

MODULUS-BASED SUCCESSIVE OVERRELAXATION METHOD FOR PRICING AMERICAN OPTIONS[†]

NING ZHENG AND JUN-FENG YIN*

ABSTRACT. We consider the modulus-based successive overrelaxation method for the linear complementarity problems from the discretization of Black-Scholes American options model. The H_+ -matrix property of the system matrix discretized from American option pricing which guarantees the convergence of the proposed method for the linear complementarity problem is analyzed. Numerical experiments confirm the theoretical analysis, and further show that the modulus-based successive overrelaxation method is superior to the classical projected successive overrelaxation method with optimal parameter.

AMS Mathematics Subject Classification : 65M06, 65M12, 65Y20, 90C33, 91B28.

Key words and phrases : American option, Black-Scholes model, linear complementarity problem, H_+ -matrix, modulus-based successive overrelaxation, projected successive overrelaxation.

1. Introduction

The trading of options has grown to tremendous scale, since the Chicago Board Options Exchange (CBOE) started to operate in 1973. The valuation of option contracts has been topic of active research and various type of mathematical models for the prices of different kinds of options are proposed during the last decades. One of the most famous models for the price of an option is based on the Black-Scholes partial differential equation introduced by F. Black and M. Scholes in 1973 [3]. For European options, it is possible to derive the analytical formulas for their prices directly. However, for American option, a so-called early exercise constraint is posed in order to avoid arbitrage opportunities since the option can be exercised anytime before the expiry. Consequently, American

Received October 17, 2012. Revised November 23, 2012. Accepted December 1, 2012.

*Corresponding author. [†]This work was supported by the National Natural Science Foundation of China (No. 11271289 and No.U1135003).

© 2013 Korean SIGCAM and KSCAM.

options pricing lead to free boundary problems in mathematics and numerical methods are usually required [1, 10, 17, 18].

In general, numerical solution of American option models consists two tasks including the discretization of the underlying partial differential equation and the solution of the linear complementarity problem (LCP). A number of discretization methods were considered and discussed in the past decades. For example, Brennan and Schwartz [4] proposed a projected direct method with finite difference discretization for pricing American options. Finite element method was presented in [1] for the discretization of Black-Scholes model. Moreover, finite volume method and penalty methods were considered by Forsyth and Vetzal in [7]. See also [10, 17, 18] for details. In this work, we take the accurate and stable finite difference schemes for use.

For the solution of large sparse LCP, one of the favorite approaches is the projected successive overrelaxation iterative method (PSOR) [5], which was also widely used for pricing American options [1, 11]. One drawback of these projected methods is that the projection of the iterated solution is required, which may be costly and complicated in actual implementations. In addition, by using an equivalent fixed-point equation, van Bokhoven [19] presented a modulus method for solving the LCP through solving a system of linear equations at each iteration, see also Section 9.2 in Murty [15]. Based on the idea of shifted splitting techniques, Hadjidimos [8] and Dong [6] accelerated the original modulus algorithms and studied their convergence conditions when the system matrix is symmetric positive definite. Recently, by utilizing the idea of matrix splittings to construct feasible and efficient methods, modulus-based matrix splitting iteration methods were proposed in [2], and its convergence theory was established when the system matrix is either a positive definite matrix or an H_+ -matrix.

In this paper, the space and time discretization is performed using central finite difference schemes and the L-stable Rannacher method, respectively. Moreover, we consider modulus methods, especially modulus-based successive overrelaxation method (MSOR) for the solution of time-dependent LCPs from American option pricing, which have not appeared in the literature. The H_+ -matrix property of the system matrix discretized from Black-Scholes American option model which guarantees the convergence of the proposed method for the linear complementarity problem is analyzed in the paper. Numerical experiments further verify the convergence and show that MSOR method is superior to PSOR method when optimal parameter is chosen.

This paper is organized as follows. The Black-Scholes American option model and the finite difference scheme are briefly reviewed in Section 2. In Section 3, after reviewing projected method for the linear complementarity problem, we discuss modulus method, especially modulus-based successive overrelaxation method for pricing American options. The H_+ -matrix property of the system matrix discretized from Black-Scholes American option model is shown in Section 4. Numerical experiments are presented in Section 5 and finally in Section 6, some brief concluding remarks are given.

2. Black-Scholes American Option Model

In this section, we briefly review Black-Scholes partial differential equation model for pricing American options and the finite difference discretization method.

An American call (put) option gives a right to buy (sell) the underlying asset for the exercise price K any time before expiry. Denote $u(x, t)$ be the price of the option with respect to the underlying asset value x and time t . The value of the option is obtained by solving the complementarity problem

$$\mathcal{L}u \geq 0, \quad u \geq g \quad \text{and} \quad \mathcal{L}u(u - g) = 0 \tag{1}$$

where the Black-Scholes partial differential operator \mathcal{L} is defined as

$$\mathcal{L} := -\frac{\partial}{\partial t} - \frac{1}{2}\sigma^2x^2\frac{\partial^2}{\partial x^2} - rx\frac{\partial}{\partial x} + r, \tag{2}$$

where $(x, t) \in [0, +\infty] \times [0, T]$, r is the risk free interest rate and σ is the constant volatility. For the call option, the boundary conditions are

$$u(0, t) = 0 \quad \text{and} \quad \lim_{x \rightarrow +\infty} u(x, t) \sim x. \tag{3}$$

For the put option, the boundary conditions are

$$u(0, t) = K \quad \text{and} \quad \lim_{x \rightarrow +\infty} u(x, t) = 0. \tag{4}$$

The final condition is

$$u(x, T) = g(x),$$

where payoff function $g(x)$ gives the price at maturity, which is defined as

$$g(x) = \max(x - K, 0) \tag{5}$$

for the call option and

$$g(x) = \max(K - x, 0) \tag{6}$$

for the put option. Due to the early exercise possibility of American options, the price $u(x, t)$ should be larger than the payoff function $g(x)$ in order to avoid arbitrage possibilities, which leads to an additional constraint

$$u(x, t) \geq g(x). \tag{7}$$

Note that the original option pricing problem is a final value problem, since the value of the option is known at the expiry. For convenience, it is usually transformed into a initial value problem by $\tau = T - t$, see [12, 17].

It is well known [9] that there is a curve $S_f(t)$ which divides the domain $(0, X)$ into two subdomains $(0, S_f(t))$ and $(S_f(t), X)$, where the price of the option equals to the payoff function in one of these subdomains while is higher than the payoff function in the other one. In the region where the constraint (7) is inactive the price satisfies Black-Scholes partial differential equation. However, the place where the constraint is active is not known priori, that is to say, the function $S_f(t)$ is not known beforehand and it has to be found together with the price of the option. Hence the option pricing problem is a free boundary

problem. The underlying asset value $S_f(t)$ indicates the time when the option should be exercised.

In the following, finite difference methods are considered for the space and time discretization of American put option pricing problem (1). The American call option pricing problem could be handled in the similar way.

2.1. Discretization of the Black-Scholes Equation. A numerical solution of the American put option pricing problem (1) requires the discretization of the Black-Scholes partial differential equation. Here, option pricing problems are posed in an infinite region $[0, +\infty) \times [0, T]$ with Dirichlet boundary conditions and a final condition. In order to discretize these problems with finite difference methods, we reformulate problems in a truncated region $[0, X] \times [0, T]$. The truncation point X is sufficiently large in order to avoid excessive error due to the truncation. On the other hand, unnecessarily large value of X will increase computational cost; see details in [13] for the practical choice of X .

An uniform grid is applied in the computational domain $[0, X] \times [0, T]$. Let m and n be the number of grid steps in the x -direction and that in the t -direction, respectively. Furthermore, the grid point values of the finite difference approximation are denoted by

$$u_{ij} \approx u(x_i, t_j) = u(ih, j\Delta\tau), \quad (8)$$

where $h := X/m$, $\Delta\tau := T/n$, $i = 0, \dots, m$ and $j = 0, \dots, n$.

The second-order central finite differences

$$\frac{\partial u}{\partial x}(x_i, t_j) \approx \frac{u_{i+1,j} - u_{i-1,j}}{2h} \quad \text{and} \quad \frac{\partial^2 u}{\partial x^2}(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \quad (9)$$

are applied for the approximation of the space derivatives of the Black-Scholes partial differential equation. The space discretization leads to a semi-discrete equation which has the matrix representation

$$\frac{\partial u}{\partial t} + Su^{(j)} = f, \quad (10)$$

where $S = \text{tridiag}\{a_i, b_i, c_i\}$ with

$$\begin{cases} a_i = -\frac{1}{2}(\sigma^2 i^2 - ri), \\ b_i = \sigma^2 i^2 + r, \\ c_i = -\frac{1}{2}(\sigma^2 i^2 + ri). \end{cases}$$

and $u^{(j)} = [u_{1,j}, u_{2,j}, \dots, u_{m,j}]^T$.

In the following, the time discretization of the semi-discrete equation (10) is discussed. The stability property of time discretization scheme is essential in the option pricing problems as the initial value (6) has discontinuous first derivative. For example, the popular Crank-Nicolson does not have good damping properties and it will lead to numerical solutions with excessive oscillations. Instead of it we employ the second-order accurate and L-stable Rannacher scheme [16].

In the Rannacher time-stepping scheme a few first time steps are performed with the implicit Euler method and then the Crank-Nicolson method is used

$$\left(\frac{1}{\Delta\tau}I + \theta S\right)u^{(j+1)} = \left(\frac{1}{\Delta\tau}I - (1 - \theta)S\right)u^{(j)} + f, \tag{11}$$

where

$$\theta = \begin{cases} 1, & j = 0, 1, 2, 3 \\ \frac{1}{2}, & j = 4, 5, \dots, n - 1 \end{cases}$$

Let

$$B = \frac{1}{\Delta\tau}I + \theta S \quad \text{and} \quad C = \frac{1}{\Delta\tau}I - (1 - \theta)S$$

The space and time discretization of the Black-Scholes model (1) leads to a sequence of linear complementarity problems

$$\begin{cases} Bu^{(j+1)} \geq Cu^{(j)} + f, \\ u^{(j+1)} \geq g, \\ (Bu^{(j+1)} - Cu^{(j)} - f)^T(u^{(j+1)} - g) = 0, \end{cases} \tag{12}$$

which is required to be solved at each time step j , where $j = 0, 1, \dots, n - 1$.

The linear complementarity problem (12) is usually transformed into the standard form where the early exercise constraint equals to zero instead of the payoff function for convenience. Let

$$z := u^{(j+1)} - g, \quad A := B \quad \text{and} \quad q := Bg - Cu^{(j)} - f,$$

the linear complementarity problem (12) can be reformulated into the standard linear complementarity problem

$$w := Az + q \geq 0, \quad z \geq 0 \quad \text{and} \quad z^T w = 0, \tag{13}$$

which is abbreviated as LCP(q, A).

3. The Solutions of the Linear Complementarity Problem

In this section, we consider numerical methods for the linear complementarity problem (13). We first briefly review the classical projected methods for the linear complementarity problem and then discuss modulus methods, especially modulus-based matrix splitting methods in detail.

3.1. Projected Methods. The projected method is a classical method for solving the linear complementarity problem [5]. The following theorem indicates that LCP(q, A) is equivalent to a fixed-point problem.

Theorem 3.1. *Let $A \in \mathbb{R}^{n \times n}$. Then, the LCP(q, A) (13) is equivalent to the following fixed-point problem*

$$(z - (Az + q))_+ - z = 0, \tag{14}$$

We omit the proof here. For a more general results, we refer the reader to [14].

Based on the above equivalence, the **projected method** is then defined as an iterative method for the solution of the fixed-point problem (14). Let $A = D - L - U$, where D , L and U are the diagonal, the strictly lower-triangular and the strictly upper triangular matrices of A , respectively. The projected iterative method for solving the linear complementarity problem (13) can be introduced as follows.

Method 3.2 ([14]). **Projected Successive Overrelaxation (PSOR)**

Given $z^0 \geq 0$,

$$z^{k+1} = (z^k - \omega D^{-1}(Az^k + q - L(z^{k+1} - z^k)))_+, \quad (15)$$

where $k = 0, 1, 2, \dots$ and

$$0 < \omega < 2.$$

If $\omega = 1$, Method 3.1 leads to the **Projected Gauss-Seidel (PGS)** method.

Remark 3.1. The PSOR method adds a projection to the original SOR method at each iteration step. Recall that the SOR iteration scheme for the linear system of equations $Az + q = 0$ is

$$z^{k+1} = z^k - \omega D^{-1}(Az^k + q - L(z^{k+1} - z^k)), \quad (16)$$

$$\Leftrightarrow (D - \omega L)z^{k+1} = [(1 - \omega)D + \omega U]z^k - \omega q, \quad (17)$$

where (16) is similar to (15) regardless of projection.

The algorithm of projected successive overrelaxation method is described as follows.

Algorithm 3.3. Projected Successive Overrelaxation Method

```

1 .   Choose  $m, \omega, tol, maxit$ 
2 .   For  $it = 1, 2, \dots, maxit$ 
3 .     For  $i = 1, \dots, m$ 
4 .        $k = [1, 2, \dots, m]$ ;
5 .        $z(i) = z(i) + \omega(b(i) - A(i, k)z(k))/A(i, i)$ ;
6 .        $z(i) = \max(z(i), 0)$ ;
7 .     End For
8 .     Res =  $\| \min(Az + q, z) \|_2$ ;
9 .     If Res <  $tol$ 
10 .       break;
11 .     End
12 .   End For

```

It is obvious that projected iterative methods can take the advantage of sparse computing, when the coefficient matrix A in linear complementarity problem is large and sparse. However, the convergence rate of the projected successive overrelaxation iteration deteriorates when the number of discretization grids

points increases. On the other hand, the choice of the relaxation parameter has a significant impact on the convergence rate; for the details, see [12]. For the convergence of these projected methods, we refer the reader to [14, 15].

3.2. Modulus Methods. The following theorem indicates that $LCP(q, A)$ is equivalent to another fixed-point problem.

Theorem 3.4 ([15]). *Let the matrix $I + A$ is nonsingular. Then the $LCP(q, A)$ (13) is equivalent to the following fixed-point problem, i.e. finding an $x \in \mathbb{R}^n$ such that*

$$x = (I + A)^{-1}(I - A)|x| - (I + A)^{-1}q \tag{18}$$

in the following sense:

- (1) if x is a solution of (18), then

$$w := |x| - x \quad \text{and} \quad z := |x| + x$$

defines a solution pair for $LCP(q, A)$ (13).

- (2) if w and z solve problem (13), then

$$x := \frac{1}{2}(z - w)$$

is a solution of the fixed-point problem (18).

Based on the above equivalence, the **modulus method** can also be defined as another iterative method for the solution of the fixed-point problem (14). By generalizing the fixed-point problem (18) with the introduction of an iteration parameter, Hadjidimos [8] and Dong [6] modified the modulus method. Moreover, Bai [2] established a class of modulus-based matrix splitting iteration methods, which included the following modulus iterative method.

Method 3.5 ([2]). **Modulus-based Successive Overrelaxation (MSOR)**
Given x^0 ,

$$(D + \Omega - \alpha L)x^{k+1} = [(1 - \alpha)D + \alpha U]x^k + (\Omega - \alpha A)|x^k| - \alpha q, \tag{19}$$

where Ω is a positive diagonal matrix and $k = 0, 1, 2, \dots$ with

$$z^{k+1} = |x^{k+1}| + x^{k+1} \quad \text{and} \quad w^{k+1} = \Omega(|x^{k+1}| - x^{k+1})$$

In particular, if $\alpha = 1$, MSOR method leads to **Modulus-based Gauss-Seidel (MGS)** method.

The algorithm of modulus-based successive overrelaxation method is described as follows.

Algorithm 3.6. Modulus-based Successive Overrelaxation method

- 1 . **Choose** $x, \Omega, \alpha, tol, maxit$;
- 2 . **For** $it = 1, 2, \dots, maxit$
- 3 . $z = |x| + x$;
- 4 . $b = [(1 - \alpha)D + \alpha U]x + (\Omega - \alpha A)|x| - \alpha q$;
- 5 . $Res = \|\min(Az + q, z)\|_2$;

```

6 .      If Res < tol
7 .          break;
8 .      End
9 .      Solve  $(D + \Omega - \alpha L)x = b$ ;
10.     End For

```

Note that the linear system of equations in line 9 of algorithm 3.6 can be solved exactly, since $D + \Omega - \alpha L$ is a lower triangular matrix. Similarly, projected successive overrelaxation method requires to solve $(D - \alpha L)x = b$ elementwisely. It is obvious that the coefficient matrix $D + \Omega - \alpha L$ can be more diagonal dominant than $D - \alpha L$ since Ω is a nonnegative diagonal matrix. Hence modulus-based successive overrelaxation method may be more efficient than projected successive overrelaxation method in numerical computation.

Similar to the projected successive overrelaxation method, the projection of the iterated solution onto the space $\mathbb{R}_+^n = \{x \in \mathbb{R}^n | x \geq 0\}$ is required when $|x|$ is computed in modulus-based successive overrelaxation method. However, this projection is vector operation and can use parallel computing.

We should remark that the convergence performance of modulus-based successive overrelaxation method depends on the choices of Ω and α . These parameters are often problem dependent and are generally difficult to be determined beforehand. Hence, the determination of the optimal parameters for modulus-based successive overrelaxation method could be still an open problem of theoretical and practical importance [2]. In the next section, theoretical analysis of the choice of Ω for the convergence of modulus-based successive overrelaxation method is given.

4. The Convergence of MSOR method

In this section, we establish the convergence theory for modulus-based successive overrelaxation method when the system matrix A comes from the discretization of Black-Scholes American option model. Some necessary notations and definitions are given as follows.

A matrix $A = [a_{i,j}] \in \mathbb{C}^{n \times n}$ is diagonally dominant if $|a_{i,i}| \geq \sum_{j=1, j \neq i}^n |a_{i,j}|$, $\forall 1 \leq i \leq n$, and it is strictly diagonally dominant if strict inequality is valid. A matrix $A = [a_{i,j}] \in \mathbb{R}^{n \times n}$ is called L -matrix if $a_{i,j} \leq 0$ for all $i \neq j$ and $a_{i,i} > 0$ for all $1 \leq i \leq n$. A nonsingular matrix $A = [a_{i,j}] \in \mathbb{R}^{n \times n}$ is called an M -matrix if it is a L -matrix and $A^{-1} \geq 0$; and an H -matrix if its comparison matrix $\langle A \rangle = [\langle a \rangle_{i,j}]$ is an M -matrix, where

$$\langle a \rangle_{i,j} = \begin{cases} |a_{i,i}|, & \text{for } i = j, \\ -|a_{i,j}|, & \text{for } i \neq j, \end{cases} \quad i, j = 1, 2, \dots, n \quad (20)$$

In particular, an H -matrix having positive diagonal entries is called an H_+ -matrix.

Lemma 4.1. *Let L and U be the lower and upper triangular L -matrices respectively. Then, both L^{-1} and U^{-1} are nonnegative matrices.*

Theorem 4.2. Let $A = [a_{i,j}]$ be a real $n \times n$ L -matrix with strictly diagonally dominant property. Then, A is an M -matrix.

Proof. We shall proceed to the final conclusion in three steps:

Step 1.: The real $n \times n$ L -matrix $A = [a_{i,j}]$ is strictly diagonally dominant, and $A = LU$ is the LU factorization of A . Then, both L and U are L -matrices.

Step 2.: Let L and U be lower and upper triangular L -matrices respectively. Then, both L^{-1} and U^{-1} are nonnegative matrices.

Step 3.: $A = [a_{i,j}]$ is an L -matrix and $A^{-1} = U^{-1}L^{-1} \geq 0$. Then, A is an M -matrix.

The second assertion have been proved in Lemma 4.1 and the final assertion is valid obviously. Hence, we only need to verify the Step 1.

If $n = 1$, $A = LU = [1][a_{1,1}]$ and the result is trivial. We proceed by induction on n and assume that $n > 1$ and that the result has been proved for A of order less than n . Suppose

$$A = \begin{bmatrix} a_{1,1} & -a_{1,2} & \cdots & -a_{1,n} \\ -a_{2,1} & a_{2,2} & \cdots & -a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} & \cdots & a_{n,n} \end{bmatrix} =: \begin{bmatrix} a_{11} & w^T \\ v & B \end{bmatrix}$$

where $a_{i,j} \geq 0$ for all $i, j = 1, \dots, n$ and $a_{i,i} > 0$ for all $i = 1, \dots, n$. B is a matrix of order $n - 1$. Define $M_k = I + \tau_k e_k^T$, $k = 1, \dots, n - 1$ with $\tau_k = [0, \dots, 0, x_{k+1}, \dots, x_n]^T$, $x_i = a_{i,k}/a_{k,k} > 0$, $i = k + 1, \dots, n$. The row Gaussian elimination is applied to A by left multiplying a matrix

$$M_1 A = \begin{bmatrix} a_{1,1} & w^T \\ 0 & A_1 \end{bmatrix}, \text{ where } A_1 = \begin{bmatrix} a_{2,2} - a_{1,2} \frac{a_{2,1}}{a_{1,1}} & \cdots & -a_{2,n} - a_{1,n} \frac{a_{2,1}}{a_{1,1}} \\ \vdots & \vdots & \vdots \\ -a_{n,2} - a_{1,2} \frac{a_{n,1}}{a_{1,1}} & \cdots & a_{n,n} - a_{1,n} \frac{a_{n,1}}{a_{1,1}} \end{bmatrix}.$$

Since A is strictly diagonally dominant, we have $a_{i,i} > a_{i,1} > a_{i,1} a_{1,i}/a_{1,1}$. Hence the diagonal elements $a_{i,i} - a_{i,1} a_{1,i}/a_{1,1}$ of A_1 are positive for all $i = 2, \dots, n$. Obviously the off-diagonal elements of A_1 are non-positive. Observe that

$$\begin{aligned} a_{i,i} - \sum_{j=2, j \neq i}^n a_{i,j} &> a_{i,1} > \frac{a_{i,1}}{a_{1,1}} \sum_{j=2}^n a_{1,j} \\ \Rightarrow a_{i,i} - a_{1,i} \frac{a_{i,1}}{a_{1,1}} &> \sum_{j=2, j \neq i}^n (a_{i,j} + a_{1,j} \frac{a_{i,1}}{a_{1,1}}). \end{aligned}$$

where $i = 2, \dots, n$. We can draw the conclusion that A_1 is still a strictly diagonally dominant L -matrix of order $n - 1$. By the induction hypothesis, $A_1 = L_1 U_1$ is the LU factorization of A_1 and both L_1 and U_1 are L -matrices. Hence

$$A = M_1^{-1} \begin{bmatrix} 1 & 0 \\ 0 & L_1 \end{bmatrix} \begin{bmatrix} a_{1,1} & w^T \\ 0 & U_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\tilde{\tau}_1 & L_1 \end{bmatrix} \begin{bmatrix} a_{1,1} & w^T \\ 0 & U_1 \end{bmatrix} =: LU$$

where $M_1^{-1} = I - \tau_1 e_1^T$ is an L -matrix and $\tau_1^T = [0, -\tilde{\tau}_1^T]$. Therefore, both L and U are L -matrices obviously. \square

The following corollary can be obtained by Theorem 4.2 directly.

Corollary 4.3. *Let $A = [a_{i,j}] \in \mathbb{R}^{n \times n}$ be a strictly diagonally dominant matrix. Then, A is an H -matrix. Moreover, A is an H_+ -matrix if A is strictly diagonally dominant with positive diagonal elements.*

Proposition 4.4. *Let S and B be the semi-discretization and discretization matrix of Black-Scholes partial differential equation (2), respectively. If $\sigma^2 > r$, then both S and B are H_+ -matrices.*

Proof. Recall from (10) that $S = \text{tridiag}\{a_i, b_i, c_i\}$ and note that

$$|b_i| = \sigma^2 i^2 + r > \left| \frac{1}{2}(\sigma^2 i^2 - ri) \right| + \left| \frac{1}{2}(\sigma^2 i^2 + ri) \right| = |a_i| + |c_i|$$

is valid for all i if $\sigma^2 > r$. Hence S is a strictly diagonal dominant matrix with positive diagonal elements, and consequently S is an H_+ -matrix from Corollary 4.3. Obviously, $B = (1/\Delta\tau)I + \theta S$ is also an H_+ -matrix in Rannacher scheme. \square

The convergence theorem of modulus-based successive overrelaxation method is introduced as follows.

Theorem 4.5 ([2]). *Let $A \in \mathbb{R}^{n \times n}$ be an H_+ -matrix, and $A = M - N$ be an H -compatible splitting of the matrix A , i.e., $\langle A \rangle = \langle M \rangle - |N|$. Assume that Ω is a positive diagonal matrix and γ is a positive constant. If the parameter matrix Ω satisfies $\Omega \geq \frac{1}{2} \text{diag}(M)$, then the iteration sequence $\{z^k\}_{k=1}^{+\infty} \subset \mathbb{R}_+^n$ generated by Algorithm 3.6 converges to the unique solution $z_* \in \mathbb{R}_+^n$ of the LCP(q, A) for any initial vector $x^{(0)} \in \mathbb{R}^n$.*

Finally, we establish the convergence theorem of Algorithm 3.6 for pricing American options.

Proposition 4.6. *Let S and B be the semi-discretization and discretization matrix of Black-Scholes partial differential equation (2), respectively. If $\sigma^2 > r$ and the parameter matrix Ω satisfies $\Omega \geq \frac{1}{2} \text{diag}(M)$, then modulus-based successive overrelaxation algorithm 3.6 for Black-Scholes American option pricing converges to the unique solution for any initial vector.*

Proof. According to the Proposition 4.4, the discretization matrix B is an H_+ -matrix. Also, $B = D - L - U$ is an H -compatible splitting of the matrix B . From Theorem 4.5, if $\Omega \geq \frac{1}{2} \text{diag}(M)$, then modulus-based successive overrelaxation algorithm 3.6 converges to the unique solution for any initial vector. \square

5. Numerical Experiments

In this section, a number of numerical experiments are presented to show the efficient of modulus-based successive overrelaxation method for the linear complementarity problem resulted from American put options pricing.

The parameters in American put options are given as follows:

$$\sigma = 0.2, r = 0.02, K = 10 \quad \text{and} \quad T = 1, \tag{21}$$

and the computational domain is chosen to be $(x, t) \in [0, 50] \times [0, 1]$ since artificial boundary $X = 50$ is five times of K in order to avoid excessive error. Note that $\sigma^2 > r$, Algorithm 3.6 is guaranteed to converge to the unique solution for any initial vector according to Theorem 4.6.

All of the computations were run on a computer where the CPU is 2.20 GHz and the memory is 2.00 GB, and the programming language was MATLAB 7.8.0.347 with machine precision $\epsilon = 1.1 \times 10^{-16}$. In order to perform a fair comparison among different methods, in all experiments, we choose the stopping criterion as

$$\text{Res}(z^k) := \|\min(Az^k + q, z^k)\|_2 < \text{tol}, \tag{22}$$

with $\text{tol} = 10^{-5}$, or k reaches the maximal number of iteration steps, e.g., 1000, where z^k represents the k th numerical solution of the linear complementarity problem $\text{LCP}(q, A)$. $\text{Res}(z^k)$ is called the residual of numerical solution z^k in $\text{LCP}(q, A)$.

In the following numerical experiment, we first depict the surface of American put option value on a fine grid, then show the convergence performance of modulus-based successive overrelaxation method (MSOR), compared with projected successive overrelaxation method (PSOR) when the parameters and the number of discretization grids is varying. The number of iteration steps, the CPU time and the relative error of MSOR and PSOR are reported. Moreover, the curves of CPU time versus the number of space discretization grids m for MSOR and PSOR are depicted respectively.

5.1. The Surface of American Put Option Value. In order to report relative errors for numerical solutions, the reference numerical solution for the American put option problem is computed by a fine grid with $(m, n) = (7680, 7680)$, the Rannacher scheme and MSOR method numerically.

Numerical results are plotted in Figure 1 where the surface represents American put option price with respect to underlying asset value x and time t , and the line in the bottom plane indicates the optimal exercise boundary of American put option.

5.2. Numerical Comparison on MSOR and PSOR. In this section, we discuss the convergence performance of modulus-based successive overrelaxation method and projected successive overrelaxation method.

Here, we set $\Omega = \beta D$ in the numerical experiment, where the positive parameter β is chosen by minimizing the number of iteration steps. The relative error (denoted by ‘Error’) is defined as

$$\text{Error} = \frac{\|x^k - x^*\|_2}{\|x^*\|_2}, \tag{23}$$

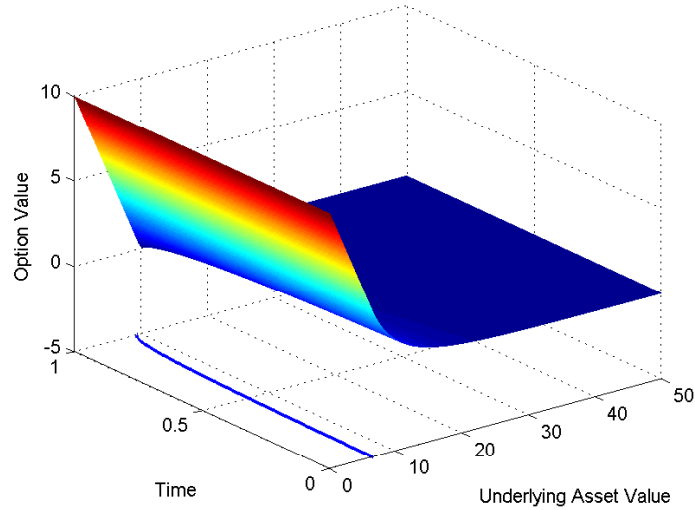


FIGURE 1. The surface of option value and the optimal exercise boundary curve.

where x^k is k th numerical solution and x^* represents reference solution, $\|\cdot\|_2$ is the l_2 -norm of a vector.

TABLE 1. Iteration number and CPU time of two methods with $(m, n) = (250, 250)$.

$\alpha = \omega$	PSOR			MSOR			
	IT	CPU	Error	IT	CPU	Error	β^*
0.8	10.3	14.45	6.19e-004	8.4	1.97	6.19e-004	0.51
0.9	8.0	11.15	6.19e-004	8.4	1.97	6.19e-004	0.70
1.0	6.0	8.31	6.19e-004	8.4	1.95	6.19e-004	0.89
1.1	4.2	5.79	6.19e-004	8.4	1.71	6.19e-004	1.08
1.2	6.1	8.56	6.19e-004	8.4	1.93	6.19e-004	1.26
1.3	8.3	11.59	6.19e-004	8.4	1.99	6.19e-004	1.45
1.4	11.2	15.66	6.19e-004	8.4	1.92	6.19e-004	1.64
1.5	14.6	20.26	6.19e-004	8.4	1.98	6.19e-004	1.83
1.6	19.7	27.52	6.19e-004	8.7	1.95	6.19e-004	2.03
1.7	27.9	39.11	6.19e-004	9.0	2.09	6.19e-004	2.13
1.8	43.7	60.56	6.19e-004	9.6	2.01	6.19e-004	2.33
1.9	89.2	124.21	6.19e-004	10.1	2.24	6.19e-004	2.49

In Tables 5.2, 2 and 3, we list the the number of iteration steps (denoted by ‘IT’) and the CPU time (denoted by ‘CPU’) in seconds of MSOR and PSOR,

respectively. In Table 5.2, the parameter $\alpha = \omega$ varies from 0.8 to 1.9 while the optimal β^* for the least iteration steps is chosen for MSOR. The number of discretization grids (m, n) is chosen to be (250, 250). Moreover, the number of iteration steps and the CPU time for MSOR and PSOR on different discretization grids (m, n) are reported in Tables 2 and 3. The optimal parameter ω^* is chosen by minimizing the number of iteration steps of PSOR.

From Table 5.2, on one hand, it is observed that the iteration steps and CPU time of PSOR method decrease at first, and then increase with respect to the parameter ω . The optimal parameter for PSOR on $(m, n) = (250, 250)$ approximates to 1.1. For MSOR method, the iteration steps and CPU time keep nearly the same when $\alpha = \omega$ and the optimal β^* is chosen. Moreover, the relative errors of numerical solutions stay constantly if (m, n) is fixed.

On the other hand, when the optimal parameters ω^* and β^* are chosen for PSOR and MSOR respectively, it can be concluded that the number of iteration steps for MSOR is higher than that of PSOR though, MSOR requires less CPU time. Moreover, in the non-optimal parameter case, the PSOR method may require more iteration steps and consequently more CPU time than MSOR method. We remark that MSOR is superior to PSOR in computational efficiency.

TABLE 2. Comparison of two methods on different discretization grids with $\alpha = 1.2$. (Option parameters: $\sigma = 0.2$ and $r = 0.02$)

(m, n)	PSOR				MSOR			
	IT	CPU	Error	ω^*	IT	CPU	Error	β^*
(60,30)	3.2	0.11	6.46e-003	1.02	5.6	0.04	6.46e-003	1.36
(60,60)	3.0	0.22	4.10e-003	1.01	5.0	0.06	4.10e-003	1.35
(60,120)	2.7	0.36	2.95e-003	1.01	4.0	0.10	2.95e-003	1.38
(120,60)	4.1	0.59	2.65e-003	1.07	7.4	0.36	2.65e-003	1.30
(120,120)	3.3	0.95	1.54e-003	1.04	6.0	0.61	1.54e-003	1.33
(120,240)	3.0	1.75	9.92e-004	1.01	5.0	1.18	9.92e-004	1.34
(240,120)	4.8	3.09	1.19e-003	1.16	11.0	1.19	1.19e-003	1.23
(240,240)	3.9	5.09	6.49e-004	1.09	8.3	1.77	6.49e-004	1.26
(240,480)	3.2	8.42	3.80e-004	1.06	6.5	3.07	3.80e-004	1.33
(480,240)	11.8	18.97	5.62e-004	1.30	17.9	4.87	5.62e-004	1.12
(480,480)	9.2	29.55	2.94e-004	1.18	12.3	6.94	2.94e-004	1.20
(480,960)	8.0	50.50	1.61e-004	1.09	9.1	10.86	1.61e-004	1.28
(960,480)	23.4	135.08	2.73e-004	1.46	31.7	25.72	2.73e-004	1.04
(960,960)	17.4	202.02	1.40e-004	1.31	23.0	38.57	1.40e-004	0.97
(960,1920)	14.4	331.28	7.31e-005	1.19	16.0	54.91	7.31e-005	1.02

From Tables 2 and 3, it can be observed that the number of iteration steps and CPU time for both PSOR and MSOR increase when the number of spatial discretization grids increase. However, the number of iteration steps decreases

TABLE 3. Comparison of two methods on different discretization grids with $\alpha = 1.2$. (Option parameters: $\sigma = 0.3$ and $r = 0.03$)

(m, n)	PSOR				MSOR			
	IT	CPU	Error	ω^*	IT	CPU	Error	β^*
(60,30)	4.9	0.20	1.00e-002	1.08	8.5	0.04	1.00e-002	1.30
(60,60)	4.0	0.30	5.67e-003	1.04	6.7	0.06	5.67e-003	1.31
(60,120)	3.4	0.45	3.54e-003	1.03	5.5	0.10	3.54e-003	1.35
(120,60)	6.3	0.92	4.46e-003	1.20	14.0	0.53	4.46e-003	1.18
(120,120)	5.0	1.45	2.40e-003	1.12	10.1	0.87	2.40e-003	1.26
(120,240)	4.1	2.42	1.38e-003	1.05	7.7	1.26	1.38e-003	1.34
(240,120)	8.4	5.46	2.10e-003	1.39	25.7	2.32	2.10e-003	1.03
(240,240)	6.5	8.36	1.09e-003	1.23	16.9	3.12	1.09e-003	1.11
(240,480)	5.1	13.35	5.89e-004	1.13	11.7	4.75	5.89e-004	1.16
(480,240)	23.0	36.96	1.02e-003	1.53	50.6	12.59	1.02e-003	0.92
(480,480)	16.8	54.35	5.18e-004	1.40	31.1	15.84	5.18e-004	0.97
(480,960)	13.4	86.02	2.70e-004	1.23	19.8	20.76	2.70e-004	1.06
(960,480)	48.6	284.98	5.00e-004	1.66	104.9	80.53	5.00e-004	0.87
(960,960)	33.9	396.77	2.52e-004	1.55	61.7	96.67	2.52e-004	0.88
(960,1920)	26.1	609.76	1.29e-004	1.40	38.5	122.03	1.29e-004	0.85

as the number of time discretization grids n increase. The reason is that the coefficient matrix $B = (1/\tau)I + \theta S$ becomes more diagonal dominant as the increase of n . Since the number of linear complementarity problems doubles as n doubles, the CPU time of MSOR and PSOR increase. Meanwhile, it is shown that the numerical solutions of linear complementarity problems from American put option pricing become more accurate as the relative errors decrease when the number of discretization grids (m, n) increase. We choose another pairs of American put option's parameters $(\sigma, r) = (0.3, 0.03)$ to do the experiments, and the similar conclusions can be reached from Table 3.

In Figure 2, the curves of CPU time of MSOR and PSOR versus the number of space discretization grids m are plotted respectively where $m = n$. It is shown that PSOR requires more CPU time than MSOR method on different discretization grids.

6. Concluding Remarks

Due to the early exercise constraint, the problem of American option pricing can be discretized into a linear complementarity problem. In this paper, we consider the modulus-based successive overrelaxation method for the numerical solution of the linear complementarity problem. The convergence theorem of modulus-based successive overrelaxation algorithm for the discretized American option pricing model is given. Numerical experiments further verify the validity

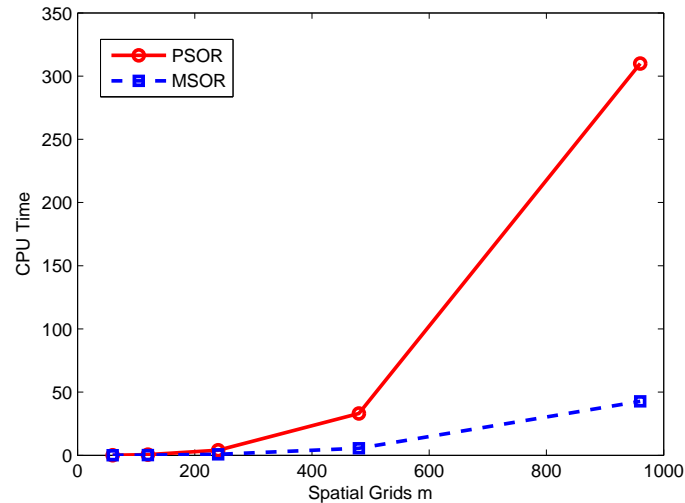


FIGURE 2. The CPU time in seconds of PSOR and MSOR with $m = n$ and $\alpha = 1.2$.

of convergence conditions and show that the proposed method is much faster than classical projected successive overrelaxation method with optimal parameter.

We should remark that both PSOR and MSOR are parameter related methods, which means their convergence performance and practical effectiveness highly depend on the choice of parameters. However, MSOR can take the advantage of vector operation and can use parallel computing while PSOR requires to do projection elementwisely. Hence, MSOR may be more effective than PSOR in practical application.

Acknowledgement

The authors are most grateful to the unknown referees who made a number of constructive comments which greatly improved the quality of the paper.

REFERENCES

1. Y. Achdou and O. Pironneau, *Computational Methods for Option Pricing*, Frontiers in Applied Mathematics, SIAM, Philadelphia, 2005.
2. Z.-Z. Bai, *Modulus-based matrix splitting iteration methods for linear complementarity problems*, Numer. Linear Algebra Appl. **6** (2010), 917-933.
3. F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Political Economy **81** (1973), 637-654.
4. M. J. Brennan and E. S. Schwartz, *The valuation of American put options*, J. Finance **32** (1977), 449-462.

5. C. W. Cryer, *The solution of a quadratic programming using systematic overrelaxation*, SIAM J. Control **9** (1971), 385-392.
6. J.-L. Dong and M.-Q. Jiang, *A modified modulus method for symmetric positive-definite linear complementarity problems*, Numer. Linear Algebra Appl. **16** (2009), 129-143.
7. P. A. Forsyth and K. R. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM J. Sci. Comput. **23** (2002), 2095-2122.
8. A. Hadjidimos and M. Tzoumas, *Nonstationary extrapolated modulus algorithms for the solution of the linear complementarity problem*, Linear Algebra Appl. **431** (2009), 197-210.
9. J. Hull, *Options, Futures, and Other Derivatives*, 8th ed., Prentice-Hall, Boston, 2012.
10. J. Hull and A. White, *Valuing derivatives securities using the explicit finite difference method*, J. Finan. Quant. Anal. **25** (1990), 87-100.
11. S. Ikonen and J. Toivanen, *Efficient numerical methods for pricing American options under stochastic volatility*, Numer. Methods for Partial Differential Equations **24** (2008), 104-126.
12. S. Ikonen and J. Toivanen, *Operator splitting methods for American option pricing with stochastic volatility*, Numer. Math. **113** (2009), 299-324.
13. R. Kangro and R. Nicolaidis, *Far field boundary conditions for Black-Scholes equations*, SIAM J. Numer. Anal. **38** (2000), 1357-1368.
14. O. Mangasarian, *Solutions of symmetric linear complementarity problems by iterative methods*, J. Optim. Theory Appl. **22** (1977), 465-485.
15. K. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann, Berlin, 1988.
16. R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numer. Math. **43** (1984), 309-327.
17. R. Seydel, *Tools for Computational Finance*, Universitext, Springer-Verlag, Berlin, 2002.
18. D. Tavella and C. Randall, *Pricing Financial Instruments: The Finite Difference Method*, John Wiley & Sons, Chichester, 2000.
19. W. M. G. van Bokhoven, *A class of linear complementarity problems is solvable in polynomial time*, Unpublished paper, Dept. of Electrical Engineering, University of Technology, The Netherlands, 1980.

Ning Zheng received bachelor degree at Tongji University. Since 2010 he has been a PhD student at Tongji University. His research interests include linear complementarity problem, iterative method and American option pricing.

Department of Mathematics, Tongji University, 1239 Siping Road, Shanghai 200092, P. R. China.

e-mail: 6zhengning@tongji.edu.cn

Junfeng Yin received Ph.D. at China Academy of Science. Since 2008 he has been an associate professor at Tongji University. His research interests include numerical linear algebra, iterative method, parallel computation, financial and biohealth computation.

Department of Mathematics, Tongji University, 1239 Siping Road, Shanghai 200092, P. R. China.

e-mail: yinjf@tongji.edu.cn