North Carolina Journal of
Mathematics and Statistics
Volume 1, Pages 1–16 (Accepted July 10, 2015, published October 1, 2015)
ISSN 2380-7539

Numerical Solutions of American Options with Dividends Using Finite Difference Methods

Nsoki Mavinga and Chi Zhang

ABSTRACT. We study the Black-Scholes model for American options with dividends. We cast the problem as a free-boundary problem for heat equations and use transformations to rewrite the problem in linear complementarity form. We use explicit and implicit finite difference methods to obtain numerical solutions. We implement and test the methods on a particular example in MATLAB®. The effects of dividend payments on option pricing are also considered.

1. Introduction

Black and Scholes (1973) proposed a valuation model for an European option, a contract that allows the holder of the option to exercise the right to buy or sell stocks at the expiration date. Unlike European options, where the payoff is determined by the price of the underlying asset at the exercise date, another primary type of options called American options give the holder the right to early exercise the options at anytime before the expiration date. The American option valuation problem can be viewed as a free-boundary problem, that is, there exists an unknown boundary dependent on time, that sets the line between early exercising and holding the option. Since an American option offers the holder greater rights than an European option, it usually has a larger value. The explicit formula for European options no longer works for American options. Our approach to obtain the result of the American option valuation problem is to rewrite the problem in a linear complementarity form. We then solve it using finite difference methods and the projected Successive Over-Relaxation (SOR) algorithm. The use of the linear complementarity form has a great advantage in that the free boundary points are implicitly included in a single constraint. Those points no longer need to be explicitly mentioned, which facilitates the computation of the option values.

The numerical solutions of American option pricing are widely studied. Wilmott (1995) and Tavella and Curt (2000) provided a detailed description of finite difference methods and its application to American options without dividend. Hull (1997) applied finite difference methods directly to the stochastic differential equations of the American put option pricing problem. The purpose of this paper is to study American Call/Put options with constant dividends and analyze the effect of dividend payments on the pricing of options. Unlike previous results in the literature, what sets our results apart is that we use both linear complementarity and finite difference methods in the analysis. We study the option value problem using the linear complementarity transformation and three finite difference methods, and we compare their numerical values. We present a framework for analyzing the properties of finite difference discretization schemes for solving the pricing equation with a detailed practical example of the analysis. Because many assets pay out

Received by the editors January 29, 2015.
2010 *Mathematics Subject Classification.* 91G60; 91G80.
Key words and phrases. Black-Scholes equations; American call options; Dividends; Finite difference methods.

dividends and there are various structures for dividend payments, a constant-dividend paying structure is important in modeling index options such as the S&P 500 and short-dated currency options where dividend refers to payments at the foreign interest rate. Thus there is a need to analyze the effect of dividend payments on the value of call options. Put options follow a similar analysis so we focus on call options.

The paper is organized as follows. Section 2 is devoted to the analysis of American option valuation problem as an obstacle problem. Section 3 provides the finite difference methods and the algorithm for numerically solving the pricing model. Section 4 presents examples of the numerical results obtained by implementing the algorithms with MATLAB®. Finally, Section 5 concludes on the effect of a constant dividend rate on the option value. The Appendix A includes the MATLAB® code for the three algorithms.

2. American Options

2.1. Free boundary problem

Definition 2.1. *An option represents a contract that offers the holder the right, but not the obligation, to buy (Call) or sell (Put) a security or other financial asset with price S at a specified strike price E. The two primary types of options are European options, which can be exercised only at a specific maturity date T, and American options, which can be exercised at any time before T.*

An American option valuation problem is uniquely specified by a set of constraints. First, the option value must be greater than or equal to the payoff function, $C(S,T) = \max(S-E,0)$, where C denotes the value of a call option. This condition follows from the assumption of no arbitrage opportunity. For an European call option with dividend, its final condition equals the payoff function. As $S \to \infty$, $C(S,t) \to Se^{-d(T-t)}$, where d is the constant dividend rate, since the option becomes the same as the asset without its dividend income. Assume $d > 0$. For large values of S, the value of the European call option is less than its intrinsic value, namely $C(S,t) < \max(S-E,0)$. If we allow for early exercising, there is a straightforward arbitrage opportunity. We can purchase the call option and if we immediately exercise the option by paying E to the seller, we earn a risk-free profit $S-E-C$. However, under the efficient market hypothesis, arbitrage opportunity would not last long because a large number of arbitragers would react instantaneously and push up the option value back to the normal level. Then we conclude that when early exercise is permitted we must impose the constraint $C(S,t) \geq \max(S-E,0)$.

The second constraint deals with the values of S that are optimal from the holder's point of view to exercise an American option. When holding the option is always optimal, the option would have the same value as an European option and the Black-Scholes equation would hold for all S. When having the option of early exercising, at each time we need to determine not only the option value, but also for each value of S, whether or not it should be exercised. This is known as a free boundary problem (see Friedman, 1982; Myneni, 1992). At time t there is a particular value of S which distinguishes two regions: one suggesting holding the option and the other one saying the holder should exercise it. We call this value $S_f(t)$, the optimal exercise price. See Figure 2.1.

Recall the second-order parabolic differential equation used to evaluate European option (see Mikosch, 1998; Wilmott, 1995) is given by

$$\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C(S,t)}{\partial S^2} + (r-d)S\frac{\partial C(S,t)}{\partial S} - rC(S,t) + \frac{\partial C(S,t)}{\partial t} = 0, \tag{2.1}$$

where $C(S,t)$ is the value of a call option, $t \in [0,T]$.

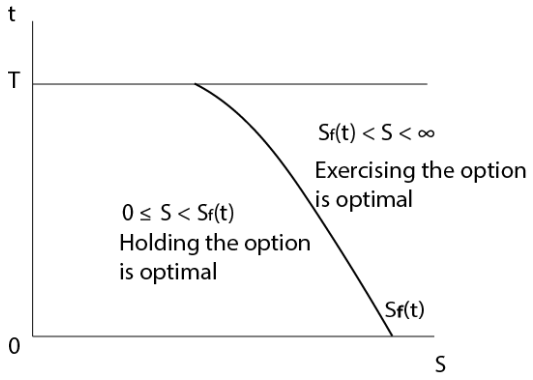

FIGURE 2.1. Free boundary problem.

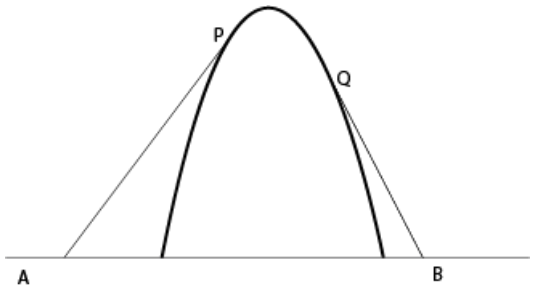

FIGURE 2.2. The obstacle problem (Wilmott, 1995).

For American options, when $C > S - E$, meaning it is optimal to hold the option, the Black-Scholes equation holds. Otherwise, $C = S - E$; it is optimal to exercise the option. The two relationships can be combined into one inequality for the Black-Scholes equation (Barles, 1995)

$$\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C(S,t)}{\partial S^2} + (r-d)S\frac{\partial C(S,t)}{\partial S} - rC(S,t) + \frac{\partial C(S,t)}{\partial t} \leq 0, \tag{2.2}$$

where $C(S,t)$ is the value of a call option, $t \in [0,T]$.

Another two constraints coming from no arbitrage assumption are that the option value has to be continuous since holders can profit from exercising when the asset price reaches the value of the discontinuity and that the change of the option value should also be continuous.

2.2. The obstacle problem

The free boundary problem can be viewed as a classical obstacle problem (Friedman, 1982; Wilmott, 1995), see Figure 2.2. Assume an elastic string is fixed at two points A and B, and passes over a smooth object protruding between the two ends.

Let $x = \pm 1$ be the end points of the string; $u(x)$ be the string displacement; $f(x)$ be the height of the obstacle. We do not have knowledge on the exact region of contact between the string and the obstacle. We only know the string must either be above or on the obstacle, and the string and the slope of the string have to be continuous. The free boundary is the set of points $P(x = x_p)$ and $Q(x = x_Q)$, the points that define the contact region. Since the contact region concaves down, $u = f$ and $u'' < 0$, while $u > f$ and $u'' = 0$ when the string is above the obstacle. We also assume

that

$$f(\pm 1) < 0, f(x) > 0 \text{ for some } -1 < x < 1, f'' < 0 \tag{2.3}$$

to ensure there exists only one contact region.

The obstacle problem is then equivalent to finding $u(x)$ and the points P, Q such that

$$\begin{array}{rcl} u(-1) & = & 0, u(1) = 0 \\ u'' & = & 0, -1 < x < x_P, x_Q < x < 1 \\ u(x_P) & = & f(x_P), u'(x_P) = f'(x_P) \\ u(x_Q) & = & f(x_Q), u'(x_Q) = f'(x_Q) \\ u(x) & = & f(x), x_P < x < x_Q. \end{array} \tag{2.4}$$

One approach to solve this problem is to first write it in the following linear complementarity form

$$u'' \cdot (u - f) = 0, -u'' \geq 0, (u - f) \geq 0, \tag{2.5}$$

where $u(-1) = u(1) = 0$, and u, u' are continuous. This transformation does not explicitly include the free boundary points. Instead, the free boundary problem is implicitly incorporated in the constraint $u \geq f$. We can then use numerical techniques such as iterative methods to solve (2.5).

2.3. Linear complementarity form for American option valuation

Now we reduce the valuation problem of American call option with dividends to a linear complementarity problem. We use the linear complementarity form since the free boundary points can now be implicitly included in a single constraint $u \geq f$. They no longer need to be explicitly mentioned. If we can solve the constraint problem, we can simply look at the resulting values of $u - f$ and the free boundaries are where this function changes from being zero to nonzero.

First, we transform the American call with dividends problem from the original (S, t) variables to (x, τ), where x and τ refer to the following transformation:

$$\begin{array}{rcl} x & = & \ln(S/E), \\ \tau & = & \frac{\sigma^2}{2}(T - t), \\ v(x, \tau) & = & C/E, \text{ and} \\ v & = & e^{\alpha x + \beta \tau} u(x, \tau), \end{array} \tag{2.6}$$

for $-\infty < x < \infty, \tau > 0$, where $\alpha = -\frac{1}{2}(k' - 1), \beta = -\frac{1}{4}(k' - 1)^2 - k$, $k' = \frac{r-d}{\sigma^2/2}$ and $k = \frac{r}{\sigma^2/2}$. Multiplying $v(x, \tau)$ by E gives us the value of the option with respect to variables S and t. Through this set of transformation, the Black-Scholes equation (2.1) becomes

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}, \ -\infty < x < \infty, \ \tau > 0, \tag{2.7}$$

which is a diffusion equation with initial condition

$$u(x, 0) = v(x, 0)e^{-\alpha x} = \max(e^{\frac{1}{2}(k'+1)x} - e^{\frac{1}{2}(k'-1)x}, 0). \tag{2.8}$$

Observe that with the transformation above, the dividend term does not appear explicitly in equation (2.7). Therefore, the obstacle problem for American call options with dividends is equivalent to finding $u(x, \tau)$ and the unknown optimal exercise boundary $x_f(\tau)$ such that

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} \text{ for } x \leq x_f(\tau) \tag{2.9}$$

$$u(x,\tau) = g(x,\tau) \text{ for } x > x_f(\tau), \tag{2.10}$$

with boundary conditions

$$\begin{array}{rcl} u(x,0) & = & g(x,0) \\ \lim\limits_{x\to\infty} u(x,\tau) & = & g(x,\tau) \\ \lim\limits_{x\to-\infty} u(x,\tau) & = & 0, \end{array} \tag{2.11}$$

where

$$g(x,\tau) = e^{(\frac{1}{4}(k'-1)^2+k)\tau} \max(e^{\frac{1}{2}(k'+1)x} - e^{\frac{1}{2}(k'-1)x}, 0). \tag{2.12}$$

We also have the constraint

$$u(x,\tau) \geq g(x,\tau). \tag{2.13}$$

Since we will be focusing on numerical solutions using finite difference methods, we will restrict the problem to a finite interval. Therefore, we consider the problem only for x in the interval $[x^-, x^+]$, where x^- is a large negative number and x^+ is a large positive number. Hence, the boundary conditions are

$$u(x^-,\tau) = 0, \quad u(x^+,\tau) = g(x^+,\tau). \tag{2.14}$$

Now, rewriting in a linear complementarity form, we obtain

$$(\frac{\partial u}{\partial \tau} - \frac{\partial^2 u}{\partial x^2}) \cdot (u(x,\tau) - g(x,\tau)) = 0. \tag{2.15}$$

with two constraints

$$\frac{\partial u}{\partial \tau} - \frac{\partial^2 u}{\partial x^2} \geq 0, u(x,\tau) - g(x,\tau) \geq 0, \tag{2.16}$$

with the initial condition $u(x,0) = g(x,0)$ and the boundary conditions

$$\begin{array}{rcl} u(x^-,\tau) & = & g(x^-,\tau) = 0, \\ u(x^+,\tau) & = & g(x^+,\tau). \end{array} \tag{2.17}$$

The above transformation is helpful because the diffusion equation is more straightforward and less cluttered than the Black-Scholes equation. It is much easier to find numerical solutions of the diffusion equation and then to convert these into numerical solutions of the Black-Scholes equation through a change of variables than to numerically solve the Black-Scholes equation directly. Thus to obtain the numerical result of American option value, our approach is to solve

$$\frac{\partial u}{\partial \tau} - \frac{\partial^2 u}{\partial x^2} = 0 \tag{2.18}$$

and make sure

$$u(x,\tau) - g(x,\tau) \geq 0. \tag{2.19}$$

3. Finite difference methods

When no analytic solution is available, there are many different numerical procedures for valuing derivatives to consider, such as finite difference methods, Monte Carlo simulation and binomial trees. In practice, the method to use depends on what type of derivative is evaluated and what accuracy level is required. Monte Carlo simulation is the most appropriate for European-style derivatives since it works forward from a derivative's life beginning to the end. Finite difference methods and tree approaches work for both American and European options since they solve the problem backward from the end to the beginning of the life of a security.

In this paper, we use three finite difference methods, the explicit, fully-implicit and Crank-Nicolson methods, where the explicit and fully-implicit methods correspond to forward and backward Euler time-stepping methods, and the Crank-Nicolson method is based on trapezoidal time-stepping all with a modification to account for the constraints. Since the purpose of this paper is to study the impact of dividends on option values, we solely focus on these three finite difference methods.

The specific procedure is as follows: We divide the (x, τ) plane into a regular finite mesh, and take finite-difference approximations of the linear complementarity form problem. We approximate the terms $\partial u/\partial \tau - \partial^2 u/\partial x^2$ by the finite differences on a regular mesh with step sizes $\delta\tau$ and δx, and we truncate so that x lies between $N^-\delta x$ and $N^+\delta x$, where N^- is a large negative number and N^+ is a large positive number. We divide the non-dimensional time to expiry of the option, $\frac{1}{2}\sigma^2 T$, into M equal time-steps so that $\delta\tau = \frac{1}{2}\sigma^2 T/M$.

The idea of finite difference methods is to replace the partial derivatives in the equation by their difference-quotient approximations based on Taylor series expansions of functions near the points of interest, and then let the computer solve the resulting differential equation. We divide the x-axis into equally spaced nodes a distance δx apart, and the τ-axis into equally spaced nodes a distance $\delta\tau$ apart. This divides the (x, τ) plane into a mesh, where the mesh points have the form $(n\delta x, m\delta\tau)$, $N^- \leq x \leq N^+$ and $0 < m \leq M$. We write $u_n^m = u(n\delta x, m\delta\tau)$ as the value of $u(x, \tau)$ at the mesh point $(n\delta x, m\delta\tau)$, and $g_n^m = g(n\delta x, m\delta\tau)$.

3.1. Explicit method

Using a forward difference to approximate $\partial u/\partial\tau$, and a second-order central difference for $\partial^2 u/\partial x^2$, we can rewrite the diffusion equation (2.18) as

$$\frac{u_n^{m+1} - u_n^m}{\delta\tau} + O(\delta\tau) = \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\delta x)^2} + O((\delta x)^2). \tag{3.1}$$

If we ignore terms of $O(\delta\tau)$ and $O((\delta x)^2)$, we can rearrange (3.1) to give the difference equation

$$u_n^{m+1} = \alpha u_{n+1}^m + (1 - 2\alpha)u_n^m + \alpha u_{n-1}^m \tag{3.2}$$

where $\alpha = \delta\tau/(\delta x)^2$.

At time step $m + 1$, we already know the values of u_n^m for all n so we can explicitly calculate u_n^{m+1}. After we calculate u_n^{m+1} using (3.2), we also need the equation to satisfy the constraint $u(x, \tau) - g(x, \tau) \geq 0$. Thus the scheme for this explicit method can be expressed as:

$$\begin{array}{rcl} y_n^{m+1} & = & \alpha u_{n+1}^m + (1 - 2\alpha)u_n^m + \alpha u_{n-1}^m \\ u_n^{m+1} & = & \max(y_n^{m+1}, g_n^{m+1}) \end{array} \tag{3.3}$$

With this approach, the stability question arises. The system (3.2) is stable if $0 < \alpha \leq \frac{1}{2}$ and unstable if $\alpha > 1/2$ (see e.g. Wilmott, 1995), which puts severe constraints on the size of time steps. Hence there is need to consider a more stable method, such as the fully-implicit method. The implicit finite-difference method is stable for any $\alpha > 0$, which suggests that we can solve the diffusion equation with larger time-steps using an implicit algorithm than we can using an explicit algorithm.

3.2. Fully-implicit method

We use the backward-difference approximation for $\partial u/\partial \tau$, and the central difference for $\partial^2 u/\partial x^2$. This procedure turns the diffusion equation (2.18) into the following:

$$\frac{u_n^{m+1} - u_n^m}{\delta\tau} + O(\delta\tau) = \frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{(\delta x)^2} + O((\delta x)^2). \tag{3.4}$$

Dropping the error terms, we obtain

$$-\alpha u_{n-1}^{m+1} + (1+2\alpha)u_n^{m+1} - \alpha u_{n+1}^{m+1} = u_n^m. \tag{3.5}$$

The new values cannot be separated out immediately and solved for explicitly in terms of the old values. This is the reason it is called the implicit scheme. The linear complementarity problem (2.15) is then approximated by

$$(-\alpha u_{n-1}^{m+1} + (1+2\alpha)u_n^{m+1} - \alpha u_{n+1}^{m+1} - u_n^m) \cdot (u_n^{m+1} - g_n^{m+1}) = 0 \tag{3.6}$$

at time-step $m+1$.

Let

$$\mathbf{u^m} = \begin{pmatrix} u_{N^-+1}^m \\ u_{N^-+2}^m \\ \vdots \\ u_{N^+-2}^m \\ u_{N^+-1}^m \end{pmatrix}, \ \mathbf{g^m} = \begin{pmatrix} g_{N^-+1}^m \\ g_{N^-+2}^m \\ \vdots \\ g_{N^+-2}^m \\ g_{N^+-1}^m \end{pmatrix} \tag{3.7}$$

$$\mathbf{b^m} = \begin{pmatrix} b_{N^-+1}^m \\ b_{N^-+2}^m \\ \vdots \\ b_{N^+-2}^m \\ b_{N^+-1}^m \end{pmatrix} = \begin{pmatrix} u_{N^-+1}^{m-1} \\ u_{N^-+2}^{m-1} \\ \vdots \\ u_{N^+-2}^{m-1} \\ u_{N^+-1}^{m-1} \end{pmatrix} + \alpha \begin{pmatrix} g_{N^-}^m \\ 0 \\ \vdots \\ 0 \\ g_{N^+}^m \end{pmatrix} \tag{3.8}$$

and the coefficient matrix

$$\mathbf{C} = \begin{pmatrix} (1+2\alpha) & -\alpha & 0 & \cdots & 0 \\ -\alpha & (1+2\alpha) & \ddots & & \vdots \\ 0 & -\alpha & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -\alpha \\ 0 & \cdots & 0 & -\alpha & (1+2\alpha). \end{pmatrix} \tag{3.9}$$

We want to solve the following constrained matrix problem

$$\mathbf{C}\mathbf{u^{m+1}} = \mathbf{b^{m+1}} \tag{3.10}$$

and check if the $\mathbf{u} \geq \mathbf{g}$ constraint is satisfied. To solve equation (3.10), observe that $\mathbf{C}$ is an invertible matrix due to the Gerschgorin theorem (Varga, 1962). Therefore, equation (3.10) has a solution.

Theorem 3.1 (Gerschgorin)**.** *Let $A = (a_{i,j})$ be an arbitrary $n \times n$ complex matrix, and let*

$$\Lambda_i \equiv \sum_{j=1, j\neq i}^{n} |a_{i,j}|, 1 \leq i \leq n. \tag{3.11}$$

Then, all the eigenvalues λ of A lie in the union of the disks

$$|z - a_{i,i}| \leq \Lambda_i, 1 \leq i \leq n. \tag{3.12}$$

Observe that the matrix $\mathbf{C}$ is symmetric so all of the eigenvalues are real. Then, let $a_{i,i} = 1+2\alpha$. For $i = 1$ and n, $\Lambda_i = \alpha$. Then $z \geq a_{i,i} - \Lambda_i = 1 + \alpha$ so for $\alpha \geq 0$, z is always positive. For $2 \leq i \leq n-1$, $\Lambda_i = (\alpha) + (\alpha) = 2\alpha$. Then $z \geq a_{i,i} - \Lambda_i = 1$. Thus, for $\alpha \geq 0$, all the eigenvalues of $\mathbf{C}$ are positive real numbers, suggesting $\mathbf{C}$ is invertible and we can rewrite equation (3.10) as

$$\mathbf{u^{m+1}} = \mathbf{C^{-1}}\mathbf{b^{m+1}}. \tag{3.13}$$

We can first form $\mathbf{b^{m+1}}$ from $\mathbf{u^m}$ and the boundary conditions. Using the known initial condition $\mathbf{u^0}$, we can obtain $\mathbf{u^{m+1}}$ sequentially using an iterative method such as Projected SOR.

3.3. Projected SOR algorithm

Projected SOR is a minor modification of the SOR method, Successive Over-Relaxation. The SOR method is used to speed up convergence of iterations. The projected SOR algorithm involves five steps. The first step is to rewrite equation (3.5) as:

$$u_n^{m+1} = \frac{1}{1+2\alpha}(b_n^{m+1} + \alpha(u_{n-1}^{m+1} + u_{n+1}^{m+1})), \tag{3.14}$$

where $N^- + 1 \leq n \leq N^+ - 1$.

Let $u_n^{m,k}$ be the k-th iterate for u_n^m. Let us denote the initial guess by $u_n^{m,0}$. As $k \to \infty$, we expect $u_n^{m,k} \to u_n^m$. We update the values as soon as they are available and add a correction term $u_n^{m,k+1} - u_n^{m,k}$ to the original $u_n^{m,k+1}$. We also incorporate an over-correction or over-relaxation parameter ω, which has optimal values between 1 and 2, and finally, we take the maximum between the estimated $u_n^{m+1,k}$ and the payoff g_n^{m+1}. The fully-implicit scheme is thus expressed as

$$\begin{array}{rcl} y_n^{m+1,k+1} & = & \frac{1}{1+2\alpha}(b_n^{m+1} + \alpha(u_{n-1}^{m+1,k+1} + u_{n+1}^{m+1,k})), \\ u_n^{m+1,k+1} & = & \max(u_n^{m+1,k} + \omega(y_n^{m+1,k+1} - u_n^{m+1,k}), g_n^{m+1}) \end{array} \tag{3.15}$$

Until the difference between $u_n^{m+1,k+1}$ and $u_n^{m+1,k}$ is small enough to be ignored, we set $u_n^{m+1} = u_n^{m+1,k+1}$.

3.4. The Crank-Nicolson method

One improvement the Crank-Nicolson Method has over the fully-implicit method is that it increases the temporal convergence rate from $O(\delta\tau)$ to $O((\delta\tau)^2)$. The equation of the scheme is

$$u_n^{m+1} - \frac{1}{2}\alpha(u_{n-1}^{m+1} - 2u_n^{m+1} + u_{n+1}^{m+1}) = u_n^m + \frac{1}{2}\alpha(u_{n-1}^m - 2u_N^m + u_{n+1}^m), \tag{3.16}$$

where $\alpha = \delta\tau/(\delta x)^2$.
Let $Z_n^m = (1-\alpha)u_n^m + \frac{1}{2}\alpha(u_{n+1}^m + u_{n-1}^m)$, then

$$(1+\alpha)u_n^{m+1} - \frac{1}{2}\alpha(u_{n+1}^{m+1} + u_{n-1}^{m+1}) = Z_n^m. \tag{3.17}$$

Define a new $\mathbf{b}$ matrix by

$$\mathbf{b^m} = \begin{pmatrix} b_{N^-+1}^m \\ b_{N^-+2}^m \\ \vdots \\ b_{N^+-2}^m \\ b_{N^+-1}^m \end{pmatrix} = \begin{pmatrix} Z_{N^-+1}^m \\ Z_{N^-+2}^m \\ \vdots \\ Z_{N^+-2}^m \\ Z_{N^+-1}^m \end{pmatrix} + \frac{1}{2}\alpha \begin{pmatrix} g_{N^-}^{m+1} \\ 0 \\ \vdots \\ 0 \\ g_{N^+}^{m+1} \end{pmatrix} \tag{3.18}$$

The tri-diagonal symmetric matrix $\mathbf{C}$ becomes

$$\mathbf{C} = \begin{pmatrix} (1+\alpha) & -\frac{1}{2}\alpha & 0 & \cdots & 0 \\ -\frac{1}{2}\alpha & (1+\alpha) & \ddots & & \vdots \\ 0 & -\frac{1}{2}\alpha & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -\frac{1}{2}\alpha \\ 0 & \cdots & 0 & -\frac{1}{2}\alpha & (1+\alpha) \end{pmatrix}. \tag{3.19}$$

Using Theorem 3.1, we see that the matrix $\mathbf{C}$ is invertible. Similar to the fully-implicit method, we adopt the projected SOR algorithm and obtain

$$\begin{array}{rcl} y_n^{m+1,k+1} & = & \frac{1}{1+\alpha}(b_n^m + \frac{1}{2}\alpha(u_{n-1}^{m+1,k+1} + u_{n+1}^{m+1,k})) \\ u_n^{m+1,k+1} & = & \max(u_n^{m+1,k} + \omega(y_n^{m+1,k+1} - u_n^{m+1,k}), g_n^{m+1}). \end{array} \tag{3.20}$$

Until the difference between $u_n^{m+1,k+1}$ and $u_n^{m+1,k}$ is small enough to be ignored, we set $u_n^{m+1} = u_n^{m+1,k+1}$.

4. Numerical results

The following results are obtained through implementing the three finite difference algorithms described in the previous sections in MATLAB®. Table 4.1 shows that the three finite difference methods provide us similar values. One limitation for the explicit scheme is stable only when the ratio of the time step to the square of the space step is not greater than $1/2$, which imposes restrictions on the number of necessary time steps. Though the explicit method is relatively easy to implement, the fully-implicit and the Crank-Nicolson methods have better stability properties.

Besides comparing the results of the three methods, we are also interested in the effects of dividend payments on option pricing. Table 4.2 suggests that as dividend rate increases, call option value decreases. A firm's dividend payout policy affects option values because a high dividend payment decreases the rate of growth of the asset price. Thus there is a lower expected rate of capital gain, leading to a lower potential payoff. In addition, since the cash dividend is received by whoever owns the stock until the ex-dividend date, holders might exercise the option just prior to the ex-dividend date. Exercising prior to the ex-dividend date is always optimal when the dividend payment is large. Early exercise will be likely to happen only if the asset is expected to pay a dividend prior the expiration date.

TABLE 4.1. A comparison of the option values of the three methods at $E = 8, d = 0.08, r = 0.1, \sigma = 0.4$ and $T = 1$.

S	Explicit	Fully-implicit	Crank-Nicolson
4	0.0472	0.0482	0.0476
6	0.4359	0.4321	0.4326
8	1.3850	1.3651	1.3663
11	3.6256	3.5338	3.5339
12	4.5139	4.3766	4.3764
15	7.4046	7.0515	7.0507

TABLE 4.2. An example of the option values corresponding to $E = 8, r = 0.1, \sigma = 0.4$ and $T = 1$ under the fully-implicit scheme.

S	d=0.03	d=0.05	d=0.06	d=0.08	d=0.11
4	0.0669	0.0587	0.0550	0.0482	0.0393
6	0.5466	0.4981	0.4753	0.4321	0.3735
8	1.6355	1.5229	1.4688	1.3651	1.2200
10	3.1750	2.9958	2.9090	2.7406	2.5021
11	4.0442	3.8343	3.7322	3.5338	3.2543
12	4.9624	4.7220	4.6048	4.3766	4.0612

5. Conclusion

Option pricing has been an increasingly popular field to study. Options provide investors limited downside in speculative trading and enable investors to hedge and to minimize risk. The value of options depends on a number of variables such as stock price S, exercise price E, Volatility σ, time to expiration T, interest rate r and dividend payments d. The study of American call options with dividends with transformations to a linear complementarity form has not been conducted before. Therefore, this paper uses three finite difference methods to solve the option valuation problem of American call with dividends and implements the methods in MATLAB to obtain numerical results that gauge the potential for computing based on the linear complementarity form. The explicit method suffers from the limitation on the number of time steps, and while both the fully-implicit method and the Crank-Nicolson method address the stability problem, the former method does not converge as fast as the later one. We see from the numerical values and financial theories that there is a negative relationship between dividend payouts and call option values. Option holders must take into consideration the effect of dividend when deciding if the option should be early exercised.

Acknowledgements

The author wishes to thank Swarthmore College for the summer research funding support.

Appendix A. MATLAB® implementation

A.1. The explicit method

```
%Give parameter values and define variables
S = 15;
Smin=0.4;
Smax=150;
E = 8;
r = 0.1;
d = 0.08;
sigma = 0.4;
T=1;
t=0;
omega=1.2;
tol=0.001;

X=log(S/E);
k=r/(0.5*sigma^2);
kd = (r-d)/(0.5*sigma^2);

M = 100;
Nminus = -100;
Nplus = 100;
N = Nplus-Nminus;
u=zeros(N+1,M+1);
dt=(0.5*sigma^2*T)/M;
Xzero = log(Smin/E);
Xmax = log(Smax/E);
dX=(Xmax-Xzero)/N;
alpha = dt/(dX*dX);

Xmesh=Xzero:dX:Xmax;
Tmesh=0:dt:(0.5*sigma^2*T);

%Establish the transformed u matrix
solution_mesh=zeros(N+1,M+1);

%Establish the payoff matrix
g = zeros(N+1,M+1);

for n=1:N+1
    for m=2:M+1
        g(n,m)=exp((0.25*(kd-1)^2+k)*((m-1)*dt))*(max((exp(0.5*(kd+1)*(n+Nminus-1)*dX)-
        exp(0.5*(kd-1)*(n+Nminus-1)*dX)),0));
    end
    g(n,1)= max((exp(0.5*(kd+1)*(n+Nminus-1)*dX)-exp(0.5*(kd-1)*(n+Nminus-1)*dX)),0);
end
```

```
g(1,:)=0;

%The boundary conditions of the u matrix
solution_mesh(:,1)= g(:,1);
solution_mesh(1,:)= g(1,:);
solution_mesh(N+1,:)= g(N+1,:);

%The projected-SOR algorithm
a = alpha;
b = 1-2*alpha;
c = alpha;

y=zeros(N+1,M+1);

for p = 1:M
    y(2:N,p+1)=b*solution_mesh(2:N,p)+a*solution_mesh(3:N+1,p)+c*solution_mesh(1:N-1,p);
    solution_mesh(2:N,p+1)=max(y(2:N,p+1),g(2:N,p+1));
end

uresult = interp1(Xmesh,solution_mesh(:,M+1),X);

%Obtain the option value
value=E*E^(0.5*(kd-1))*S^(-0.5*(kd-1))*exp((-(1/4)*(kd-1)^2-k)*0.5*sigma^2*(T-t))*uresult
```

A.2. The fully-implicit method

```
%Give parameter values and define variables
S = 15;
Smin=0.4;
Smax=150;
E = 8;
r = 0.1;
d = 0.08;
sigma = 0.4;
T=1;
t=0;
omega=1.2;
tol=0.001;

X=log(S/E);
k=r/(0.5*sigma^2);
kd = (r-d)/(0.5*sigma^2);
M = 200;
Nminus = -100;
Nplus = 100;
N = Nplus-Nminus;
u=zeros(N+1,M+1);
dt=(0.5*sigma^2*T)/M;
Xzero = log(Smin/E);
```

```
Xmax = log(Smax/E);
dX=(Xmax-Xzero)/N;
alpha = dt/(dX*dX);
Xmesh=Xzero:dX:Xmax;
Tmesh=0:dt:(0.5*sigma^2*T);

%Establish the payoff matrix
g = zeros(N+1,M+1);
for n=1:N+1
    for m=2: M+1
        g(n,m)=exp((0.25*(kd-1)^2+k)*((m-1)*dt))*(max((exp(0.5*(kd+1)*(n+Nminus-1)*dX)-
        exp(0.5*(kd-1)*(n+Nminus-1)*dX)),0));
    end
    g(n,1)= max((exp(0.5*(kd+1)*(n+Nminus-1)*dX)-exp(0.5*(kd-1)*(n+Nminus-1)*dX)),0);
end
g(1,:)=0;

%The boundary conditions of the u matrix
u(:,1)= g(:,1);
u(1,:)= g(1,:);
u(N+1,:)= g(N+1,:);

a = -alpha;
b = 1+2*alpha;
c = -alpha;

%The projected SOR algorithm
for p=1:M
    temp=zeros(N-1,1);
    temp(1)=a*g(1,p+1);
    temp(end)=c*g(N+1,p+1);
    RHS=u(2:N,p)-temp;
    b=RHS;
    x=max(u(2:N,p),g(2:N,p+1));
    xold=1000*x;
    n=length(x);

    while norm(xold-x)>tol
        xold=x;
        for i=1:n
            if i==1
                z = (b(i)+alpha*x(i+1))/(1+2*alpha);
                x(i) = max(omega*z + (1-omega)*xold(i),g(i,p));
            elseif i==n
                z = (b(i)+alpha*x(i-1))/(1+2*alpha);
                x(i) = max(omega*z + (1-omega)*xold(i),g(i,p));
            else
                z = (b(i)+alpha*(x(i-1)+x(i+1)))/(1+2*alpha);
                x(i) = max(omega*z + (1-omega)*xold(i),g(i,p));
            end
```

```
        end
    end
    u(2:end-1,p+1)=x;
end

uresult=interp1(Xmesh,u(:,M+1),X);

%Obtain the option value
value = E*E^(0.5*(kd-1))*S^(-0.5*(kd-1))*exp((-(1/4)*(kd-1)^2-k)*0.5*sigma^2*T)*uresult
```

A.3. The Crank-Nicolson method

```
%Give parameter values and define variables
S = 15;
Smin=0.4;
Smax=150;
E = 8;
r = 0.1;
d = 0.08;
sigma = 0.4;
T=1;
t=0;
omega=1.2;
tol=0.001;

X=log(S/E);
k=r/(0.5*sigma^2);
kd = (r-d)/(0.5*sigma^2);

M = 200;
Nminus = -100;
Nplus = 100;
N = Nplus-Nminus;
u=zeros(N+1,M+1);
dt=(0.5*sigma^2*T)/M;
Xzero = log(Smin/E);
Xmax = log(Smax/E);
dX=(Xmax-Xzero)/N;
alpha = dt/(dX*dX);

Xmesh=Xzero:dX:Xmax;
Tmesh=0:dt:(0.5*sigma^2*T);

%Establish the payoff matrix
g = zeros(N+1,M+1);

for n=1:N+1
    for m=2: M+1
        g(n,m)=exp((0.25*(kd-1)^2+k)*((m-1)*dt))*(max((exp(0.5*(kd+1)*(n+Nminus-1)*dX)-
```

```
        exp(0.5*(kd-1)*(n+Nminus-1)*dX)),0));
    end
    g(n,1)= max((exp(0.5*(kd+1)*(n+Nminus-1)*dX)-exp(0.5*(kd-1)*(n+Nminus-1)*dX)),0);
end
g(1,:)=0;

%The boundary conditions of the u matrix
u(:,1)= g(:,1);
u(1,:)= g(1,:);
u(N+1,:)= g(N+1,:);

a = -alpha;
b = 1+2*alpha;
c = -alpha;

%The projected SOR algorithm
zmatrix = zeros(N+1,M+1);
for p=1:M
    temp=zeros(N-1,1);
    temp(1)=a*g(1,p+1);
    temp(end)=c*g(N+1,p+1);
    zmatrix(2:N,p)=(1-alpha)*u(2:N,p)+0.5*alpha*(u(3:N+1,p)+u(1:N-1,p));
    RHS=zmatrix(2:N,p)-temp;
    b=RHS;
    x=max(u(2:N,p),g(2:N,p+1));
    xold=1000*x;
    n=length(x);

    while norm(xold-x)>tol
        xold=x;
        for i=1:n
            if i==1
                z = (b(i)+0.5*alpha*x(i+1))/(1+alpha);
                x(i) = max(omega*z + (1-omega)*xold(i),g(i,p));
            elseif i==n
                z = (b(i)+0.5*alpha*x(i-1))/(1+alpha);
                x(i) = max(omega*z + (1-omega)*xold(i),g(i,p));
            else
                z = (b(i)+0.5*alpha*(x(i-1)+x(i+1)))/(1+alpha);
                x(i) = max(omega*z + (1-omega)*xold(i),g(i,p));
            end
        end
    end
    u(2:end-1,p+1)=x;
end

uresult=interp1(Xmesh,u(:,M+1),X);

%Obtain the option value
value = E*E^(0.5*(kd-1))*S^(-0.5*(kd-1))*exp((-(1/4)*(kd-1)^2-k)*0.5*sigma^2*T)*uresult
```

References

Barles, G. (1995). Critical stock price near expiration. *Mathematical Finance*, 5:77–95.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *The Journal of Political Economy*, 81:637–654.

Friedman, A. (1982). *Variational Principles and Free-Boundary Problems*. Wiley, New York.

Hull, J. C. (1997). *Options, Futures, and Other Derivatives*. Prentice-Hall, New Jersey.

Mikosch, T. (1998). *Elementary Stochastic Calculus with Finance in View*. World Scientific Publishing Co. Pte. Ltd., Singapore.

Myneni, R. (1992). The pricing of the american option. *The Annals of Applied Probability*, 2:1–23.

Tavella, D. and Curt, R. (2000). *Pricing Financial Instruments: The Finite Difference Method*. John Wiley&Sons, New York.

Varga, R. S. (1962). *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Wilmott, P. (1995). *The Mathematics of Financial Derivatives*. Cambridge University Press, London; New York.

(Nsoki Mavinga) SWARTHMORE COLLEGE, SWARTHMORE, PA 19081, USA
E-mail address: nmaving1@swarthmore.edu

(Chi Zhang) SWARTHMORE COLLEGE, SWARTHMORE, PA 19081, USA
E-mail address, Corresponding author: czhang1@swarthmore.edu, czsasha@gmail.com