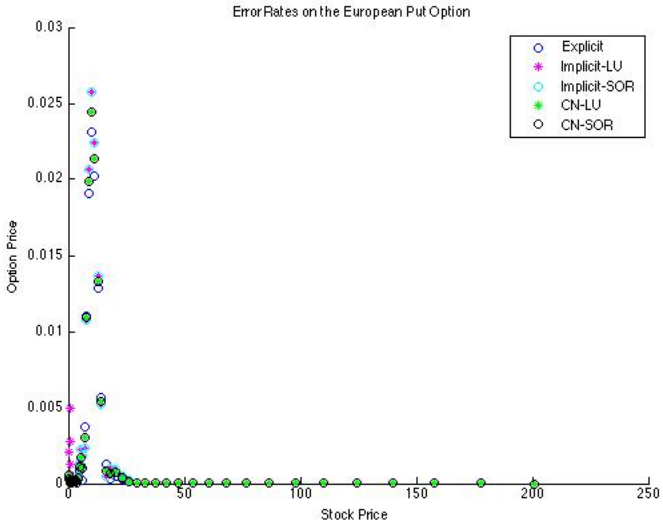

Figure 3: Example of Error Rate Plotted

The hypothesis for the explicit and implicit method are that the error will change quadratically with respect to δx and linearly respect to $\delta\tau$, since the error bound is $O(\delta x^2) + O(\delta\tau)$. Similarly, error bound for the Crank-Nicolson method is $O(\delta x^2) + O(\delta\tau^2)$, so the error will change quadratically with respect to δx and $\delta\tau$.

Table 2 displays maximum error values of the European put option evaluated by different numerical methods at $t = 0$ with all conditions maintained from above, except the error values were demonstrated with greater variety of grid sizes.

Table 2 illustrates the error behaviours on option price computed by different methods. The size of M and N were chosen from 10 to 700. For the purpose of asserting $O(\delta x^2) + O(\delta\tau)$ and $O(\delta x^2) + O(\delta\tau^2)$ of each method, consider observing a case when $\delta\tau$ is relatively small compare to δx, and vice versa. This way the influence of $\delta\tau$ on the error would not have as much effect.

Figure 4 demonstrates the error rate when $M = 100$ and N between 20 to 40. According to the graph, all methods have a linear growth in error rates as δx^2 grow linearly. Moreover, Figure 5 demonstrates the error rate when $N = 700$ and $M = 10$ to 40. The graph indicates that error rates for the implicit method and Crank-Nicolson method grows linearly, and Rannacher-Smoothing grows quadratically. These results illustrate that the errors follow our hypothesis.

M	N	λ	$\delta\tau$	δx^2	Explicit	Implicit-LU	Implicit-SOR	CN-LU	CN-SOR	RS
10	25	0.27	0.004	0.0144	0.0110	0.0380	0.0380	0.0241	0.0241	0.0279
10	30	0.40	0.004	0.01	0.0038	0.0298	0.0298	0.0164	0.0164	0.0201
10	35	0.54	0.004	0.00734	0.1367	0.0251	0.0251	0.0119	0.0119	0.0155
10	40	0.71	0.004	0.00562	11.7710	0.0220	0.0220	0.0090	0.0090	0.0126
100	20	0.01	0.0004	0.0225	0.0378	0.0407	0.0407	0.0393	0.0393	0.0393
100	25	0.02	0.0004	0.0144	0.0231	0.0258	0.0258	0.0244	0.0244	0.0245
100	30	0.04	0.0004	0.01	0.0154	0.0181	0.0181	0.0167	0.0167	0.0168
100	35	0.05	0.0004	0.00734	0.0109	0.0135	0.0135	0.0122	0.0122	0.0123
100	40	0.07	0.0004	0.00562	0.0080	0.0106	0.0106	0.0093	0.0093	0.0093
10	200	4.44	0.004	0.0009	∞	0.0157	0.0157	0.0087	0.0087	0.0049
20	200	2.22	0.002	0.0009	∞	0.0084	0.0084	0.0011	0.0011	0.0021
30	200	1.48	0.0013	0.0009	∞	0.0060	0.0060	0.0011	0.0011	0.0016
40	200	1.11	0.001	0.0009	∞	0.0048	0.0048	0.0011	0.0011	0.0014
10	700	54.4	0.004	0.000073	∞	0.0146	0.0146	0.0290	0.0290	0.0039
20	700	27.2	0.002	0.000073	∞	0.0074	0.0074	0.0104	0.0104	0.0011
30	700	18.1	0.0013	0.000073	∞	0.0049	0.0050	0.0039	0.0039	0.0006
40	700	13.6	0.001	0.000073	∞	0.0037	0.0038	0.0014	0.0014	0.0004

Table 2: Maximum Error Values of a European Put Option evaluated by different methods at $t = 0$

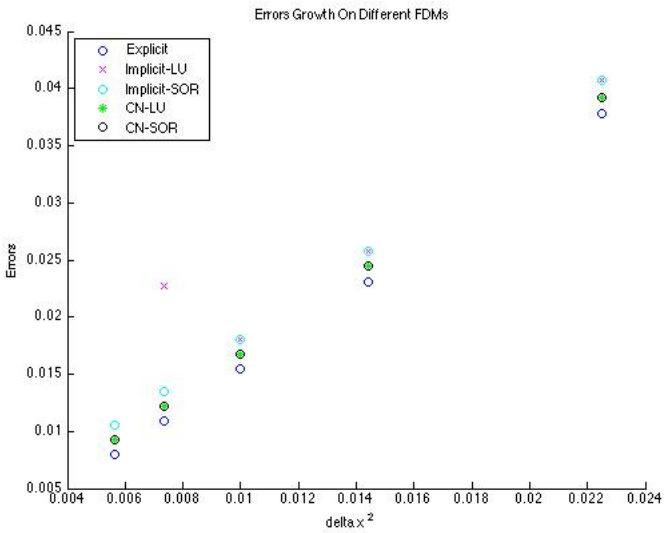

Figure 4: Error Rate from Explicit Method on fixed $M = 100$ and N from 20 to 40

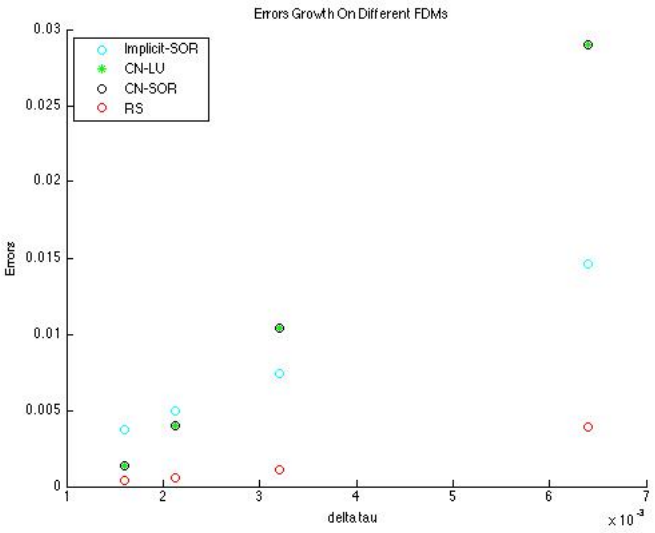

Figure 5: Error Rate from Explicit Method on fixed $N = 700$ and M from 10 to 40

6 American Option Valuation

The European Option Valuation only allowed exercising at its maturity date. However, because the American Option concedes exericising any time before or at its matuirty date, we have extra constraints while solving the PDE. More precisely, determining this free boundary, $x = x_f(t)$, is a key part in solving this problem, which is the border line between the exercising is an optimality or not. Similarly, to solve for the fair market price for an American put option, we must decide when is the optimal time to exercise the option.

6.1 Linear Complementary Problem

Let $V(S,t)$ be the price of the option, where S is the current stock price and t is the current time. Note $t \leq T$, where T is the maturity date. Also, let E be the strike price and $P(S) = \max(E - S, 0)$ be the payoff value when the option is exercised.

Next, let $S_f(t)$ be the yet to be determined position of the free-boundary at time t. It can be shown that, for $S \leq S_f(t)$ the option is exercised and $V(S,t) = P(S)$. On the other hand, for $S > S_f(t)$, the option is held and its value is determined as described below.

Furthermore, let $\mathcal{L}$ be the linear differential operator given by

$$\mathcal{L}V(S,t) = \frac{\partial V(S,t)}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,t)}{\partial S^2} + rS\frac{\partial V(S,t)}{\partial S} - rV(S,t)$$

$\mathcal{L}V(S,t) = 0$ is the famous *Black-Scholes* (or *Black-Scholes-Merton*) PDE. Finally, let the terminal condition and boundary conditions be

$$\begin{aligned} &V(S,T) = P(S) = \max(E - S, 0), \\ &V(0,t) = E, \qquad V(S,t) \to 0 \text{ as } S \to \infty \\ &V \text{ and } \frac{\partial V}{\partial S} \text{ are continuous} \end{aligned}$$

As noted above, we exercise the option if the stock price, S, satisfies $0 \leq S \leq S_f(t)$. In this case, it can be shown that

$$V = E - S \geq 0, \qquad \mathcal{L}V > 0$$

and, when early exercising is not optimal (i.e., $S_f(t) < S < \infty$), it can be shown that

$$V > E - S, \qquad \mathcal{L}\,V = 0$$

where S is the current asset value, t is the current time, V is the price of the American put option, and E is the strike price. Thus, we see that the problem takes the form of a *continuous linear complimentary problem*:

$$\begin{aligned} &\mathcal{L}\,V \geq 0\ , \\ &V - P \geq 0 \\ &\mathcal{L}\,V \cdot (V - P) = 0 \end{aligned}$$

where the last equation above is a short way of saying that, for every S and t, either $\mathcal{L}\,V(S,t) = 0$ or $V(S,t) - P(S) = 0$.

A major advantage of the linear complimentary formulation of the problem is that we do not need to determine the free boundary $S_f(t)$ explicitly. Once we solve the linear complimentary problem, the free boundary, $S_f(t)$, can be determined as the largest value of S for which $V(S,t) - P(S) = 0$.

To make the problem simpler to solve, we can transform V space to u space by changing variables from (S,t) to (x,τ):

$$S = Ee^x,\ t = T - \tau/\frac{1}{2}\sigma^2,\ V = Ee^{\alpha x+\beta\tau}u(x,\tau),\ \alpha = \frac{-1}{2}(k-1),\ \beta = \frac{-1}{4}(k+1)^2,\ k = \frac{2r}{\sigma^2}$$

Having made this change of variables, the differential operator $\mathcal{L}$ becomes

$$\mathcal{L}\,u(x,\tau) = \frac{\partial u(x,\tau)}{\partial \tau} - \frac{\partial^2 u(x,\tau)}{\partial x^2}$$

This is the operator associated with the well-known *heat equation.*

The initial and fixed boundary conditions become

$$\begin{gathered} u(x,0) = g(x,0), \\ u(x,\tau) \text{ is continuous,} \\ \frac{\partial u}{\partial x}(x,\tau) \text{is as continuous as } g(x,\tau), \\ \lim_{x\to\pm\infty} u(x,\tau) = \lim_{x\to\pm\infty} g(x,\tau) \end{gathered}$$

where $g(x,\tau)$ is payoff function throughout the option's lifetime.

6.2 Finite-Difference Formulation Setup

First, the payoff function, $g(x,\tau)$, and the option price function, $u(x,\tau)$ must be dicretized. Let $g(x,\tau) = g(n\delta x, m\delta\tau)$ and denote it as g_n^m. Also, let $\mathbf{u}^m$ be a vector that contains u_i for all $i = -N, \cdots, N$ at time m, and $\mathbf{g}^m$ be a vector that contains g_i for all $i = -N, \cdots, N$ at time m.
Then we can write $V - P \geq 0$ as

$$\mathbf{u}^m \geq \mathbf{g}^m$$

Moreover, we can formulate discretized versions of two equations $\mathcal{L}V$ and $V - P$ in a matrix form. Let $\mathbf{M}$ be the matrix that represents the left side of the linear system, and $\mathbf{b}^m$ be the right hand side of the linear system. Then our linear complimentary problem becomes

$$\begin{gathered} \mathbf{M}\mathbf{u}^{m+1} \geq \mathbf{b}^m, \\ \mathbf{u}^{m+1} \geq \mathbf{g}^m, \\ (\mathbf{u}^{m+1} - \mathbf{g}^{m+1}) \cdot (\mathbf{M}\mathbf{u}^{m+1} - \mathbf{b}^m) = 0 \end{gathered}$$

in terms of the matrix form.

Since the equality in the *heat equation* is converted to inequality, we can do the same for the implicit and the Crank-Nicolson method.

$$\text{Implicit: } (1+2\lambda)u_n^{m+1} - \lambda(u_{n-1}^{m+1} + u_{n+1}^{m+1}) \geq u_n^m$$

$$\text{Crank-Nicolson: } Z_n^m = (1-\lambda)u_n^m + \frac{\lambda}{2}(u_{n-1}^m + u_{n+1}^m)$$
$$(1+\lambda)u_n^{m+1} - \frac{\lambda}{2}(u_{n-1}^{m+1} + u_{n+1}^{m+1}) \geq Z_n^m$$

The $\mathbf{b}^m$ and $\mathbf{M}$ remains the same as the european option valuation case for the implicit and for the Crank-Nicolson method. For the implicit finite-difference method:

$$\mathbf{b}^m = \begin{pmatrix} u_{-N+1}^m \\ u_{-N+2}^m \\ \vdots \\ u_{N-2}^m \\ u_{N-1}^m \end{pmatrix} + \begin{pmatrix} g_{-N}^{m+1} \\ 0 \\ \vdots \\ 0 \\ g_N^{m+1} \end{pmatrix}, \text{ and } \mathbf{M} = \begin{pmatrix} 1+2\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & 1+2\lambda & -\lambda & & 0 \\ 0 & -\lambda & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\lambda \\ 0 & 0 & & -\lambda & 1+2\lambda \end{pmatrix}$$

For the Crank-Nicolson finite-difference method:

$$\mathbf{b}^m = \begin{pmatrix} Z_{-N+1}^m \\ Z_{-N+2}^m \\ \vdots \\ Z_{N-2}^m \\ Z_{N-1}^m \end{pmatrix} + \begin{pmatrix} g_{-N}^{m+1} \\ 0 \\ \vdots \\ 0 \\ g_N^{m+1} \end{pmatrix}, \text{ and } \mathbf{C} = \begin{pmatrix} 1+\lambda & -\frac{\lambda}{2} & 0 & \cdots & 0 \\ -\frac{\lambda}{2} & 1+\lambda & -\frac{\lambda}{2} & & 0 \\ 0 & -\frac{\lambda}{2} & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\frac{\lambda}{2} \\ 0 & 0 & & -\frac{\lambda}{2} & 1+\lambda \end{pmatrix}$$

7 Iterative Methods for the American Option Valuation

The notation and the grid setup from the European option valuation problem will be used again in the American option valuation problem. Some methods for the American option valuation builds up from solving the European option valuation.

7.1 The Projected Successive Over-Relaxation

The projected successive over-relaxation (PSOR) extended from SOR method. The SOR method is an iterative method, which also starts by an initial guess and improves the solution until it converges to the exact solution.

$$\text{Fully Implicit FDM: } y_n^{m+1,k+1} = \frac{1}{1+2\lambda}(b_n^m + \lambda(u_{n-1}^{m+1,k+1} + u_{n+1}^{m+1,k})),\ -N < n < N$$

$$\text{Crank-Nicolson FDM: } y_n^{m+1,k+1} = \frac{1}{1+\lambda}(b_n^m + \frac{\lambda}{2}(u_{n-1}^{m+1,k+1} + u_{n+1}^{m+1,k})),\ -N < n < N$$

$$u_n^{m+1,k+1} = u_n^{m+1,k} + \omega(y_n^{m+1,k+1} - u_n^{m+1,k})$$

$$||\mathbf{u}^{m+1,k+1} - \mathbf{u}^{m+1,k}||^2 < \epsilon$$

For the American option, we know that V must be greater than or equal to P, $u_n^m \geq g_n^m$. Then, at the end of each iteration, we can take

$$\max(\mathbf{u}^{m+1,k} + \omega(\mathbf{y}^{m+1,k+1} - \mathbf{u}^{m+1,k}), \mathbf{g}^m)$$

We can observe that this does not violate our linear complimentary constraints.
Case1:

$$\text{If } \mathbf{u}^{m+1,k} + \omega(\mathbf{y}^{m+1,k+1} - \mathbf{u}^{m+1,k}) \leq \mathbf{g}^m,$$
$$\text{then } \mathbf{u}^{m+1,k+1} = \mathbf{g}^m \text{ and } \mathbf{M}\mathbf{u}^{m+1,k+1} > \mathbf{b}^m$$

As consequence, we get

$$(\mathbf{u}^{m+1,k+1} - \mathbf{g}^{m+1}) \cdot (\mathbf{M}\mathbf{u}^{m+1,k+1} - \mathbf{b}^m) = 0$$

The reason for the inequality on the linear system is because, $\mathbf{u}^{m+1,k+1} = \mathbf{g}^m$ implies that it is optimal to exericse the option at time, $m+1$.
Case2:

$$\text{If } \mathbf{u}^{m+1,k} + \omega(\mathbf{y}^{m+1,k+1} - \mathbf{u}^{m+1,k}) > \mathbf{g}^m,$$
$$\text{then } \mathbf{M}\mathbf{u}^{m+1,k+1} = \mathbf{b}^m$$

This is because, $\mathbf{u}^{m+1,k+1} > \mathbf{g}^m$ implies that exercising is not the optimal choice. Hence, it satisfies the Black-Scholes formula. Therefore, we still maintain

$$(\mathbf{u}^{m+1,k+1} - \mathbf{g}^{m+1}) \cdot (\mathbf{M}\mathbf{u}^{m+1,k+1} - \mathbf{b}^m) = 0$$

The fact that $\mathbf{M}$ being the symmetric positive definite matrix guarantees that $\mathbf{M}\mathbf{u}^{m+1} \geq \mathbf{b}^m$. Furthermore, the optimal relaxation parameter, ω, can be determined when $\mathbf{M}$ is the symmetric positive definite matrix.

7.2 Penalty Method

The penalty method is quite different from the PSOR method, where PSOR explicitly tries to satisfy the LCP constraints on each iteration step of the method, the penalty method implicitly pleases the constraints by constantly penalizing the PDE if constraints are not satisfied. The penalty method uses

the non-smoothing Newton iteration, which is a simple application of Newton's equation for solving the equations.

In the penalty method, the penalty term, ϵ, is added to $\mathcal{L}u = 0$ for penalizing whenever $V < P$. This ensures the early exercise constraints would be satisfied. This penalty term is prominent to the option price curve because the option curve, V, tends to be lower than the payoff curve, P, on the left side of strike price. This is the natural behaviour for the European price as shown in the Figure 6.

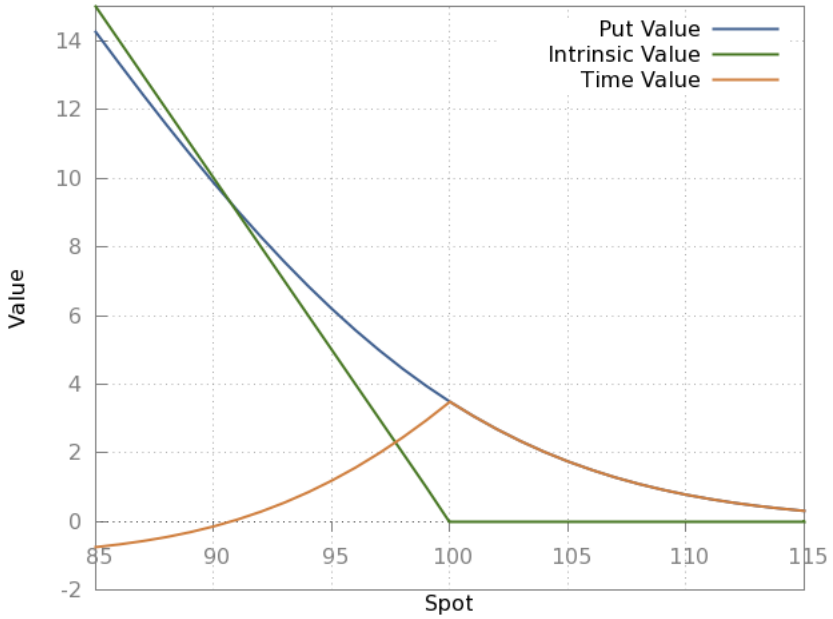

Figure 6: European Put Price Curve

Among various possible choices of penality formulas, one of the penalty term that I used is,

$$\epsilon = L\frac{V - P}{\delta\tau}$$

where L is the *penalty factor*, that is chosen so that $L \simeq O(\frac{1}{tol})$, and *tol* is the convergence tolerance. Then the penalty term for each point in the grid can be defined by:

$$p_n^{m+1,k} = \begin{cases} \frac{u_n^{m+1,k} - g_n^m}{\delta\tau \cdot L}, & \text{if } \mathbf{u}^{m+1,k} < \mathbf{g}^m \\ 0, & \text{if } \mathbf{u}^{m+1,k} \geq \mathbf{g}^m \end{cases}$$

This makes the our PDE, $\mathcal{L}u - \epsilon > 0$ to be a non-linear PDE, and this non-linear equation is solved by Newton's iterations. Also, by formulating the Black-Scholes equation in following manner, we can incorporate the explicit, implicit and Crank-Nicolson methods.

$$\frac{u_n^{m+1} - u_n^m}{\delta\tau} = (1-\theta)\mathcal{L}u_n^{m+1} + \theta\mathcal{L}u_n^m + p_n^{m+1}$$

where θ is a parameter that determines the method that will be applied.

On each Newton iteration, the penalty will be enforced to the $\mathcal{L}u$ until the error, *err* converges within the tolerance, *tol* or the penality applied in $m+1$ iterations is exactly same as the the penalty applied in

m. The error quantity is determined by

$$\max_{m,n} \frac{\max(|\mathbf{u}^{m+1,k+1} - \mathbf{u}^{m+1,k}|)}{\max(1, |\mathbf{u}^{m+1,k+1}|)}$$

In practice, the penality method is very appealing because the iteration is converged within one to three iterations.

7.3 The Improved PSOR

Last but not least, the improved PSOR is a clever technique that uses the knowledge of the estimated free boundary value. In practice, the PSOR algorithm is not used much because it takes such a long iterations to converge, especially for the fine grid. The idea of the improved PSOR algorithm is to run PSOR for just a few iterations, compute the exact value of Black-Scholes formula for the right side of the free boundary, and then re-iterate PSOR.

The early exercise is not optimal on the right side of the free boundary, $x_f(t) \leq x < x_{max}$, we can directly compute the Black-Scholes formula along that region. However, since we only iterated PSOR a few times, the free boundary, $x_f(t)$, might not be exactly accurate. Hence, we have to re-iterate PSOR to find the accurate solution. Indeed, the free boundary does get approximately correct within few grid steps. Thus, computing PSOR for the second time does not take many iterations. Moreoever, the fact that we approximated all values on the right side of the free boundary space with the exact Black-Scholes formula reduces a lot of computations.

Improved PSOR has three phases in its algorithm:

1. Perform Projected SOR
2. Reduce-Space
3. Re-perform Projected SOR

On the third phase, when PSOR is re-performed, it should run until the free boundary gets settled, or the option value at the $k+1$ iteration step, $\mathbf{u}^{m+1,k+1}$ is the same as $\mathbf{u}^{m+1,k}$. These two conditions illustrates that the option price has been resolved.

8 American Option Valuation Experiments

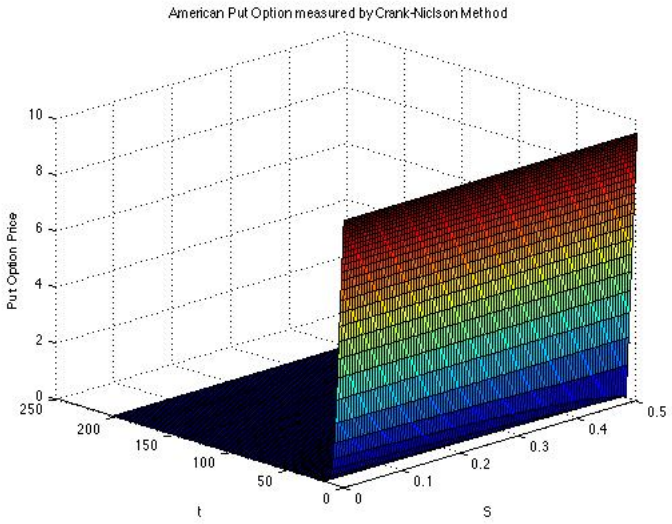

Figure 7: American Put Price Curve

In our American option valuation experiments, we explored the PSOR method, the penalty method, and the improved PSOR method. The implementation correctness and performance experiments were evaluated by error comparisons and iteration step comparisons. All the comparisons were performed on a 2.4 GHz Intel Core with 4 GB RAM, using Mac OS X.

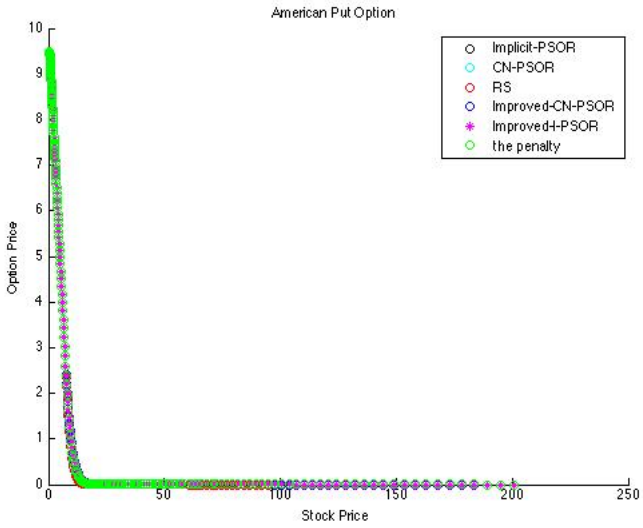

Figure 8: American Put Price Curve

For the benchmark of experiment, we used the same condition again as the European option price, where we set the interest rate, $r = 0.1$, option exercise price, $K = 10$, volatility, $\sigma = 0.4$, and maturity of the option T to be 0.5 yr. We did several trials by fixing the size of the grid with respect to τ and

changing the size of the grid with respect to x, and vice versa.

Here is the chart for the prices of an American option evaluated by different methods at $t = 0$. The parameters for the chart below are $M = 100$, $N = 25$, $\delta\tau = 0.1$, $\delta x = 0.0004$, $\lambda = 0.04$, x is bounded by ± 3 :

Price	Implicit-PSOR	CN-PSOR	RS-PSOR	Improved-CN-PSOR	Improved-I-PSOR
0.4979	9.5021	9.5021	9.5021	9.5021	9.5021
0.5613	9.4387	9.4387	9.4387	9.4387	9.4387
0.6329	9.3671	9.3671	9.3671	9.3671	9.3671
0.7136	9.2864	9.2864	9.2864	9.2864	9.2864
0.8046	9.1954	9.1954	9.1954	9.1954	9.1954
0.9072	9.0928	9.0928	9.0928	9.0928	9.0928
1.0228	8.9772	8.9772	8.9772	8.9772	8.9772
1.1533	8.8467	8.8467	8.8467	8.8467	8.8467
1.3003	8.6997	8.6997	8.6997	8.6997	8.6997
1.4661	8.5339	8.5339	8.5339	8.5339	8.5339
1.6530	8.3470	8.3470	8.3470	8.3470	8.3470
1.8637	8.1363	8.1363	8.1363	8.1363	8.1363
2.1014	7.8986	7.8986	7.8986	7.8986	7.8986
2.3693	7.6307	7.6307	7.6307	7.6307	7.6307
2.6714	7.3286	7.3286	7.3286	7.3286	7.3286
3.0119	6.9881	6.9881	6.9881	6.9881	6.9881
3.3960	6.6040	6.6040	6.6040	6.6040	6.6040
3.8289	6.1711	6.1711	6.1711	6.1711	6.1711
4.3171	5.6829	5.6829	5.6829	5.6829	5.6829
4.8675	5.1325	5.1325	5.1325	5.1325	5.1325
5.4881	4.5119	4.5119	4.5119	4.5119	4.5119
6.1878	3.8122	3.8122	3.8122	3.8122	3.8122
6.9768	3.0232	3.0232	3.0232	3.0232	3.0232
7.8663	2.1866	2.1861	2.1865	2.1866	2.1861
8.8692	1.4655	1.4642	1.4654	1.4655	1.4642
10.0000	0.8946	0.8930	0.8945	0.8946	0.8930
11.2750	0.4913	0.4900	0.4912	0.4913	0.4900
12.7125	0.2411	0.2407	0.2411	0.2411	0.2407
14.3333	0.1057	0.1059	0.1056	0.1057	0.1059
16.1607	0.0415	0.0419	0.0415	0.0415	0.0419
18.2212	0.0146	0.0150	0.0146	0.0146	0.0150
20.5443	0.0047	0.0049	0.0047	0.0047	0.0049
23.1637	0.0014	0.0015	0.0014	0.0014	0.0015
26.1170	0.0004	0.0004	0.0004	0.0004	0.0004
29.4468	0.0001	0.0001	0.0001	0.0001	0.0001
33.2012	0.0000	0.0000	0.0000	0.0000	0.0000

Table 3: Price of an American Put Option evaluated by different methods at $t = 0$

Evaluating the correctness of an American put option is more indirect because we do not have the benchmark values to compare from, like we do in the European option valuation. However, we know the error rate of each method. There is a trick that we can employ to verify the correctness of the implementation. For each method, the error rate is either $O(\delta\tau) + O(\delta x^2)$ or $O(\delta\tau^2) + O(\delta x^2)$. Consider fixing the grid along M, so that $O(\delta\tau)$ or $O(\delta\tau^2)$ is small, and consider the grid along x with $N = n, \frac{n}{2}, and \frac{n}{4}$. This way, we will get three error values per each method, err_1, err_2, and err_3 with respect

to $O(\delta x_n)$, $O(\delta x_{n/2})$ and $O(\delta x_{n/4})$. Then we can evaluate the following formula to examine the error quantity

$$\frac{err_1 - err_2}{err_2 - err_3} \qquad (*)$$

Following this manner, we ought to get either approximately 4 or 2 depending on the error rate of the method, assuming that our implmentation is correct. The proof is on appendix A.3

$\delta\tau$	$\delta\mathbf{x}_1$	$\delta\mathbf{x}_2$	$\delta\mathbf{x}_3$	**I**	**CN**	**RS**	**I-I**	**I-CN**	**I-RS**
0.0004	0.15	0.075	0.0375	4.0944	4.1165	4.0980	4.1158	4.2050	4.1132
0.0004	0.12	0.060	0.0300	4.1845	4.1970	4.1915	4.2659	3.7509	4.2228
0.0004	0.075	0.0375	0.01875	3.9247	3.9861	3.9708	3.1821	2.7145	2.9424
0.0001	0.075	0.0375	0.01875	3.9035	3.9301	3.9151	4.2043	3.8704	4.1984

Table 4: Error Validation of an American put option evaluated with respect to $O(\delta x^2)$ by different methods at $t = 0$. Here are the names for each label in the Table 4: I = the implicit method, CN : the Crank-Nicolson method, RS : the Rannacher-Smoothing method, 'I-' refers to the improved version of PSOR. For example, I-I refers to the improved version of PSOR method

The results of computing error checking equation $(*)$ is shown in the Table 4 and Table 5. We made $\delta\tau$ to be small compared to δx, and then ran the program with different delta sizes. Notice on the third row of Table 4, I-I, I-CN, I-RS are not exactly near 4. Thence, with same δxs, we tried decreasing the $\delta\tau$, and then the error ratio came to be near 4. This illustrates that $\delta\tau$ was not small enough to not effect the $O(\delta x^2)$. Therefore, Table 4 demonstrates that the error ratio for each method is 4, which verifies the correctness of the algorithm.

$\delta\mathbf{x}$	$\delta\tau_1$	$\delta\tau_2$	$\delta\tau_3$	**I**	**CN**	**RS**	**I-I**	**I-CN**	**I-RS**
0.010000	0.004	0.002	0.001	3.2622	1.8792	2.7836	2.6862	1.8579	3.0001
0.008571	0.004	0.002	0.001	2.5878	1.8798	2.7525	1.9561	1.0646	1.8599
0.006667	0.004	0.002	0.001	2.0447	1.8787	2.7629	1.3215	30.8960	1.0556
0.006667	0.0004	0.0002	0.0001	1.9918	1.9226	2.4511	21.6636	1.1298	2.8492

Table 5: Error Validation of an American Put Option evaluated with respect to $O(\delta\tau)$ by different methods at $t = 0$. Here are the names for each label in the Table 5: I = the implicit method, CN : the Crank-Nicolson method, RS : the Rannacher-Smoothing method, 'I-' refers to the improved version of PSOR. For example, I-I refers to the improved version of PSOR method

It is harder to observe from Table 5 that the program is outputing the correct error ratio. We tried to make the δx^2 to be much smaller than $\delta\tau$, in order to observe the effect of $O(\delta\tau)$. The correct error ratio for the implicit method is 2 since its error bound is $O(\delta\tau)+O(\delta x^2)$. Table 5 shows that I and I-I did get close to 2. However, it seems that other methods are not getting 2 nor 4. Notice that δx^2 are much smaller than $\delta\tau$, but δx^2 is still about the same size as $\delta\tau^2$. This causes the error ratio to be affected by both $O(\delta x^2)$ and $O(\delta\tau^2)$. We could not make δx^2 to be much smaller than $\delta\tau^2$, because it is computationally unfeasible.

Table 6 shows the price of an American Option with environment as Table 3 computed by the penalty method. The reason why it is calculated separately from other methods is because they are computed from the different space. The formulae for the penalty method is derived from S and τ space, whereas the formulae for other methods are derived from x and τ space. The relationship between S and x is

$$S = Ee^x$$

Notice that there exists an exponential component involved, which makes the evenly space grid to be uneven when it gets transformed to another space. However, our linear system $\mathbf{Mu}^m = \mathbf{b}^m$does not solve the PDE, $\mathcal{L}V$ correctly when the grid is unevenly spaced. Hence, the direct comparison will not be a fair evaluation.

Price	Penalty
0	10.0000
1.2500	8.7500
2.5000	7.5000
3.7500	6.2500
5.0000	5.0000
6.2500	3.7500
7.5000	2.5000
8.7500	1.5379
10.0000	0.8933
11.2500	0.4975
12.5000	0.2692
13.7500	0.1430
15.0000	0.0752
16.2500	0.0393
17.5000	0.0205
18.7500	0.0107

Table 6: Price of an American Put Option evaluated using the penalty method at $t = 0$

Moreover, Table 7 verifies the correctness of the algorithm with respect to $O(\delta S^2)$. But the correctness of the algorithm with respect to $O(\delta\tau^2)$ was not validated. This is due to the suffering from the computational feasibility when we try to increase size of N until $O(\delta\tau^2)$ dominates.

$\delta\tau$	$\delta\mathbf{S}_1$	$\delta\mathbf{S}_2$	$\delta\mathbf{S}_3$	**Implicit-Penalty**	**CN-Penalty**
0.0017	2.5	1.25	0.625	4.5213	4.5429
0.0017	2	1	0.5	3.9739	3.9416
0.0017	1	0.5	0.25	4.1351	3.9247
0.0001	0.075	0.0375	0.01875	3.7679	3.9035

Table 7: Error Validation of an American Put Option evaluated with respect to $O(\delta x^2)$ by the penalty method at $t = 0$.

Next, we evaluated the performance of each method. Typically, the PSOR iteration takes a lot of iterations given its fine grid. However, we suspected that improved PSOR version ought to decrease the iteration step enormously compared to the regular PSOR. Table 8 illustrates the huge improvement, where it converges within eight iterations. This indicates that using the Black-Scholes formula to evaluate the left side of the free boundary does help a lot, and free boundary converges within a very few steps, especially even when the granularity of the grid is very small. In constrast, the penalty method, normally converges within the two iterations.

For the purpose of assessing the overall performance of evaluating an option and to realize the practical runtime scale, we measured the CPU time of each method. We mention that the penalty method and the methods which involves PSOR were computed in the different space, so in order to have a fair comparision between the two, we enforced the matrices sizes to be same. Moreover, we imposed the size of δx and

M	N	$\delta\tau$	δx	Imp-PSOR	CN-PSOR	RS	Improved-Imp	Improved-CN
100	500	0.0004	0.012	63	37	63	8	8
100	550	0.0004	0.0109	74	43	74	8	8
100	650	0.0004	0.01	87	50	87	8	8
100	600	0.0004	0.0092	99	57	99	8	8
10	500	0.0004	0.012	536	305	536	8	8
20	500	0.0002	0.012	278	158	278	8	8
30	500	0.00013	0.012	190	108	190	8	8
40	500	0.0001	0.012	146	83	146	8	8

Table 8: Maximum iteration step to compute an American Put Option Price by different methods

$\delta\tau$ to be approximately the same, so that the errors are allocated distributedly. Although the improved version of the PSOR method made great enhancement in speed, Table 9 indicates that the penalty method has the fastest performance in general. Therefore, the penalty method is the most suitable method in practice.

M	N	I	CN	RS	II	ICN	IRS	Imp-Penalty	CN-Penalty
10	500	0.267633	0.193060	0.222842	0.080366	0.071208	0.070966	0.021927	0.023686
10	600	0.397176	0.232855	0.273083	0.138788	0.095581	0.095268	0.089753	0.067151
10	1000	1.325334	0.789202	0.883911	0.333361	0.320676	0.318319	0.041413	0.072715

Table 9: CPU time for each method on computing an American Put Option Price Here are the names for each label in the Table 9: I = the implicit method, CN : the Crank-Nicolson method, RS : the Rannacher-Smoothing method, 'I-' refers to the improved version of PSOR. For example, I-I refers to the improved version of PSOR method

9 Conclusion

In this paper, we went over evaluating the fair price of the European and American put option. Starting from solving an European option to an American option valuation, we observed that the American option is definitely a challenging problem compared to the European option pricing due to the free boundary condition. We considered finite-difference approaches to solve the Black-Scholes PDE. As such, we still carried out the methods, such as the explicit, implicit, Crank-Nicolson, Rannacher-Smoothing from the European option valuation to the American option valuation problem. This indicates the significance of Black-Scholes PDE, where it is the fundamental PDE equations to all various option pricing problem. Moreover, we investigated the penalty method, which was very practical and efficient for solving an American option pricing. The LU factorization and SOR method were applied for the solution of the Black-Scholes PDE, and the PSOR and penalty method were employed for the solution of the LCP. Our main interest was to inspect the performance between the improved version of PSOR, and the penalty method. After thorough examinations, we concluded that the penalty method is more favorable in terms of performance speed. However, the improved version of the PSOR still elicited big improvement compared to the regular PSOR. Through this work, we consolidated our understanding of the Black-Scholes PDEs and LCPs, and comprehended details of each numerical method on solving PDE problems. Additionally, the emphasis on the importance of the profound understanding and sophisiticated evaluation of an option is again recognized throughout this work. This work can be easily extended to solving an American call option, and can definitely be arched over to solving an American put option with the additional cash flows from dividends.

A Appendix

A.1 Stabiltiy on Explicit Finite-Difference Method

Let $\mathbf{U}^{(m)}$ be option values at time m, and let α be $\frac{\partial\tau}{(\partial x)^2}$. The explicit finite-difference formula in terms on matrix form is

$$\mathbf{u}^{(m+1)} = (\mathbf{I} + \alpha\mathbf{A})\mathbf{u}^{(m)} + \mathbf{b}^{(m)}$$

Let $\mathbf{M} = \mathbf{I} + \alpha\mathbf{A}$. Then

$$\mathbf{u}^{(m+1)} = \mathbf{M}\mathbf{u}^{(m)} + \mathbf{b}^{(m)}$$

Throughout the time step, m, from 1 to m, we will recursively have to compute

$$\mathbf{u}^{(m)} = \mathbf{M}^m\mathbf{u}^{(0)} + \sum_{i=0}^{m-1}\mathbf{M}^i\mathbf{b}^{(i)}$$

Let λ be the eigenvalue of $\mathbf{M}$ such that $\mathbf{M}\mathbf{v} = \lambda\mathbf{v}$

$$\begin{aligned}
&\Rightarrow \mathbf{M}^m\mathbf{v} = \mathbf{M}^{m-1}\mathbf{M}\mathbf{v} = \mathbf{M}^{m-1}\lambda\mathbf{v} = \lambda^m\mathbf{v} \\
&\Rightarrow |\lambda| \leq 1 \text{ , otherwise } \mathbf{M}^m\mathbf{v} \to \infty \text{ as } m \to \infty
\end{aligned}$$

As a consequence of Fourier analysis, assume (without any loss of generality) that u_n^m takes the term $u_n^m = \lambda^n sin(n\omega)$.

$$\begin{aligned}
&\Rightarrow u_n^{m+1} = (u_{n+1}^m + u_{n+1}^m)\alpha + (u_n^m - 2\alpha u_n^m) \\
&\Rightarrow \lambda^{m+1}\sin(n\omega)\alpha = (\lambda^m\sin((n+1)\omega) + \lambda^m\sin((n-1)\omega))\alpha + \lambda^m\sin(n\omega)(1-2\alpha) \\
&\Rightarrow \lambda^{m+1}\sin(n\omega) = \alpha\lambda^m(\sin((n+1)\omega) + \sin((n-1)\omega)) + \lambda^m\sin(n\omega)(1-2\alpha) \\
&\Rightarrow \lambda\sin(n\omega) = \alpha(\sin((n+1)\omega) + \sin((n-1)\omega)) + \sin(n\omega)(1-2\alpha) \\
&\quad \text{since } 2\sin\theta\cos\phi = \sin(\theta+\phi) + \sin(\theta-\phi) \\
&\Rightarrow \lambda\sin(n\omega) = \alpha 2\sin(n\omega)\cos(\omega) + \sin(n\omega)(1-2\alpha) \\
&\Rightarrow \lambda = \alpha 2\cos(\omega) + (1-2\alpha) \\
&\Rightarrow \lambda = \alpha 2(\cos(\omega) - 1) + 1 \\
&\quad \text{since } \sin^2 z = \frac{1-\cos(2z)}{2} \\
&\Rightarrow \lambda = -\alpha 4\sin(\frac{\omega}{2}) + 1 \\
&\Rightarrow \max(\lambda) = 1 \text{ , } \min(\lambda) = 1 - 4\alpha
\end{aligned}$$

Since we know that $|\lambda| \leq 1$,

$$\begin{aligned}
&\Rightarrow -1 \leq 1 - 4\alpha \leq 1 \\
&\Rightarrow -2 \leq 4\alpha \leq 0 \\
&\Rightarrow 0 \leq \alpha \leq \frac{1}{2}
\end{aligned}$$

A.2 Convergence for the Crank-Nicolson Finite-Difference Method

The Crank-Nicolson Finite-Difference formula is

$$\begin{aligned}
\frac{\partial u}{\partial \tau} &= \frac{\partial^2 u}{\partial x^2} \\
\frac{u(x,\tau+\delta\tau) - u(x,\tau)}{\delta\tau} &= \frac{u(x+\delta x,\tau) - 2u(x,\tau) + u(x-\delta x,\tau)}{\delta x^2} \qquad (*)
\end{aligned}$$

Applying a Taylor's series expenson on each term in the equation $(*)$, we get

$$\begin{aligned}
u(x,\tau+\delta\tau) &= (1+\frac{\delta\tau}{2}\frac{\partial}{\partial\tau}+\frac{\delta\tau^2}{4}\frac{\partial^2}{\partial\tau^2}+\frac{\delta\tau^3}{2\cdot 3!}\frac{\partial^3}{\partial\tau^3}+...)u(x,\tau+\frac{\delta\tau}{2})\\
u(x,\tau) &= (1+\frac{\delta\tau}{2}\frac{\partial}{\partial\tau}-\frac{\delta\tau^2}{4}\frac{\partial^2}{\partial\tau^2}+\frac{\delta\tau^3}{2\cdot 3!}\frac{\partial^3}{\partial\tau^3}-...)u(x,\tau+\frac{\delta\tau}{2}))\\
u(x+\delta x,\tau) &= (1+\delta x\frac{\partial}{\partial x}+\frac{\delta x^2}{2}\frac{\partial^2}{\partial x^2}+\frac{\delta x^3}{3!}\frac{\partial^3}{\partial x^3}+...)u(x,\tau)\\
u(x-\delta x,\tau) &= (1+\delta x\frac{\partial}{\partial x}-\frac{\delta x^2}{2}\frac{\partial^2}{\partial x^2}+\frac{\delta x^3}{3!}\frac{\partial^3}{\partial x^3}-...)u(x,\tau))
\end{aligned}$$

Expanding and reformulating the left side of equation $(*)$:

$$\begin{aligned}
\frac{u(x,\tau+\delta\tau)-u(x,\tau)}{\delta\tau} &= \frac{1}{\delta\tau}((1+\frac{\delta\tau}{2}\frac{\partial}{\partial\tau}+\frac{\delta\tau^2}{4}\frac{\partial^2}{\partial\tau^2}+\frac{\delta\tau^3}{2\cdot 3!}\frac{\partial^3}{\partial\tau^3}+...)u(x,\tau+\frac{\delta\tau}{2})\\
&\quad -(1+\frac{\delta\tau}{2}\frac{\partial}{\partial\tau}-\frac{\delta\tau^2}{4}\frac{\partial^2}{\partial\tau^2}+\frac{\delta\tau^3}{2\cdot 3!}\frac{\partial^3}{\partial\tau^3}-...)u(x,\tau+\frac{\delta\tau}{2}))\\
&= \frac{\partial u}{\partial\tau}(x,\tau+\frac{\delta\tau}{2})+\frac{\partial^3 u}{\partial\tau^3}(x,\tau+\frac{\delta\tau}{2})\frac{\delta\tau^2}{2\cdot 3!}\\
&= \frac{\partial u}{\partial\tau}(x,\tau+\frac{\delta\tau}{2})+O((\delta\tau^2))
\end{aligned}$$

Expanding and reformulating the right side of equation $(*)$:

$$\begin{aligned}
\frac{u(x+\delta x,\tau)-u(x-\delta x,\tau)}{2\delta x} &= \frac{1}{2\delta x}((1+\delta x\frac{\partial}{\partial x}+\frac{\delta x^2}{2}\frac{\partial^2}{\partial x^2}+\frac{\delta x^3}{3!}\frac{\partial^3}{\partial x^3}+...)u(x,\tau)\\
&\quad -(1+\delta x\frac{\partial}{\partial x}-\frac{\delta x^2}{2}\frac{\partial^2}{\partial x^2}+\frac{\delta x^3}{3!}\frac{\partial^3}{\partial x^3}-...)u(x,\tau))\\
&= \frac{\partial u}{\partial x}(x,\tau)+\frac{\partial^3 u}{\partial\tau^3}(x,\tau+\frac{\delta\tau}{2})\frac{\delta\tau^2}{3!}\\
&= \frac{\partial u}{\partial x}(x,\tau)+O((\delta x^2))
\end{aligned}$$

Then combining both the left and right derivations aboves, we attain

$$\frac{\partial u}{\partial\tau}+O((\delta\tau^2))=\frac{\partial u}{\partial x}+O((\delta x^2))$$

A.3 Indirectly measuring the correctness of American Option Pricing

Assume that $O(\delta\tau)$ is small compare to $O((\delta x)^2)$, so that $O((\delta x)^2)$ dominates. Consider three possible grids where M is fixed and $N=n$, $\frac{n}{2}$, and $\frac{n}{4}$. Let err_i be the error, such that $v_i - v$ for all $i=1,2,3$, and let v be the true American option price.

$$\begin{aligned}
&= \frac{err_1-err_2}{err_2-err_3}\\
&= \frac{(v_1-v)-(v_2-v)}{(v_2-v)-(v_3-v)}\\
&= \frac{v_1-v_2}{v_2-v_3}\\
&= \frac{c\delta x_n^2-c(\frac{\delta x_n}{2})^2}{c(\frac{\delta x_n}{2})^2-c(\frac{\delta x_n}{4})^2}\\
&= 4
\end{aligned}$$

This can also work vice versa, where we fix N and consider M to be m, $\frac{m}{2}$, and $\frac{m}{4}$.

References

[1] D. Dang, *Adaptive Finite Difference Methods for Valuing American Options*, MSc Thesis, Computer Science Dept., University of Toronto, 2007.

[2] L. Feng, V. Linetsky, J. Luis Morales and J. Nocedal, *On the Solution of Complementarity Problems Arising in American Options Pricing*, Optimization Methods and Software, iFirst, 2010, pp. 1–13.

[3] J. Hull *Options, Futures, and Other Derivatives*, Pearson Education International, Inc., New Jersey, 07458.

[4] J. Morales, J. Nocedal and M. Smelyanskiy, *An Algorithm for the Fast Solution of Symmetric Linear Complementarity Problems*, Numer. Math., 2008, pp. 251-266.

[5] P. Wilmot, S. Howison, and J. Dewynne, *The Mathematics of Financial Derivatives*, Cambridge University Press, 1995.

[6] J. Witte, and C. Reisinger *Penalty Methods for the Solution of Discrete HJB Equations – Continuous Control and Obstacle Problems*, SIAM J. Numer. Anal. 50(2), 595-625, 2012.