International Journal of Applied Mathematics

Volume 28 No. 6 2015, 691-701
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v28i6.4

APPROXIMATION OF THE FREE BOUNDARY OF AN AMERICAN CALL OPTION BY FINITE DIFFERENCES ON PARALLELOGRAMS

Juan C. Aguilar

Instituto Tecnológico Autónomo de México (ITAM)
Departamento de Matemáticas, México, D.F. 01080
Rio Hondo No. 1 Col. Progreso Tizapan
Delegacion Alvaro Obregon, C.P. 01080, MEXICO D.F.

Abstract: In this work we construct a numerical scheme based on finite differences to approximate the free boundary of an American call option. Points of the free boundary are calculated by approximating the solution of the Black-Scholes partial differential equation with finite differences on domains that are parallelograms for each time step. Numerical results are reported.

AMS Subject Classification: 65M06, 65M32, 91G60
Key Words: American call option, free boundary, Black-Scholes equation, Crank-Nicolson discretization method, Newton's method, finite differences

1. Introduction

In this paper we propose a numerical method to approximate the free boundary and the value of an American call option $C(S,t)$ given as the solution of the Black-Scholes equation ([1])

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + (r - D_0) S \frac{\partial C}{\partial S} - rC = 0, \tag{1}$$

where σ is the volatility of the underlying asset, T is the expiry, r is the interest

Received: August 6, 2015

rate, and D_0 is a constant and continuous dividend yield. The time variable t is on an interval $[0,T]$, the asset value S is on the range $[0,\infty)$. The call value C is known at time T, $C(S,T) = \max(S-E,0)$, where the positive constant E is the exercise price. The domain where $C(S,t)$ satisfies (1) is of the form $0 \leq t < T$, $0 < S < S_f(t)$, where S_f an unknown function of time representing a free boundary interpreted as the optimal exercise price. Boundary values of the equation are of the form $C(0,t) = 0$ for any $t \in [0,T]$, and $C(S_f(t),t) = S_f(t) - E$, $\frac{\partial C}{\partial S}(S_f(t),t) = 1$, for any $t \in [0,T]$.

We restrict this work to the case where $r > D_0 > 0$. At $t = T$ the value of the free boundary is known to be rE/D_0 (see [6]). In the next two section we describe an algorithm to approximate the free boundary $S_f(t)$, and the call value $C(S,t)$. The method approximates the free boundary by solving numerically a sequence of differential equations defined on parallelograms and rectangles by means of the Crank-Nicolson discretization combined with a shooting method (see [10]), where Newton's algorithm to approximate roots is implemented. The total computational cost of the algorithm is q times the computational cost of the Crank-Nicolson method, where q is the number of iterations of Newton's method; in this work we used $q = 7$. Other approaches to approximate the free boundary are based on integral equations [9], finite differences (see [5, 11] for instance), binomial methods [4].

2. Change of Variables

In this section we rewrite Equation (1) as a constant coefficient differential equation. Following [11], by making the change of variables $S = Ee^x$, $t = T - 2\tau/\sigma^2$, and introducing the function $u = u(x,\tau)$ related to the call value $C(S,t)$ by the equation $C(S,t) = S - E + Eu(x,\tau)$, the Black-Scholes equation can be written in the new variables x, τ as

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} + (k_1 - 1)\frac{\partial u}{\partial x} - ku + f(x), \tag{2}$$

where $k = \dfrac{2r}{\sigma^2}$, $k_1 = 2(r - D_0)/\sigma^2$, and $f(x) = (k_1 - k)e^x + k$. The variable x is in the range $(-\infty,\infty)$, and $\tau \in [0, \sigma^2 T/2]$. At $\tau = 0$ the function u has values $u(x,0) = \max(1 - e^x, 0)$. Moreover the condition $C(0,t) = 0$ implies that $\lim_{x\to-\infty} u(x,\tau) = 1$ for any τ. On the new variables the free boundary is an unknown function x_f of the variable τ with the properties: $u(x_f(\tau),\tau) = 0$, $\frac{\partial u}{\partial x}(x_f(\tau),\tau) = 0$, $x_f(0) = \ln(r/D_0)$.

3. Description of the Algorithm Based on the Crank-Nicolson Method

We now describe how to approximate the free boundary and the solution $u = u(x, \tau)$ of Equation (2). Let $0 = \tau_0 < \tau_1 < ... < \tau_n = \sigma^2 T/2$ be a partition of $[0, \sigma^2 T/2]$. For $\tau \in (\tau_0, \tau_1]$, the domain where $u = u(x, \tau)$ satisfies Equation (2) is of the form shown in Figure 1.

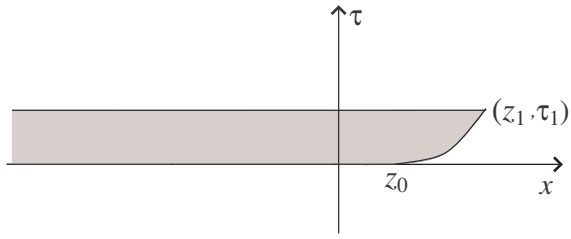

Figure 1: Domain of the solution u of Equation (2).

Let $z_i = x_f(\tau_i)$, $i = 0, 1, ..., n$. The point $(z_i, \tau_i) = (x_f(\tau_i), \tau_i)$ on the free boundary curve is unknown if $i > 0$, and the point (z_0, τ_0) is known with $z_0 = \ln(r/D_0)$. Since $\lim_{x \to -\infty} u(x, \tau) = 1$ for any τ, let $L > 0$ be a constant large enough so that $u(z_i - L, \tau_i) \approx 1$ for $i = 0, 1, ..., n$. On the parallelogram with vertices $(z_i - L, \tau_i)$, (z_i, τ_i), (z_{i+1}, τ_{i+1}), $(z_{i+1} - L, \tau_{i+1})$ the function u satisfies equation (2) together with the boundary conditions $u(z_{i+1}, \tau_{i+1}) = 0$, $u(z_{i+1} - L, \tau_{i+1}) \approx 1$, and initially u has values $u(x, \tau_0) = \max(1 - e^x, 0)$ for $x \in [z_0 - L, z_0]$. Furthermore, $\dfrac{\partial u}{\partial x}(z_{i+1}, \tau_{i+1}) = 0$. We now describe how to approximate $z_1, z_2, ..., z_n$; as we will see, Newton's method approximates z_{i+1} by trying to solve the equation $\dfrac{\partial u}{\partial x}(z, \tau_{i+1}) = 0$ using a suitable approximation to the function u and an initial approximation to z_{i+1}. Define $\tilde{z}_0 = z_0$ and suppose an approximation $\tilde{z}_i$ to z_i has been calculated. Define $\tilde{u}(x, \tau_0) = \max(1 - e^x, 0)$ for $x \in [\tilde{z}_0 - L, \tilde{z}_0]$ and assume $\tilde{u}(x, \tau_i)$ has been calculated for $x \in [\tilde{z}_i - L, \tilde{z}_i]$. To

compute an approximation $\tilde{z}_{i+1}$ to z_{i+1}, we construct a sequence of q numbers $z_{i+1}^{(1)}$, $z_{i+1}^{(2)}$,...,$z_{i+1}^{(q)}$ that approximate z_{i+1} by means of Newton's method starting with an initial approximation $z_{i+1}^{(0)}$ which we define as $z_{i+1}^{(0)} = 2\tilde{z}_i - \tilde{z}_{i-1}$ if $i > 0$, and $z_1^{(0)} = z_0 + 0.9034\sqrt{\tau_1}$ [1]. We will define $\tilde{z}_{i+1}$ as $z_{i+1}^{(q)}$ and it will be the approximation to z_{i+1}. In what follows we describe how to compute $\tilde{z}_{i+1}$ using Newton's method and how to approximate $u(x, \tau_{i+1})$ for $x \in [\tilde{z}_{i+1} - L, \tilde{z}_{i+1}]$. We first construct a sequence of parallelograms $A_0, A_1, ...$ as follows: with the initial approximation $z_{i+1}^{(0)}$ to z_{i+1} we define the parallelogram A_0 with vertices $(\tilde{z}_i - L, \tau_i)$, $(\tilde{z}_i, \tau_i)$, $(z_{i+1}^{(0)}, \tau_{i+1})$, $(z_{i+1}^{(0)} - L, \tau_{i+1})$ shown in Figure 2. Now A_0 is used to obtain a new approximation $z_{i+1}^{(1)}$ to z_{i+1}, and $z_{i+1}^{(1)}$ is used to defined a parallelogram A_1 with vertices $(\tilde{z}_i - L, \tau_i)$, $(\tilde{z}_i, \tau_i)$, $(z_{i+1}^{(1)}, \tau_{i+1})$, $(z_{i+1}^{(1)} - L, \tau_{i+1})$, and so on. In general, the parallelogram A_p has vertices $(\tilde{z}_i - L, \tau_i)$, $(\tilde{z}_i, \tau_i)$, $(z_{i+1}^{(p)}, \tau_{i+1})$, $(z_{i+1}^{(p)} - L, \tau_{i+1})$.

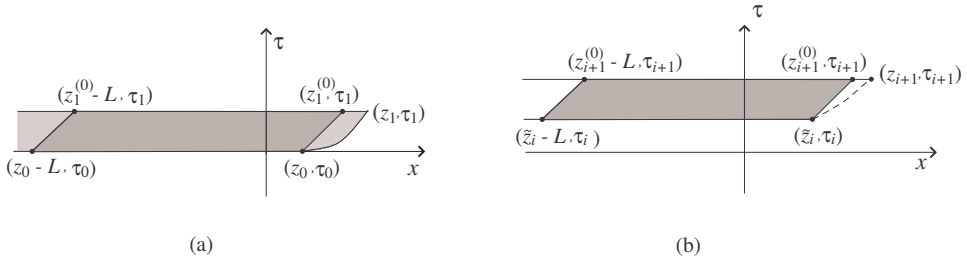

Figure 2: (a) Starting parallelogram A_0 in the approximation of z_1. (b) Starting parallelogram A_0 in the approximation of z_{i+1}.

We describe now how to obtain $z_{i+1}^{(p+1)}$ from $z_{i+1}^{(p)}$. Let $u_i^{(p)}$ be a solution of Equation (2) on the parallelogram A_p with boundary conditions $u_i^{(p)}(z_{i+1}^{(p)} - L, \tau_{i+1}) = 1$, $u_i^{(p)}(z_{i+1}^{(p)}, \tau_{i+1}) = 0$ and initial data $u_i^{(p)}(x, \tau_i) = \tilde{u}(x, \tau_i)$ for $x \in$

[1] The value of the initial approximation $z_1^{(0)}$ is based on the expansion of the free boundary for small values of τ (see [11]); the initial approximation $z_{i+1}^{(0)} = 2\tilde{z}_i - \tilde{z}_{i-1}$ if $i > 0$ was obtained by intersecting the straight line through the points $(\tilde{z}_{i-1}, \tau_{i-1})$, $(\tilde{z}_i, \tau_i)$ with the line $\tau = \tau_{i+1}$ assuming that $\tau_{i+1} - \tau_i = \tau_i - \tau_{i-1}$.

$[\tilde{z}_i - L, \tilde{z}_i]$. Once $u_i^{(p)}$ has been computed, $z_{i+1}^{(p+1)}$ is defined as

$$z_{i+1}^{(p+1)} = z_{i+1}^{(p)} - \frac{\frac{\partial u_i^{(p)}}{\partial x}(z_{i+1}^{(p)}, \tau_{i+1})}{\frac{\partial^2 u_i^{(p)}}{\partial x^2}(z_{i+1}^{(p)}, \tau_{i+1})}. \tag{3}$$

Since $u_i^{(p)}$ will be approximated numerically, the partial derivatives $\frac{\partial u_i^{(p)}}{\partial x}(z_{i+1}^{(p)}, \tau_{i+1})$ and $\frac{\partial^2 u_i^{(p)}}{\partial x^2}(z_{i+1}^{(p)}, \tau_{i+1})$ are approximated through backward finite differences.

We now turn to the problem of approximating the function $u_i^{(p)}$. By definition the function $u_i^{(p)}$ satisfies the partial differential equation

$$\frac{\partial u_i^{(p)}}{\partial \tau} = \frac{\partial^2 u_i^{(p)}}{\partial x^2} + (k_1 - 1)\frac{\partial u_i^{(p)}}{\partial x} - k u_i^{(p)} + f(x), \tag{4}$$

on the parallelogram A_p. In addition, $u_i^{(p)}$ satisfies the boundary conditions $u_i^{(p)}(z_{i+1}^{(p)} - L, \tau_{i+1}) = 1$, $u_i^{(p)}(z_{i+1}^{(p)}, \tau_{i+1}) = 0$, and initial value $u_i^{(p)}(x, \tau_i) = \tilde{u}(x, \tau_i)$ for $x \in [\tilde{z}_i - L, \tilde{z}_i]$. Introduce a new variable ω related to the x and τ variables with the formula $w = x + s_{i,p}(\tau - \tau_i) + R_i$ where $s_{i,p} = \dfrac{\tilde{z}_i - z_{i+1}^{(p)}}{\tau_{i+1} - \tau_i}$, and $R_i = z_0 - \tilde{z}_i$. In the $w - \tau$ plane, the point (w, τ) belongs to a rectangle with vertices $(z_0 - L, \tau_i), (z_0, \tau_i), (z_0 - L, \tau_{i+1}), (z_0, \tau_{i+1})$ if the point (x, τ) is on parallelogram A_p.

Define a function $\tilde{v}$ at τ_0 as $\tilde{v}(w, \tau_0) = \max(1 - e^w, 0)$, and $\tilde{v}(w, \tau_i) = \tilde{u}(w - R_i, \tau_i)$ for $w \in [z_0 - L, z_0]$. Let $v_i^{(p)}$ be a function given by $v_i^{(p)}(w, \tau) = u_i^{(p)}(x, \tau) = u_i^{(p)}(w - s_{i,p}(\tau - \tau_i) - R_i, \tau)$. Since $u_i^{(p)}$ is a solution to Equation (4), it follows that $v_i^{(p)}$ satisfies the constant coefficient differential equation

$$\frac{\partial v_i^{(p)}}{\partial \tau} = \frac{\partial^2 v_i^{(p)}}{\partial w^2} + (k_1 - 1 - s_{i,p})\frac{\partial v_i^{(p)}}{\partial w} - k v_i^{(p)} + g_{i,p}(w, \tau), \tag{5}$$

where $g_{i,p}(w, \tau) = f(w - s_{i,p}(\tau - \tau_i) - R_i)$.

Equation (5) together with the boundary conditions $v_i^{(p)}(z_0 - L, \tau_{i+1}) = 1$, $v_i^{(p)}(z_0, \tau_{i+1}) = 0$, and initial value $v_i^{(p)}(w, \tau_i) = \tilde{v}(w, \tau_i)$ for $w \in [z_0 - L, z_0]$, can be discretized using the Crank-Nicolson method obtaining $m+1$ approximation values v_0, v_1,...,v_m of $v_i^{(p)}$, with $v_i^{(p)}(w_j, \tau_{i+1}) \approx v_j$ where $z_0 - L = w_0 < w_1 < w_2 < ... < w_m = z_0$ are $m + 1$ equally spaced numbers on $[z_0 - L, z_0]$. Therefore $u_i^{(p)}(x_j, \tau_{i+1}) \approx v_j$ where $x_j = w_j - s_{i,p}(\tau_{i+1} - \tau_i) - R_i$, $j = 0, ..., m$ are $m + 1$ equally distributed numbers on $[z_{i+1}^{(p)} - L, z_{i+1}^{(p)}]$. The values v_0 and

v_m are given as boundary conditions, $v_0 = 1$, $v_m = 0$, while $v_1, ..., v_{m-1}$ are calculated by solving a tridiagonal linear system of equations $A\vec{v} = \vec{b}$, where $A \in \mathbb{R}^{(m-1)\times(m-1)}$ is defined as

$$A = \begin{pmatrix} \alpha_2 & \alpha_3 & & & & \\ \alpha_1 & \alpha_2 & \alpha_3 & & & \\ & \alpha_1 & \alpha_2 & \alpha_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \alpha_1 & \alpha_2 & \alpha_3 \\ & & & & \alpha_1 & \alpha_2 \end{pmatrix},$$

with

$$\Delta\tau_{i+1} = \tau_{i+1} - \tau_i, \tag{6}$$

$$\Delta w = L/m, \tag{7}$$

$$\alpha_1 = \frac{\Delta\tau_{i+1}}{2(\Delta w)^2} - (k_1 - 1 - s_{i,p})\frac{\Delta\tau_{i+1}}{4\Delta w}, \tag{8}$$

$$\alpha_2 = -\frac{\Delta\tau_{i+1}}{(\Delta w)^2} - k\frac{\Delta\tau_{i+1}}{2} - 1, \tag{9}$$

$$\alpha_3 = \frac{\Delta\tau_{i+1}}{2(\Delta w)^2} + (k_1 - 1 - s_{i,p})\frac{\Delta\tau_{i+1}}{4\Delta w}, \tag{10}$$

$$\beta_1 = -\alpha_1, \tag{11}$$

$$\beta_2 = -\alpha_2 - 2, \tag{12}$$

$$\beta_3 = -\alpha_3. \tag{13}$$

The right hand side $\vec{b} = (b_1, ..., b_{m-1})$ is defined as

$$b_1 = \sum_{j=1}^{3} \beta_j \tilde{v}_{j-1} - \frac{\Delta\tau_{i+1}}{2}(g(w_1, \tau_i) + g_{i,p}(w_1, \tau_{i+1})) - \alpha_1,$$

$$b_{m-1} = \sum_{j=1}^{3} \beta_j \tilde{v}_{m+j-3} - \frac{\Delta\tau_{i+1}}{2}(g_{i,p}(w_{m-1}, \tau_i) + g_{i,p}(w_{m-1}, \tau_{i+1})),$$

$$b_r = \sum_{j=1}^{3} \beta_j \tilde{v}_{r+j-2} - \frac{\Delta\tau_{i+1}}{2}(g_{i,p}(w_{r+j-1}, \tau_i) + g_{i,p}(w_{r+j-1}, \tau_{i+1})),$$

$$\text{for } r = 2, ..., m-2,$$

where $\tilde{v}_j = \tilde{v}(w_j, \tau_i)$. Having approximated u_i^p, $z_{i+1}^{(p+1)}$ is calculated with (3) and backward finite differences. This process is done to calculate $z_{i+1}^{(1)}, ..., z_{i+1}^{(q)}$. We

used the value $q = 7$ in numerical tests. We define $\tilde{z}_{i+1} = z_{i+1}^{(q)}$ as our approximation to z_{i+1}, and define the function $\tilde{v}$ at τ_{i+1} as $\tilde{v}(w, \tau_{i+1}) = v_i^{(q)}(w, \tau_{i+1})$, and set $\tilde{u}(x, \tau_{i+1}) = u_i^p(x, \tau_{i+1})$, $x \in [\tilde{z}_{i+1} - L, \tilde{z}_{i+1}]$ (actually x is restricted to a set of $m+1$ equally distributed points $\tilde{z}_{i+1} - L = x_0 < x_1 < .... < x_m = \tilde{z}_{i+1}$).

In Figure 3 (a) it is shown the final domain where the solution $\tilde{u}$ is approximated; such domain is a union of parallelograms. The domain of function $\tilde{v}$ is a union of rectangles shown in Figure 3 (b). The solution u of the Black-Scholes equation (2) is approximated by $\tilde{u}$ as follows: for each $i = 1, ..., n$, $u(x_j, \tau_i) \approx \tilde{u}(x_j, \tau_i)$ where $\tilde{z}_i - L = x_0 < x_1 < .... < x_m = \tilde{z}_i$, with $x_j = \tilde{z}_i - L + j\frac{L}{m}$, $j = 0, ..., m$. The value of the call $C(S, t)$ is approximated as $C(S_j, t_i) \approx S_j - E + E\tilde{u}(x_j, \tau_i)$, where $S_j = Ee^{x_j}$, x_j defined as before, and $t_i = T - 2\tau_i/\sigma^2$.

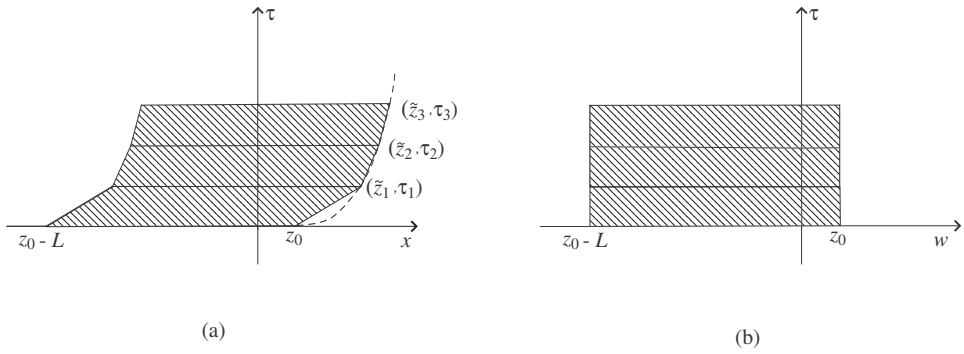

Figure 3: a) Domain of $\tilde{u}$. The dashed curve represents the free boundary. b) Domain of $\tilde{v}$.

4. Numerical Tests

In this section we provide two numerical tests of the algorithm to approximate the free boundary at several times to expiry.

Example 1. In this example we used $\sigma = 0.8$, $r = 0.25$, $D_0 = 0.2$, $E = 1$. The free boundary was calculated at 1 year, 6 months, and 3 months to expiry. The values shown on Table 1 were calculated using 2000 points to discretize the x variable on the interval $[-15, z_0]$, where $z_0 = \ln(r/D_0) = 0.22314355...$ At one year to expiry the variable τ has the value 0.32. We used 100 time steps to calculate the free boundary in each case. The estimated relative error

of approximation was calculated using a larger interval for x (that is, a bigger value of L), a larger number of points in the x variable, and a larger number of time steps. The algorithm calculates the free boundary x_f; the tabulated values of the free boundary correspond to S_f which is given by the relation $S_f = Ee^{x_f}$. On Table 2 are shown the values of the free boundary with a slight change of the algorithm: instead of a second order, it uses a fourth order finite difference discretization of first and second derivatives in the x variable. We used 1000 points to discretize the x variable on the interval $[-15, z_0]$ and 100 time steps in each case.

Example 2. In this example we used $\sigma = 0.2$, $r = 0.1$, $D_0 = 0.05$, $E = 1$. The free boundary was calculated at 1 year, 6 months, and 3 months to expiry. The values shown on Table 3 were calculated using 2000 points to discretize the x variable on the interval $[-15, z_0]$, where $z_0 = \ln(r/D_0) = 0.6931471805...$ At one year to expiry, the variable τ has the value 0.02. We used 100 time steps to calculate the free boundary in each case. On Table 4 are shown the values of the free boundary using a fourth order finite difference discretization in the x variable with 1000 points, and 100 time steps to calculate the free boundary.

Time to expiry	Value of the free boundary S_f	Approximated relative error
1 year	2.8095239	8.4×10^{-6}
6 months	2.4419893	1.0×10^{-5}
3 months	2.1113720	3.0×10^{-5}

Table 1: Approximation of the free boundary S_f of Example 1 using second order discretization in the x variable. We used 2000 points to discretize the x variable on the interval $[-15, z_0]$ and 100 time steps to calculate the free boundary in each case.

Remark 4.1. The numerical scheme proposed in this work can be combined with other methods to deal with the fact that the function $\max(S-E, 0)$ for the initial value of the solution of Equation (1) has a corner at $S = E$ (see [8]). The Rannacher time-stepping with a few initial backward Euler steps (see [7]) can be easily incorporated; for the calculation of the free boundary in our numerical tests it made little difference in the results. The advantage is that it produces a smooth approximation of the call value $C(S, t)$ for $t \in [0, T)$. We will report on future work a more challenging modification of the algorithm so

Time to expiry	Value of the free boundary S_f	Approximated relative error
1 year	2.8095166	5.8×10^{-6}
6 months	2.4419988	6.6×10^{-6}
3 months	2.1114250	5.3×10^{-6}

Table 2: Approximation of the free boundary S_f of Example 1 using fourth order discretization in the x variable. We used 1000 points to discretize the x variable on the interval $[-15, z_0]$ and 100 time steps to calculate the free boundary in each case.

Time to expiry	Value of the free boundary S_f	Approximated relative error
1 year	2.2377075	2.9×10^{-5}
6 months	2.1725228	3.8×10^{-5}
3 months	2.1240239	5.1×10^{-5}

Table 3: Approximation of the free boundary S_f of Example 2 using second order discretization in the x variable. We used 2000 points to discretize the x variable on the interval $[-15, z_0]$ and 100 time steps to calculate the free boundary in each case.

that it includes an initial time-stepping method of higher order than Euler's method (see [2, 3]).

Remark 4.2. Although this method gives a higher precision than other finite difference methods that approximate the free boundary by choosing certain grid points on a fixed mesh, it requires an initial approximation for Newton's method. If the size of Δx is "big", the algorithm may not converge due to the way we choose the initial approximation in Newton's method. Using less than 40 points per unit length in the spatial variable x produce no convergence of the method in our tests. Taking into account that the algorithm is computationally efficient, using more tan 40 points per unit length is not a severe restriction.

Time to expiry	Value of the free boundary S_f	Approximated relative error
1 year	2.23764219	2.3×10^{-7}
6 months	2.17243864	3.5×10^{-7}
3 months	2.12390951	2.3×10^{-6}

Table 4: Approximation of the free boundary S_f of Example 2 using fourth order discretization in the x variable. We used 1000 points to discretize the x variable on the interval $[-15, z_0]$ and 100 time steps to calculate the free boundary in each case.

5. Conclusion

We have presented an algorithm to approximate the free boundary and call value of an American call option given as the solution of the Black-Scholes partial differential equation. By making a well known change of variables, the solution of the equation is sought on a domain bounded from the right by the free boundary, and from the left by a translation of the free boundary. This domain needs to be approximated since the free boundary curve is unknown. We used 7 iterations of Newton's method to construct a sequence of parallelograms that approximate such domain. We used finite differences and the Crank-Nicolson method to solve the Black-Scholes equation with modified boundary values that approximate the original ones on each parallelogram. The computational cost of the algorithm is 7 times the cost of the Crank-Nicolson method which is proportional to $m \times n$, where n is the number of points to discretize the spatial variable, and m is the number of time steps. The accuracy in the approximation of the free boundary is higher than methods based on finite differences that use a fixed mesh. The Rannacher initial time-stepping with some initial backward Euler steps can be easily added to the method; this has the advantage of smoothing out the approximate solution for the call value. On the other hand, the algorithm can be improved if a method of higher order than Euler's method for the initial time-stepping is implemented. We will report on this improvement on future work.

Acknowledgements

This work was supported by Asociación Mexicana de Cultura A.C.

References

[1] F. Black, M. Scholes, The pricing of options and corporate liabilities, *J. Political Economy*, **81** (1973), 637-654.

[2] M.M. Chawla, Generalized trapezoidal formulas for pricing bond options, *Internat. J. Appl. Math.*, **24** (2011), 229-244.

[3] M.M. Chawla, Boundary conditions for the solution of bond pricing equation by finite differences, *Internat. J. Appl. Math.*, **25** (2012), 59-81.

[4] J.C. Cox, M. Rubinstein, *Options Markets*, Prentice Hall, 1985.

[5] J. Crank, *Free and Moving Boundary Problems*, Oxford University Press, 1984.

[6] J.N. Dewynne, S.D. Howison, J. Rupf, P. Wilmott, Some mathematical results in the pricing of American options, *Euro. Jnl. Appl. Math.*, **4** (1993), 381-398.

[7] M.B. Giles, R. Carter, Convergence analysis of Crank-Nicolson and Rannacher time-marching, *Journal of Computational Finance*, **9**, No 4 (2006), 89-112.

[8] D.M. Pooley, P.A. Forsyth, K.R. Vetzal, Convergence remedies for non-smooth payoffs in option pricing, *Journal of Computational Finance*, **6**, No 4 (2003).

[9] D. Ševčovič, Analysis of the free boundary for the pricing of an American call option, *Euro. Jnl. Applied Mathematics*, **12** (2001), 25-37.

[10] J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1993.

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