J. KSIAM Vol.19, No.1, 1–21, 2015 http://dx.doi.org/10.12941/jksiam.2015.19.001

FINITE-DIFFERENCE BISECTION ALGORITHMS FOR FREE BOUNDARIES OF AMERICAN OPTIONS

SUNBU KANG[1], TAEKKEUN KIM[2], AND YONGHOON KWON[3†]

[1]DEPARTMENT OF MATHEMATICS, KOREA AIR FORCE ACADEMY, CHEONGJU 635, KOREA
E-mail address: sbkang@postech.ac.kr

[2]NH INVESTMENT & SECURITIES, YEONGDEUNGPO-GU, SEOUL 150-747, KOREA
E-mail address: kz_nhis@nhis.co.kr

[3]DEPARTMENT OF MATHEMATICS, POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY, POHANG 790-784, KOREA
E-mail address: ykwon@postech.ac.kr

ABSTRACT. This paper presents two algorithms based on the Jamshidian equation which is from the Black-Scholes partial differential equation. The first algorithm is for American call options and the second one is for American put options. They compute numerically free boundary and then option price, iteratively, because the free boundary and the option price are coupled implicitly.

By the upwind finite-difference scheme, we discretize the Jamshidian equation with respect to asset variable s and set up a linear system whose solution is an approximation to the option value. Using the property that the coefficient matrix of this linear system is an M-matrix, we prove several theorems in order to formulate a bisection method, which generates a sequence of intervals converging to the fixed interval containing the free boundary value with error bound h. These algorithms have the accuracy of $O(k + h)$, where k and h are step sizes of variables t and s, respectively. We prove that they are unconditionally stable.

We applied our algorithms for a series of numerical experiments and compared them with other algorithms. Our algorithms are efficient and applicable to options with such constraints as $r > d$, $r \leq d$, long-time or short-time maturity T.

1. INTRODUCTION

The main stream of earlier option contracts was concerned with *European* options, but the option markets nowadays has been related to *American* options. Unlike European option,

---

Received by the editors December 5 2014; Revised March 17 2015; Accepted in revised form March 17 2015; Published online March 18 2015.

2000 *Mathematics Subject Classification.* 65M06, 91B28, 65M12.

Key words and phrases. American call and put options, Black-Scholes equations, free boundary, fixed point, upwind finite-difference scheme, M-matrix, bisection method, unconditional stability.

† Corresponding author. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(NRF-2011-0022508).

American option may be exercised at any moment before its maturity time. Hence at each time t, it is very important to know the option price depending on the asset price s. Additionally, we have to know the optimal asset price s in order to decide whether to exercise the option right.

As an approach to these questions, the Black-Scholes equation has been used widely [2]. Most of the models for pricing American options have been derived from parabolic free boundary problems which are related with the Black-Scholes equations or with variational inequalities [7, 8]. At each time t, let $S(t)$ be the asset price optimal to exercise the option right. The curve $S(t)$ is called *the free boundary* or *the optimal exercise curve* of the option. Option price depends on the time variable t and the underlying asset variable s. Because option price and free boundary are the solutions of implicitly coupled equations, option price can not be computed without knowing the free boundary $S(t)$. Unfortunately, no explicit formula of the free boundary has been obtained yet. In option markets, Binomial Tree methods [3] have been widely used in setting the option price, which is assumed to be independent of the free boundary.

Assume that the underlying asset pays a continuous dividend $d > 0$ with a risk-free interest rate $r > 0$. Let T denote the date of maturity, $\sigma > 0$ the constant volatility of the underlying asset, and E the exercise price. Let $C(t,s)$ be the price of an American call option and $S_c(t)$ its free boundary. Similarly, let $P(t,s)$ be the price of an American put option and $S_p(t)$ be its free boundary.

The Jamshidian equation [5] option offers a particular approach to pricing the values of American options. This equation is a non-homogeneous Black-Scholes equation of a parabolic partial differential equation in the infinite domain $D_\infty = \{(t,s) \,|\, 0 < t < T,\, 0 < s < \infty\}$. Let $H(x)$ be the Heaviside function

$$H(x) = \begin{cases} 0 & \text{for } x < 0, \\ 1 & \text{for } x \geq 0. \end{cases} \tag{1.1}$$

Let $(x)_+ = \max(x, 0)$. It is shown in [5, 6] that the American call-option price $C(t,s)$ with exercise price E satisfies the Jamshidian equation

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 C}{\partial s^2} + (r-d)s\frac{\partial C}{\partial s} - rC = -(ds - rE)_+ \, H(s - S_c(t)) \quad \text{on } D_\infty, \tag{1.2}$$

subject to the terminal condition of the pay-off function

$$C(T,s) = \max(s - E, 0).$$

Similarly, the American put-option price $P(t,s)$ with exercise price E satisfies the Jamshidian equation

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 P}{\partial s^2} + (r-d)s\frac{\partial P}{\partial s} - rP = -(rE - ds)_+ \, H(S_p(t) - s) \quad \text{on } D_\infty, \tag{1.3}$$

subject to the terminal condition of the pay-off function

$$P(T, s) = \max(E - s, 0).$$

A semi-explicit formula for an American call option was presented in Ševčoviç [10]. For numerical approximations to $C(t, s)$, the Jamshidian equation (1.2) was solved explicitly with already somehow [6] known approximations to the free boundary.

Because of the relation of $C(t, s)$ and $S_c(t)$ in the Jamshidian equation (1.2), the option price should be computed in connection with the values of the free boundary, which is the most difficult part in pricing an American option. This problem has been overcome partially by Kholodnyi [6]. In his work, he derived a semilinear Black-Scholes equation for an American option and proved the existence and the uniqueness of the solution.

In this paper, unlike other explicit numerical schemes, Algorithm 3.10 for American call options computes iteravively the implicit solution $(C(t, s), S_c(t))$ of System (2.4)–(2.5). A bisection method implementing fixed-point iterations computes the free boundary $S_c(t)$ and then finite-difference upwind methods computes the option price $C(t, s)$. The free boundary $S_c(t)$ is treated as a fixed point.

This paper is organized as follows. In Section 2, we shall review the Jamshidian equation for American options to derive an implicitly coupled system of the Jamshidian equation and the free boundary equation. The existence and the uniqueness of the solution to this system was analyzed in Kholodnyi [6].

In Section 3, by using the upwind finite-difference scheme, we shall discretize the Jamshidian equation and set up a linear system. The coefficient matrix of this system is an M-matrix. We shall use the theory of M-matrix to derive Theorem 3.9, which generates a sequence of intervals converging to the fixed interval which contains $S_c(t)$ with error bound h. The free boundary $S_c(t)$ is treated as a fixed point. We shall present two main algorithms. Algorithm 3.10 is for American call options and Algorithm 3.11 is for American put options.

In Section 4, we shall report the numerical results computed by our algorithms for a series of American call and put options to compare the algorithms with the method in Ševčoviç [10] and the binomial tree method. We tested various model problems with such as $r > d$, $r \leq d$, long-time and short-time maturities.

In Section 5, we shall make some remarks on the conclusion and discuss about application of our algorithms to some other research area.

2. The Implicit System for Option Price and Free Boundary

In this section, after reviewing the Jamshidian equation [5] which comes from the Black-Scholes equation, we shall derive an implicitly coupled system of the option price and the free boundary.

Let us introduce some properties of the free boundary $S_c(t)$ for an American call-option price $C(t,s)$. Although any explicit formula of $S_c(t)$ has not yet been known, there are several well-known results [1, 10, 12]:

- **(P1)** At the maturity time T, the free boundary $S_c(t)$ is independent of σ and satisfies the condition $S_c(T) = \max(E, \frac{r}{d}E)$.
- **(P2)** The free boundary $S_c(t)$ is a nonincreasing function.
- **(P3)** The free boundary $S_c(t)$ can be defined by $S_c(t) = \inf\{\, s \mid C(t,s) = s - E \quad \text{for } s \geq S_c(T)\}$.
- **(P4)** The free boundary $S_c(t)$ has the lower and upper bounds such that

$$S_c(T) \leq S_c(t) \leq S_u = \frac{\lambda E}{\lambda - 1} \qquad \text{for } t \in [0,T],$$

where

$$\lambda = \frac{\sigma^2/2 - r + d + \sqrt{(\sigma^2/2 - r + d)^2 + 2\sigma^2 r}}{\sigma^2}.$$

By the definition in Property **(P3)**, the free boundary $S_c(t)$ divides the infinite region D_∞ into D_c and D_e as in Figure 1; $D_\infty = D_c \cup D_e$ ([1, 12]), where

$$D_c = \{(t,s) \mid 0 < t < T,\ 0 < s < S_c(t)\},$$
$$D_e = \{(t,s) \mid 0 < t < T,\ S_c(t) \leq s < \infty)\}.$$

We call D_e *the exercise region* where early exercise of option right is optimal and D_c *the continuation region* where it is not optimal.

On the continuation region D_c, the call-option price $C(t,s)$ is greater than the pay-off function and is the solution of the Black–Scholes equation:

$$\begin{cases} C(t,s) > \max(s-E,0), \quad (t,s) \in D_c. \\ \frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 C}{\partial s^2} + (r-d)s\frac{\partial C}{\partial s} - rC = 0, \quad (t,s) \in D_c\,. \end{cases} \tag{2.1}$$

On the exercise region D_e, the call-option price $C(t,s)$ is the pay-off function; $C(t,s) = s - E$. Plugging the pay-off function $s - E$ into $C(t,s)$ of the above equation, we have

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 C}{\partial s^2} + (r-d)s\frac{\partial C}{\partial s} - rC = rE - ds.$$

Hence, we have

$$\begin{cases} C(t,s) = s - E, \quad (t,s) \in D_e, \\ \frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 C}{\partial s^2} + (r-d)s\frac{\partial C}{\partial s} - rC = rE - ds, \quad (t,s) \in D_e. \end{cases} \tag{2.2}$$

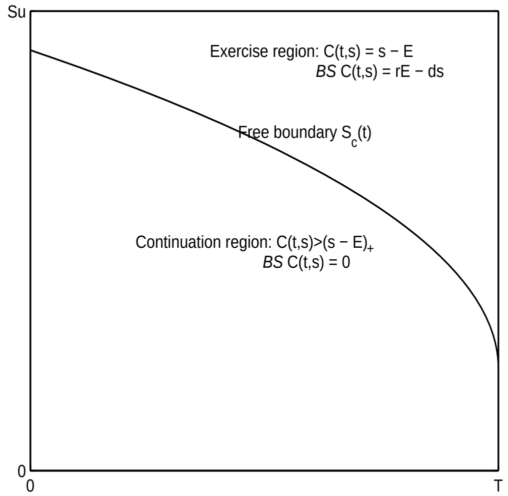

FIGURE 1. Free boundary: continuation region and exercise region.

From Properties **(P1)**–**(P2)**, it follows that for $s \geq S_c(t)$

$$ds - rE \geq dS_c(T) - rE \geq 0. \tag{2.3}$$

Hence,

$$rE - ds = -(ds - rE) \leq 0, \quad (t, s) \in D_e.$$

Using the Heaviside function H in (1.1) and the free boundary $S_c(t)$, we can represent the differential equations in (2.1)–(2.2) by one equation over one region $D_\infty = D_c \cup D_e$:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 C}{\partial s^2} + (r-d)s\frac{\partial C}{\partial s} - rC = -(ds - rE)_+ H(s - S_c(t)), \quad (t, s) \in D_\infty.$$

This equation is called the Jamshidian equation [5, 6]. It can not be solved until the free boundary $S_c(t)$ is given.

For computations we need to restrict the infinite domain D_∞ to a finite domain. Using the upper bound S_u in Property **(P4)**, let us choose a sufficiently large underlying asset value S with $S > S_u$ and take the finite domain

$$D = \{(t, s) \mid 0 < t < T,\ 0 < s < S\} \supset D_c.$$

Since $C(t, s) > s - E$ on D_c, for practical computations we replace the definition of the free boundary in Property **(P3)** by

$$S_c(t) = \inf\{\, s \mid C(t, s) \leq s - E \quad \text{for } s \geq S_c(T)\}, \quad (t, s) \in D,$$

where "$=$" is replaced by "$\leq$".

In this paper, we shall consider the following system with the Jamshidian equation and the free boundary, for $(t,s) \in D$,

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 C}{\partial s^2} + (r-d)s\frac{\partial C}{\partial s} - rC = (rE - ds)H(s - S_c(t)), \tag{2.4}$$

$$S_c(t) = \inf\{\, s \mid C(t,s) \leq s - E \quad \text{for } s \geq S_c(T)\}, \tag{2.5}$$

subject to the terminal and boundary conditions

$$S_c(T) = \max(E, \frac{r}{d}E), \quad C(T,s) = \max(s-E, 0) \quad \text{for } 0 \leq s \leq S, \tag{2.6}$$

$$C(t,0) = 0, \quad C(t,S) = S - E, \quad \text{for } 0 \leq t \leq T. \tag{2.7}$$

Let ϕ be the function defined by

$$\phi(u) = \inf\{\, s \mid C(t,s;u) \leq s - E \quad \text{for } s \geq S_c(T)\}, \tag{2.8}$$

where

$$\frac{\partial C(t,s;u)}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 C(t,s;u)}{\partial s^2} + (r-d)s\frac{\partial C(t,s;u)}{\partial s} - rC(t,s;u) = (rE - ds)H(s-u).$$

Then, the free boundary $S_c(t)$ in (2.4)–(2.5) is a fixed point of ϕ such that $S_c(t) = \phi(S_c(t))$.

Similarly for the American put option, we consider the system coupled with the Jamshidian equation and the free boundary, for $(t,s) \in D$,

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 P}{\partial s^2} + (r-d)s\frac{\partial P}{\partial s} - rP = (ds - rE)H(S_p(t) - s), \tag{2.9}$$

$$S_p(t) = \sup\{\, s \mid P(t,s) \leq E - s \quad \text{for } s \leq S_p(T)\}, \tag{2.10}$$

subject to the terminal and boundary conditions

$$S_p(T) = \min(E, \frac{r}{d}E), \quad P(T,s) = \max(E-s, 0) \quad \text{for } 0 \leq s \leq S, \tag{2.11}$$

$$P(t,0) = E, \quad P(t,S) = 0, \quad \text{for } 0 \leq t \leq T. \tag{2.12}$$

Let ψ be the function defined by

$$\psi(u) = \sup\{\, s \mid P(t,s;u) \leq E - s \quad \text{for } s \leq S_p(T)\}, \tag{2.13}$$

where

$$\frac{\partial P(t,s;u)}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 P(t,s;u)}{\partial s^2} + (r-d)s\frac{\partial P(t,s;u)}{\partial s} - rP(t,s;u) = (ds - rE)H(u-s).$$

Then, the free boundary $S_p(t)$ in (2.9)–(2.10) is a fixed point of ψ such that $S_p(t) = \psi(S_c(t))$.

3. Numerical Algorithms

In this section, we shall present Algorithm 3.10 to solve numerically the implicit system (2.4)–(2.7) of American call options. Also, we shall present Algorithm 3.11 for American put options. These algorithms are unconditionally stable.

Let us choose two positive integers N and M. Let $k = T/N$ and $h = S/M$ as the step sizes of time variable t and underlying asset variable s, respectively. Divide the interval $[0, T]$ into N sub-intervals with the grid points

$$t_j = jk, \quad j = N, N-1, \cdots, 0, \tag{3.1}$$

and the interval $[0, S]$ into M sub-intervals with the grid points

$$s_i = ih, \quad i = 0, 1, \cdots, M. \tag{3.2}$$

Plugging the forward-difference scheme

$$\frac{\partial C(t_j, s_i)}{\partial t} = \frac{C(t_{j+1}, s_i) - C(t_j, s_i)}{k} + O(h),$$

the central-difference scheme

$$\frac{\partial^2 C(t_j, s_i)}{\partial s^2} = \frac{C(t_j, s_{i-1}) - 2C(t_j, s_i) + C(t_j, s_{i+1})}{h^2} + O(h^2),$$

and the up-wind scheme

$$\frac{\partial C(t_j, s_i)}{\partial s} = D^\nu(C(t_j, s_i)) + O(k),$$

where

$$D^\nu(C(t_j, s_i)) = \begin{cases} \dfrac{C(t_j, s_{i+1}) - C(t_j, s_i)}{h} & \text{if } r \geq d, \\ \dfrac{C(t_j, s_i) - C(t_j, s_{i-1})}{h} & \text{if } r < d, \end{cases} \tag{3.3}$$

into the Jamshidian equation (2.4) and neglecting the O-small terms, we approximate the system (2.4)–(2.7) by the following discrete system (3.4)–(3.7) for unknown values $c_i^j \approx C(t_j, s_i)$ and $s_c^j \approx S_c(t_j)$ with already known values c_i^{j+1} and s_c^{j+1}: for $j = N-1, \cdots, 0$ and $i = 1, 2 \cdots, M-1$

$$\frac{c_i^{j+1} - c_i^j}{k} + \frac{1}{2}\sigma^2 s_i^2 \frac{c_{i-1}^j - 2c_i^j + c_{i+1}^j}{h^2} + (r-d)s_i D^\nu(c_i^j) - rc_i^j = (rE - ds_i)H(s_i - s_c^j), \tag{3.4}$$

$$s_c^j = \min\{\, s_i \mid c_i^j \leq s_i - E \quad \text{for } s_i \geq s_c^{j+1}\}, \tag{3.5}$$

with terminal and boundary conditions

$$s_c^N = S_c(t_N) = \max(E, \frac{r}{d}E), \quad c_i^N = \max(s_i - E, 0), \tag{3.6}$$

$$c_0^j = 0, \quad c_M^j = s_M - E, \qquad \text{for } j = N-1, \ldots, 1, 0. \tag{3.7}$$

Since local truncation errors are $O(h^2)$, $O(h)$, and $O(k)$, the finite-difference equation (3.4) is consistent with the Jamshidian equation (2.4) if $\partial^2 C/\partial t^2(t,s)$ and $\partial^4 C/\partial s^4(t,s)$ are continuous.

Collecting the like terms of c_{i-1}^j, c_i^j, and c_{i+1}^j in the left-side and all the other terms in the right-side gives the linear equations, for $i = 1, 2, \ldots, M$,

$$a_{i,i-1}c_{i-1}^j + a_{ii}c_i^j + a_{i+1,i}c_{i+1}^j = c_i^{j+1} + kb_i, \tag{3.8}$$

where

$$\begin{aligned}
a_{i,i-1} &= k\left(-\frac{\sigma^2 i^2}{2} + \min(0, r-d)i\right), \\
a_{ii} &= 1 + k\left(\sigma^2 i^2 + |r-d|i + r\right), \\
a_{i,i+1} &= k\left(-\frac{\sigma^2 i^2}{2} - \max(0, r-d)i\right), \\
b_i &= (ds_i - rE)H(s_i - s_c^j).
\end{aligned}$$

Using the boundary conditions in (3.7), let us add two more equations

$$c_0^j = c_0^{j+1} \quad \text{and} \quad c_M^j = c_M^{j+1}. \tag{3.9}$$

Let $\mathbf{c}^j = (0, c_1^j, \ldots, c_{M-1}^j, s_M - E)^t$ and $\mathbf{b} = (0, b_1, \ldots, b_{M-1}, 0)^t$ be the column vectors such that

$$\mathbf{c}^j = \begin{bmatrix} 0 \\ \vdots \\ c_i^j \\ \vdots \\ s_M - E \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ \vdots \\ (ds_i - rE)H(s_i - s_c^j) \\ \vdots \\ 0 \end{bmatrix}.$$

Using the equations (3.8) and (3.9), we set up the following system (3.10)–(3.11) with unknown value s_c^j and vector $\mathbf{c}^j$: for $j = N-1, N-2, \ldots, 0$

$$\begin{aligned}
A\mathbf{c}^j &= \mathbf{c}^{j+1} + k\mathbf{b}(s_c^j), && (3.10)\\
s_c^j &= \min\{\, s_i \mid c_i^j \le s_i - E \quad \text{for } s_i \ge s_c^{j+1}\}, && (3.11)
\end{aligned}$$

where $A = (a_{ij})$ is the tri-diagonal $(M+1) \times (M+1)$ matrix such that

$$A = \begin{bmatrix} 1 & 0 & & & \\ a_{10} & a_{11} & a_{12} & & \\ & a_{i-1,i} & a_{ii} & a_{i,i+1} & \\ & & a_{M-1,M-2} & a_{M-1,M-1} & a_{M-1,M} \\ & & & 0 & 1 \end{bmatrix},$$

In order to solve the above system for s_c^j and $\mathbf{c}^j$, let us introduce the theory of M-matrix [9]. Let $\mathbf{x} = (x_1, \ldots, x_n)^t$ be a column vector and $L = (l_{ij})$ be an $n \times n$ matrix. Let

$$\mathbf{x} \geq \mathbf{0} \quad \text{if and only if} \quad x_i \geq 0 \quad \text{for all } i,$$

and

$$L \geq \mathbf{0} \quad \text{if and only if} \quad l_{ij} \geq 0 \quad \text{for all } i \text{ and } j.$$

Definition 3.1. *([9]) A matrix L is an M-matrix, if its entries l_{ij} satisfy $l_{ij} \leq 0$ for $i \neq j$ and its inverse L^{-1} exists with $L^{-1} \geq 0$.*

Let us use the vector and matrix norms defined by

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|, \qquad \|L\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |l_{ij}|.$$

The following theorem can be found in [9].

Theorem 3.2. *(M-criterion, [9]) Let the matrix L satisfy $l_{ij} \leq 0$ for $i \neq j$. Then L is an M-matrix if and only if there exists a vector $\mathbf{e} > \mathbf{0}$ such that $L\mathbf{e} > \mathbf{0}$. Furthermore, if L is an M-matrix, then there exists its inverse matrix L^{-1} with*

$$\|L^{-1}\|_\infty \leq \frac{\|\mathbf{e}\|_\infty}{\min_i (L\mathbf{e})_i}.$$

Theorem 3.3. *The matrix A in (3.10) is an M-matrix. Furthermore, there exists its inverse matrix A^{-1} with $\|A^{-1}\|_\infty \leq 1$.*

Proof. Clearly, the tridiagonal matrix A satisfies the condition $a_{ij} \leq 0$ for $i \neq j$. For the column vector $\mathbf{e} = (1, 1, \cdots, 1)^t > \mathbf{0}$, since $r \geq 0$ we have

$$A\mathbf{e} \geq (1, 1 + rk, \ldots, 1 + rk, 1)^t > \mathbf{0}.$$

Hence, by Theorem 3.2 the matrix A is an M-matrix. Furthermore,

$$\|A\|_\infty \leq \frac{\|\mathbf{e}\|_\infty}{\min_i (A\mathbf{e})_i} = \frac{\|\mathbf{e}\|_\infty}{(A\mathbf{e})_1} = 1.$$

This completes the proof. ☐

Since A is nonsingular, the linear system system (3.10) can be written as

$$\mathbf{c}^j = A^{-1}\mathbf{c}^{j+1} + kA^{-1}\mathbf{b}(s_c^j), \qquad j = N-1, N-2, \ldots, 0. \tag{3.12}$$

Theorem 3.4. *The problem to compute* $\mathbf{c}^j$ *by (3.12) is unconditionally stable with respect to the time-marching iteration* j*. This means that the initial error does not grow as* j *increases. The step sizes* k *and* h *can be taken independently.*

Proof. It is enough to prove that the spectral radius $\lambda(A) \le 1$. By Theorem 3.3, the matrix A is an M-matrix with $\|A^{-1}\|_\infty \le 1$. It is well-known that $\lambda(A) \le \|A^{-1}|_\infty$. Hence, $\lambda(A) \le 1$. This completes the proof. □

3.1. **Fixed-Point Iteration and Fixed Interval.** Define a column vector $\mathbf{b}(u)$ with a variable u by

$$\mathbf{b}(u) = \begin{bmatrix} 0 \\ (ds_1 - rE)H(s_1 - u) \\ \vdots \\ (ds_{M-1} - rE)H(s_{M-1} - u) \\ 0 \end{bmatrix}. \tag{3.13}$$

Specifying the dependency on variables $\mathbf{c}^j$ and s_c^j, we rewrite the system (3.10)–(3.11) as

$$\begin{cases} \mathbf{c}^j(s_c^j) = A^{-1}\mathbf{c}^{j+1} + kA^{-1}\mathbf{b}(s_c^j), \\ s_c^j(c^j) = \min\{s_i \mid c_i^j(s_c^j) \le s_i - E \quad \text{for } s_i \ge S_c(T)\}, \end{cases} \tag{3.14}$$

for each $j = N-1, N-2, \cdots, 0$.

Define the function ϕ_h of variable u by

$$\phi_h(u) = \min\{\, s_i \mid c_i^j(u) \le s_i - E \quad \text{for } s_i \ge S_c(T)\}, \tag{3.15}$$

where

$$\mathbf{c}^j(u) = A^{-1}\mathbf{c}^{j+1} + kA^{-1}\mathbf{b}(u) \tag{3.16}$$

is defined with the terminal and boundary conditions in (3.6)–(3.7). Then, the free boundary s_c^j is a fixed point of the function ϕ_h such that $s_c^j = \phi_h(s_c^j)$.

As for American put options, the free boundary s_p^j is a fixed point of the function ψ_h defined by

$$\psi_h(u) = \max\{\, s_i \mid p_i^j(u) \le E - s_i \quad \text{for } s_i \le S_p(T)\}, \tag{3.17}$$

where

$$\mathbf{p}^j(u) = A^{-1}\mathbf{p}^{j+1} + kA^{-1}\mathbf{b}(u) \tag{3.18}$$

is defined with the terminal and boundary conditions in (2.11)–(2.12).

Remark. Note that $\phi_h(u)$ and $\psi_h(u)$ belong to the finite set $\{s_0, s_1, \ldots, s_M\}$.

Theorem 3.5. *For each $u \geq S_c(T)$, the column vector $\mathbf{c}^j(u)$ in (3.16) becomes positive:*

$$\mathbf{c}^j(u) \geq 0, \quad j = N-1, \cdots, 1, 0. \tag{3.19}$$

Proof. By (3.6), we have $\mathbf{c}^N \geq 0$. Suppose that $\mathbf{c}^{j+1} \geq 0$ for $j = N-1, N-2, \ldots, 0$. From (2.3),

$$ds_i - rE \geq 0 \quad \text{if } s_i \geq u \geq S_c(T), \qquad i = 1, 2, \ldots, M-1. \tag{3.20}$$

This gives

$$\begin{aligned} b_i^j(u) &= (ds_i - rE)H(s_i - u) \\ &= \begin{cases} ds_i - rE & \text{if } s_i \geq u, \\ 0 & \text{if } s_i < u, \end{cases} \\ &\geq 0. \end{aligned}$$

Consequently, $\mathbf{b}^j(u) \geq 0$ for $u \geq S_c(T)$. From Theorem 3.3, A is an M-matrix. Hence, $A^{-1} > \mathbf{0}$. Therefore,

$$\mathbf{c}^j(u) = A^{-1}(\mathbf{c}^{j+1} + k\mathbf{b}^j(u)) \geq 0.$$

This complete the proof. □

Theorem 3.6. *(Inverse-Monotone) Let $u_1 \geq S_c(T)$ and $u_2 \geq S_c(T)$ be two numbers. If $u_1 \leq u_2$, then*

$$\mathbf{c}^j(u_1) \geq \mathbf{c}^j(u_2), \quad j = N-1, \cdots, 1, 0. \tag{3.21}$$

Proof. By (2.3), we have $ds_i - rE \geq 0$ for $s_i \geq u_1$. Hence,

$$\begin{aligned} b_i^j(u_1) - b_i^j(u_2) &= (ds_i - rE)\{H(s_i - u_1) - H(s_i - u_2)\} \\ &= \begin{cases} ds_i - rE & \text{if } u_1 \leq s_i < u_2, \\ 0 & \text{otherwise}, \end{cases} \\ &\geq 0, \end{aligned}$$

for each $i = 1, 2, \ldots, M-1$.

Thus, the inequality $A^{-1} \geq 0$ gives

$$\begin{aligned} \mathbf{c}^j(u_1) - \mathbf{c}^j(u_2) &= kA^{-1}\Big(\mathbf{b}^j(u_1) - \mathbf{b}^j(u_2)\Big) \\ &= kA^{-1}\Big(0, b_1^j(u_1) - b_1^j(u_2), \ldots, b_{M-1}^j(u_1) - b_{M-1}^j(u_2), 0\Big)^t \\ &\geq \mathbf{0}, \end{aligned}$$

for each $j = N-1, N-2, \ldots, 0$. This completes the proof. □

Theorem 3.7. *(Fixed Interval Theorem) Let ξ be the fixed point at time t_j of the function ϕ in (2.8). If h and k are sufficiently small, then there exists an interval $[hq, h(q+1)] = [s_q, s_{q+1}]$ which has the properties:*

$$\xi \in [s_q, s_{q+1}] \quad \text{and} \quad \phi_h(u) \in [s_q, s_{q+1}] \quad \text{for each } u \in [s_q, s_{q+1}]. \tag{3.22}$$

This interval $[s_q, s_{q+1}]$ is called the **fixed interval** *of ϕ_h.*

Proof. Since ξ is a fixed point of ϕ, there exists a number $\delta > 0$ such that, for each $\delta_1 < \delta$,

$$\phi(u) \in [\xi - \delta_1, \xi + \delta_1] \quad \text{for each } u \in [\xi - \delta_1, \xi + \delta_1].$$

Note that the component c_i^j of the vector $\mathbf{c}^j$ approximates $C(t_j, s_i)$ with error bound $O(k+h)$. Hence, if k and h are sufficiently small, then there exists a positive number δ_2 with $\delta_2 < \delta_1$ such that

$$\phi_h(u) \in [\xi - \delta_2, \xi + \delta_2] \quad \text{for each } u \in [\xi - \delta_2, \xi + \delta_2].$$

For each h, there exists an unique interval $[s_q, s_{q+1}]$ that contains ξ. Since the interval is contained in $[\xi - \delta_2, \xi + \delta_2]$ for sufficiently small h, we may assume that $\phi_h[s_q, s_{q+1}] \subset [s_q, s_{q+1}]$. This completes the proof. □

Let us define the distance between an interval and a number ζ by

$$\|[a, b] - \zeta\|_\infty = \max_{w \in [a,b]} |w - \eta|.$$

Theorem 3.8. (Bracket Theorem) *Let k and h be sufficiently small and $[s_q, s_{q+1}]$ be the fixed interval of ϕ_h. Then at time t_j, we have*

$$\xi \in [s_q, s_{q+1}] \subset [\min(u, \phi(u)) - h,\ \max(u, \phi(u)) + h] \quad \text{for each } u \geq S_c(T). \tag{3.23}$$

Proof. Since both ξ and $\phi_h(\xi)$ are in the fixed interval $[s_q, s_{q+1}]$,

$$s_q \leq \xi \leq s_{q+1} \qquad \text{and} \qquad s_q \leq \phi_h(\xi) \leq s_{q+1}.$$

(i) If $u \in [s_q, s_{q+1}]$, then both u and $\phi_h(u)$ are in the fixed interval $[s_q, s_{q+1}]$. Hence,

$$[s_q, s_{q+1}] \subset [\min(u, \phi(u)) - h,\ \max(u, \phi(u)) + h].$$

(ii) If $S_c(T) \leq u < s_q$, then $u \leq \xi$. By the inverse-monotone theorem 3.6, we have $\phi_h(\xi) \leq \phi_h(u)$. Thus $s_{q+1} \leq \phi_h(\xi) + h \leq \phi_h(u) + h$. Hence,

$$[s_q, s_{q+1}] \subset [u, \phi_h(u) + h] \subset [u - h,\ \phi(u) + h].$$

(iii) If $s_{q+1} < u$, then $\xi \leq u$. Thus $\phi_h(u) \leq \phi_h(\xi)$. Since $\phi_h(u) - h \leq \phi_h(\xi) - h \leq s_q$,

$$[s_q, s_{q+1}] \subset [\phi(u),\ u + h] \subset [\phi(u) - h,\ u + h].$$

Consequently, we have

$$[s_q, s_{q+1}] \subset [\min(u, \phi(u)) - h,\ \max(u, \phi(u)) + h].$$

This completes the proof. □

3.2. **Bisection Method.** Choose a number u_0 with $S_c(T) \le u_0 \le S$ as an initial approximation to $\xi = S_c^j$ and let $I_0 = [a_0 - h, b_0 + h]$ where

$$a_0 = \min(u_0, \phi_h(u_0)) \quad \text{and} \quad b_0 = \max(u_0, \phi_h(u_0)).$$

Then by (3.23) in Theorem 3.8, $\xi \in [s_q, s_{q+1}] \subset I_0$.

Choose the mid point of I_0 and denote it by $u_0 = (a_0 + b_0)/2$. Again by (3.23) in Theorem 3.8, we have $\xi \in [s_q, s_{q+1}] \subset [\min(u_0, \phi(u_0)) - h,\ \max(u_0, \phi_h(u_0)) + h]$. Hence,

$$\xi \in [s_q, s_{q+1}] \subset ([\min(u_0, \phi(u_0)) - h,\ \max(u_0, \phi_h(u_0)) + h] \cap I_0)$$

If $u_0 < \phi_h(u_0)$, then set $a_1 = u_0$ and $b_1 = b_0$; otherwise, set $a_1 = a_0$ and $b_1 = u_0$. Let $I_1 = [a_1 - h, b_1 + h]$, then $\xi \in [s_q, s_{q+1}] \subset I_1 \subset I_0$. Continuing this process n times, we obtain a sequence of intervals $\{I_0, I_1, I_2, \dots I_n\}$ such that

$$\xi \in [s_q, s_{q+1}] \subset I_n \subset I_{n-1} \subset \cdots \subset I_0, \tag{3.24}$$

where $I_n = [a_n - h, b_n + h]$. Since $b_n - a_n = (b_0 - a_0)/2^n$, we have $|I_n| = |[a_n - h, b_n + h]| = (b_0 - a_0)/2^n + 2h$. So far we have proved the following theorem.

Theorem 3.9. (Bisection Method) *If h and k are sufficiently small, then the intervals in (3.24) have the properties:*

(i) $|I_n| = (b_0 - a_0)/2^n + 2h$ *for each* $n = 1, 2, \dots$.
(ii) *Since* $\xi \in I_n$, $\|I_n - \xi\|_\infty \le |I_n|$ *for each* $n = 1, 2, \dots$.

Now, we formulate the main algorithm 3.10, which implements the bisection method in Theorem 3.9 to produce numerical values of s_c^j and $\mathbf{c}^j$, where $s_c^j \approx S_c(t_j)$ and $c_i^j \approx C(t_j, s_i)$.

Algoritm 3.10. (Americal call option)

```
for j = N − 1, ··· , 1, 0
    Choose an initial approximation u_0 with S_c(T) ≤ u_0 ≤ S;
    Compute c^j(u_0);
    Compute φ_h(u_0);
    u_a := min(u_0, φ_h(u_0));
    u_b := max(u_0, φ_h(u_0));
    while (u_b − u_a ≥ ε)
        u_c := (u_a + u_b)/2;
        Compute c^j(u_c);
        Compute φ_h(u_c);
```

if $(u_c < \phi_h(u_c))$
$u_a := u_c$;
else
$u_b := u_c$;
end
$u_c := (u_a + u_b)/2$;
$s_c^j := u_c$;
Compute $\mathbf{c}^j(u_c)$;
end

Since the free boundary $S_c(t)$ is a nondecreasing function of t, $s_c^{j+1} \geq S_c(T)$. Thus at time t_j, the already computed value s_c^{j+1} is recommended as a good initial approximation u_0.

For American put options, we formulate the following algorithm:

Algoritm 3.11. (American put option)
for $j = N-1, \cdots, 1, 0$
Choose an initial value u_0 *with* $0 < u_0 \leq S_p(T)$;
Compute $\mathbf{p}^j(u_0)$;
Compute $\psi_h(u_0)$;
$u_a := \min(u_0, \psi_h(u_0))$;
$u_b := \max(u_0, \psi_h(u_0))$;
while $(u_b - u_a \geq \varepsilon)$
$u_c := (u_a + u_b)/2$;
Compute $\mathbf{p}^j(u_c)$;
Compute $\psi_h(u_c)$;
if $(u_c < \psi_h(u_c))$
$u_a := u_c$;
else
$u_b := u_c$;
end
$u_c := (u_a + u_b)/2$;
$s_p(t_j) := u_c$;
Compute $\mathbf{p}^j(u_c)$;
end

4. Numerical Experiments

In this section, we tested Algorithm 3.10 and Algorithm 3.11 for American options with various parameters as well as with short-time maturity T and long-time maturity T. We shall report the numerical outputs computed by these Algorithms and compare them with the outputs computed by other methods. All numerical computations were done with C-Language. Most of the comparisons were done together with Binomial Tree method, which is popular for pricing American options.

Using the outputs done with American call options, we present Tables 1–4 and Figures 2–5. And also, with American put option, we present Table 5 and Figure 6. In Figures 2–6, all graphs have been plotted by linear interpolations obtained with numerical values on the grid points.

4.1. **Numerical Results for American Call Option.** In Table 1, in the upper block we duplicate the numerical outputs out of [10], where computations were done for parameters $\sigma = 0.2$, $r = 0.1$, $d = 0.05$, $E = 10$, and the maturity $T = 1$. The free boundary value $S_c(0) = 22.3754$ was reported there. To compare Algorithm 3.10 with the algorithm in [10] fairly, we took the same parameters σ, r, d, E, and T. With the numbers $S = 25$, $M = 200$, $N = 200$, and $\varepsilon = 0.5h$, we tested Algorithm 3.10 and got the numerical value $s_c^0 = 22.4401$ of the free boundary $S_c(0)$. In the lower block, we present the numerical outputs computed by Algorithm 3.10 and the output done by Binomial Tree method [4] with 100 depth of the tree.

In Figure 2, we plot the graphs of the output computed by Algorithm 3.10, with numerical values of free boundary $S_c(t)$ in the left-side, and with numerical values of option price $C(0, s)$ in the right-side. We plot together four graphs of free boundary obtained with $M = N = 200$, 400, 800, and 1600. For $M = N = 1600$ we had the numerical value $s_c(0) = 22.3833$ of the free boundary $S_c(0)$.

TABLE 1. An American call option: numerical values of $C(0, s)$.

method \ s	15	18	20	21	22.3754
Method of [10] (Table 1, [10])	5.15	8.09	10.03	11.01	12.37
Trinomial Tree (Table 1, [10])	5.15	8.09	10.03	11.01	12.37
Other Finite Difference (Table 1, [10])	5.49	8.48	10.48	11.48	12.48
Analytic Approximation (Table 1, [10])	5.23	8.10	10.04	11.02	12.38
method \ s	15	18	20	21	22.4401
Algorithm 3.10	5.2316	8.0936	10.0304	11.0106	12.4399
Binomial Tree	5.2308	8.0932	10.0301	11.0105	12.4401

Comparing the numerical values in Table 1, we see that the outputs of our Algorithm 3.10 are likely best among all the methods for pricing the American call option. Since the algorithm uses tridiagonal systems, it is fast and accurate. We see that the values $C(0, s)$ done by Algorithm 3.10 are as accurate as those by Binary Tree.

In Figure 2, the graphs of the free boundary are non-increasing. They have staircases, but seem to converge with the expected $O(k + h)$-error bound. For large asset variable $s > E$, the graph of the price $C(0, s)$ is sufficiently close to that of the pay-off function $\max(s - E, 0)$.

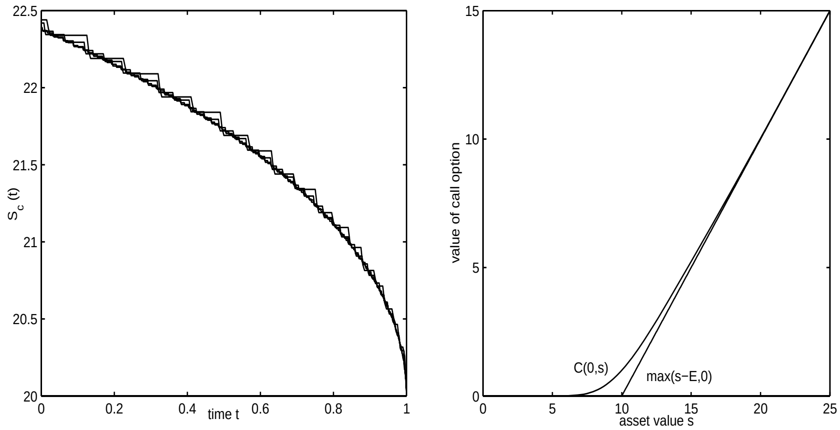

FIGURE 2. An American call option: numerical values of $S_c(t)$ and of $C(0, s)$ by Algorithm 3.10.

4.2. **Numerical Results for Long-time American Call Option.** We tested Algorithm 3.10 for a long-time American call option, with the numbers $T = 100$, $S = 30$, $M = 30000$, $N = 40000$, and $\varepsilon = 0.5h$. We took the parameters $\sigma = 0.2$, $r = 0.1$, $d = 0.05$ and $E = 10$. Property **(P4)** produced the upper bound $S_u = 26.4339$ of the free boundary $S_c(t)$. On the other hand, Algorithm 3.10 produced the numerical value $s_c^0 = 26.4346$ of the free boundary $S_c(0)$.

In Table 2, Algorithm 3.10 produced the numerical values of $C(0, s)$ as accurately as Binomial Tree method did with 5000 depth of the tree. Algorithm 3.10 seems to be reliable and stable for the numerical computations a long maturity T.

In Figure 3, Algorithm 3.10 plots graphs for numerical values for the free boundary $S_c(t)$ and the option price $C(0, s)$. The graph of $S_c(t)$ is non-increasing. For large asset variable $s > E$, the graph of $C(0, s)$ is sufficiently close to that of the pay-off function. Algorithm 3.10 behave correctly for the long-time American call option.

TABLE 2. A long-time American call option: numerical values of $C(0, s)$.

method \ s	15	18	21	24	26.4346
Algorithm 3.10	6.6061	8.8573	11.3497	14.0690	16.4346
Binomial Tree	6.6042	8.8551	11.3484	14.0675	16.4346

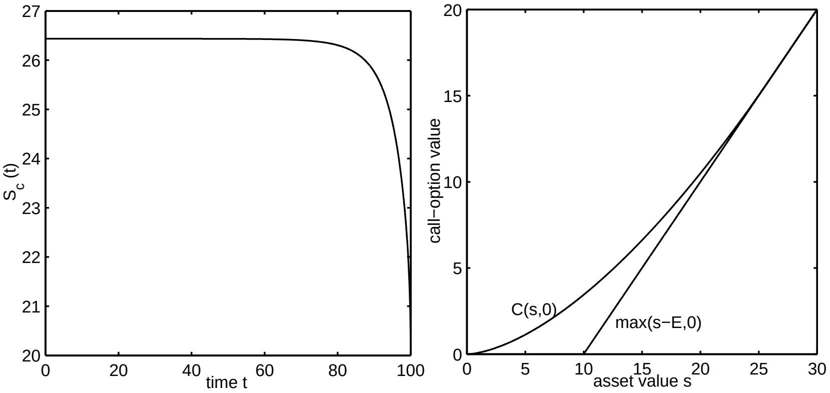

FIGURE 3. A long-time American call option: numerical values of $S_c(t)$ and of $C(0, s)$ by Algorithm 3.10.

4.3. **Numerical Results for Short-time American Call Option.** We tested Algorithm 3.10 for a short-time American call option with numbers $T = \frac{1}{365}$, $S = 25$, $M = N = 5000$, $\varepsilon = 0.5h$. The short-time behavior of $S_c(t)$ for $r > d$ is known in [12] as

$$S_c(t) \approx \frac{rE}{d}\left(1 + \xi_0\sqrt{\frac{1}{2}\sigma^2(T-t)} + \cdots\right) \quad \text{for } \xi_0 = 0.9034. \tag{4.1}$$

We took parameters $\sigma = 0.2$, $r = 0.1$, $d = 0.05$, and $E = 10$. Algorithm 3.10 produced the numerical value $s_c^0 = 20.1336$ of $S_c(0)$, while Approximation (4.1) did $S_c(0) = 20.1337$.

In Table 3, Algorithm 3.10 produced numerical values of $C(0, s)$ as accurately as Binomial Tree method did with 5000 depth of the tree.

In Figure 4, the graph of $S_c(t)$ obtained by Algorithm 3.10 is non-increasing and close within the accuracy to the graph done by Approximation (4.1). Algorithm 3.10 behaves correctly for the short-time American call option.

TABLE 3. A short-time American call option: numerical values of $C(0, s)$.

method $\backslash$ s	10	12	15	18	20.1336
Algorithm 3.10	0.04244	2.00110	5.00069	8.00027	10.1336
Binomial Tree	0.04244	2.00110	5.00068	8.00027	10.1336

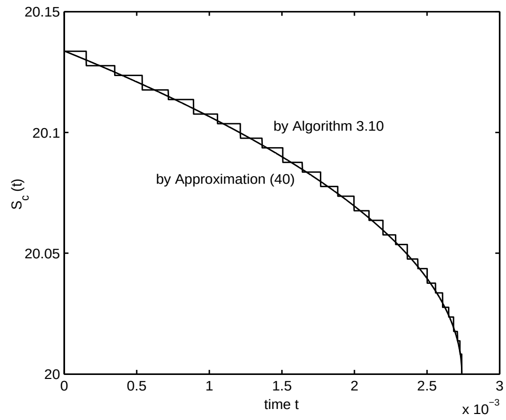

FIGURE 4. A short-time American call option: numerical values of $S_c(t)$ by Algorithm 3.10 and Approximation (4.1)

4.4. **Numerical Results for American Call Option with** $r < d$**.** We tested Algorithm 3.10 for an American call option with the parameters $\sigma = 0.45$, $r = 0.05$, $d = 0.1$, and $E = 10$. In this case $r < d$. We took the numbers $T = 1$, $S = 20$, $M = N = 2000$, and $\varepsilon = 0.5h$. Algorithm 3.10 produced the numerical value $s_c^0 = 17.5073$ of the free boundary $S_c(0)$.

In Table 4, Algorithm 3.10 produced numerical values of $C(0, s)$ as accurate as Binomial Tree method with 1000 depth of the tree.

In Figure 5, the numerical values of $S_c(t)$ and $C(0, s)$ computed by Algorithm 3.10 behave correctly for the American call option with $r < d$

TABLE 4. An American call option for $r < d$: numerical values of $C(0, s)$.

method \ s	8	10	12	15	17	17.5073
Algorithm 3.10	0.6393	1.5085	2.7441	5.1372	7.0052	7.5073
Binomial Tree	0.6392	1.5082	2.7443	5.1372	7.0052	7.5073

4.5. **Numerical Results for American Put Option.** Finally, we tested Algorithm 3.11 for the American put option in the case $r > d$. We took parameters $\sigma = 0.35$, $r = 0.07$, $d = 0.01$ $(r > d)$, $E = 10$, and numbers $T = 1$, $S = 30.0$, $M = N = 3000$, $\varepsilon = 0.5h$. Algorithm 3.11 produced the numerical value $s_p^0 = 6.6048$ of the free boundary $S_p(0)$.

In Table 5, the numerical values of $P(0, s)$ computed by Algorithm 3.11 are as accurate as those computed by Binomial Tree method with 1000 depth of the tree.

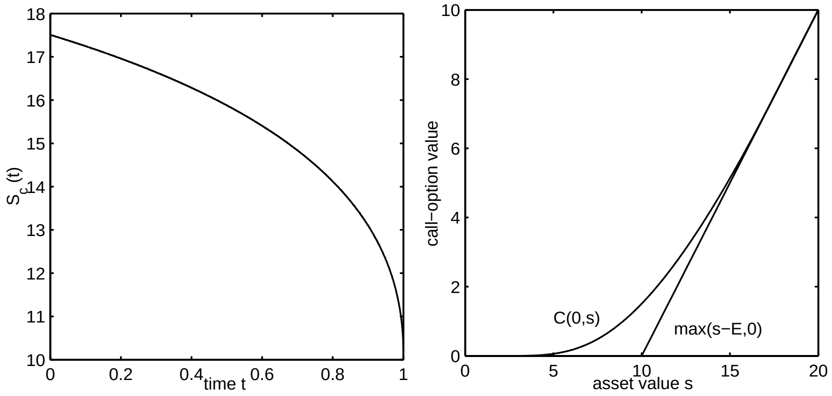

FIGURE 5. An American call option for $r < d$: numerical values of $S_c(t)$ and of $C(0, s)$ by Algorithm 3.10.

In Figure 6, the graph of numerical values of $S_p(t)$ is non-decreasing. when s is small with $s < E$, the graph of numerical values of $P(0, s)$ is close to that of the pay-off function $\max(E - s, 0)$. Algorithm 3.11 behaves correctly for the American put option with $r > d$.

TABLE 5. An American put option for $r > d$: numerical values of $P(0, s)$ by Algorithm 3.11.

method \ s	6.6048	7	9	10	11	12
Algorithm 3.11	3.3952	3.0183	1.5967	1.1347	0.7968	0.5542
Binomial Tree	3.3952	3.0182	1.5966	1.1344	0.7968	0.5542

5. CONCLUSIONS

In this paper, we present two numerical algorithms. Algorithm 3.10 is for American call options and Algorithm 3.11 is for American put option. Those algorithms compute numerical values of option price and free boundary. We showed that the system is unconditionally stable. The option-price values are computed by the linear system derived by an upwind finite-difference scheme to the Jamshidian equation (2.4), with $O(k + h)$-error bound. The free boundary values related with the option price are computed by a bisection method, which generates a sequence of intervals converging to the fixed interval containing the free boundary value within the error h.

Algorithms 3.10 is applicable to American call options not only with a wide range of parameters r and d but also with short-time and long-time maturities. It produced numerical values

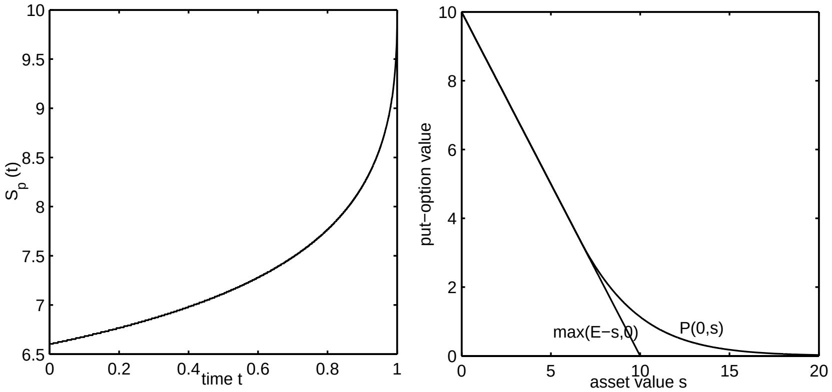

FIGURE 6. An American put option for $r > d$: numerical values of $S_p(t)$ and of $P(0, s)$ by Algorithm 3.11.

of option price and free boundary as accurately and efficiently as Binomial Tree method did. Also, Algorithm 3.11 for an American put option showed the same performance.

Both algorithms are applicable to the volatility depending on time variable t and asset-price variable s. Hence, these algorithms can be used in estimating the local volatility with market data of option prices.

REFERENCES

[1] L. BADEA AND J. WANG, *A new formulation for the valuation of American options, I: Solution uniqueness*, Analysis and Scientific Computing, Eun-Jae Park and Jongwoo Lee, eds., (2000).

[2] F. BLACK AND M. SCHOLES, *The pricing of options and corporate liabilities*, Journal of Political Economy, 81 (1973), pp. 637–654.

[3] J. C. COX, S. A. POSS, AND M. RUBINSTEIN, *Option Pricing: A Simplified Approach*, Journal of Financial Economics, 7 (1979), pp. 229–264.

[4] J. C. HULL, *Introductioin to Futures and Options Markets (3rd edn.)*, Prentice-Hall, New Jersey, 1998.

[5] F. JAMSHIDIAN, *An Analysis of American Options*, Review of Futures Markets, 11 (1992), pp. 72–80.

[6] V. A. KHOLODNYI, 1997 *A Nonlinear partial differential equation for American options in the entire domain of the state variable*, Nonlinear Analysis, Theory, Methods and Applications, 30 (1997), pp. 5059–5070.

[7] D. LAMBERTON AND B. LAPEYRE, *Introduction to Stochastic Calculus Applied to Finance*, Chapman and Hall, New York 1996.

[8] R. C. MERTON, *Theory of rational option pricing*, Bell Journal of Economics and Management Science, 4 (1973), pp. 141–183.

[9] H. G. ROOS, M. STYNES, AND L. TOBISKA, *Numerical Methods for Singularly Perturbed Differential Equations*, Springer-Verlag, Berlin, 1996.

[10] D. ŠEVČOVIÇ, *Analysis of the free boundary for the pricing of an American call option*, European Journal of Applied Mathematics, 12 (2001), pp. 25–37.

[11] G. D. SMITH, *Numerical solution of partial differential equations - finite difference methods (3rd edn.)*, Oxford University Press, New York, 1985.

[12] P. WILMOTT, J. DEWYNNE, AND S. HOWISION, *Option Pricing: Mathematical Models and Computation*, Oxford Financial Press, Oxford, 1995.