Fast solutions of complementarity formulations in American put pricing

Artan Borici

Department of Physics, University of Tirana

King Zog I Boulevard, Tirana-Albania

and

Hans-Jakob Lüthi

Institute for Operations Research, ETH Zürich

CH-8082 Zurich, Switzerland

Abstract

The finite differences value function of an American put option can be computed by solving a sequence of Linear Complementarity Problems (LCP). The state of the art methods that solve these equations converge slowly. Recently, Borici and Lüthi have shown that in the case of the implicit discretisation scheme it is possible to solve LCPs in a computer time which grows linearly with the number of spatial grid points. In this paper we show that this result can be generalised for the more accurate discretisation scheme of Crank-Nicolson. We give examples that illustrate this result.

1 Introduction

Pricing American put options is a challenging numerical task. First of all there is no closed form and exact solution to this problem. The basic property of an American option is the early exercise feature of the option. This leads to an optimization problem, the Linear Complementary Probelm (LCP), which must be solved on top of the Black-Scholes partial differential equation (PDE). These features make the American option pricing a complex problem. The methods of solution can be devided into tree and PDE methods [17].

Most popular methods are those based on trees or lattices [6]. On trees the LCPs are trivial to solve and hence very fast but the accuracy is of first order in time. A more accurate approach is the finite diference scheme of Crank-Nicolson (CN) which is of second order in time. But in this case LCPs are non-trivial to solve.

A fast solution of the LCP is critical for the option evaluation. The general pivoting algorithms may be too slow. Fortunately the discretization leads to special class of Z-matrices for which the algorithm of Chandrasekaran converges in polynomial time [5]. Other discretization schemes lead to different types of matrices for which corresponding algorithm often exists [13].

So far, the most popular algorithm has been the projected successive overrelaxation (PSOR) iterative method of Cryer [7]. The method is the usual SOR method for solving linear systems which is modified to update only non-negative SOR solutions. But the number of iterations required to converge may vary and sometimes can grow rapidly with the number of spatial grid points [9].

Recently, Dempster, Hutton & Richards [9] proposed an algorithm which evaluates the American option in times that are observed to vary linearly with the space discretization. The authors show the equivalence to a corresponding LP and solve it using plausible assumptions on the form of the optimal basis. Later, Borici and Lüthi [2] proved that a complementary feasible basis alluded by [9] exists and gave the corresponding algorithm which finds it. However, the algorithm was shown to be valid for the implicit difference scheme only.

In this paper we show that the algorithm can be extended for the more accurate scheme of Crank-Nicolson under certain conditions. It also compares the new algorithm to the PSOR algorithm and their complexity for a fixed accuracy.

The paper is organized as follows: in the next section we set the notations and define the problem of evaluation of the American option as a sequence of LCPs. In section 3 we describe the new algorithm and prove it in general for any difference scheme. In section 4 we discuss implementation and give examples of computations with corresponding run times. Complexity and accuracy are discussed in section 5.

2 Definition of the pricing problem

The American pricing problem can be formulated as the solution of an optimal stopping problem using variational inequalities (VI) which lead to the linear complementarity problem (LCP) [14],[10] or a linear program (LP) [8].

Let us assume a Black-Scholes economy with one risky asset price S modeled by a geometric Brownian motion with constant volatility σ and a savings account with constant risk-free rate $r \geq 0$.

An European option gives the holder the right to buy or sell one unit of the asset for a price K, the *strike price*, at the *maturity* date T. In contrast, an American option can be exercised at any time τ to maturity, i.e. $\tau \in [0, T]$. The payoff of an American put option is a function $\psi : \mathbb{R}^+ \to \mathbb{R}$ defined by:

$$\psi(S_\tau) = (K - S_\tau)^+$$

The value function $v : \mathbb{R}^+ \times [0, T] \to \mathbb{R}$ is the "fair" value $v(x, t)$ at asset price $x > 0$ and at time $t \in [0, T]$. It can be formulated as the solution of an optimal stopping problem, namely choose the stopping time which maximizes the conditional expectation of the discounted payoff. The stopping time may be shown to be the first time the value falls to the payoff at exercise [16]. In particular, the (x, t) domain may be partitioned as follows:

$$\mathcal{C} = \{(x,t) \in \mathbb{R}^+ \times [0,T) : v(x,t) > \psi(x)\}$$
$$\mathcal{S} = \{(x,t) \in \mathbb{R}^+ \times [0,T) : v(x,t) = \psi(x)\}$$

On the *continuation region* $\mathcal{C}$, v has to satisfy the Black-Scholes PDE:

$$(\mathcal{L}_{BS} + \partial_t)v = 0 \quad \text{with} \quad v > \psi \tag{2.1}$$

whereas on the *stopping region* $\mathcal{S}$ one avoids arbitrage by requiring:

$$(\mathcal{L}_{BS} + \partial_t)v \leq 0 \quad \text{with} \quad v = \psi$$

with $\mathcal{L}_{BS}$, the Black-Scholes operator defined by:

$$\mathcal{L}_{BS} = \frac{\sigma^2}{2}x^2\partial_x^2 + rx\partial_x - r \tag{2.2}$$

Conditions (2.1-2) lead to the following order complementarity problem (OCP) for the fair value of the American put option [3]:

$$\begin{aligned}
&\text{OCP} \\
&\text{Find} \quad v \in \mathcal{F} \quad \text{such that:} \\
&-(\mathcal{L}_{BS} + \partial_t)v \wedge (v - \psi) = 0 \\
&\text{where} \\
&\mathcal{F} = \{v : v - \psi \geq 0, -(\mathcal{L}_{BS} + \partial_t)v \geq 0\}
\end{aligned} \tag{2.3}$$

where $\wedge$ denotes the pointwise minimum of two functions with respect to a *vector lattice* Hilbert space (see [3] for further discussion).

Note that the Black-Scholes PDE is a linear elliptic PDE with non-constant coefficients. In fact, a log-transformed stock price $\xi = \log x$ is useful to define a constant coefficient Black-Scholes operator:

$$\mathcal{L}_{BS} = \frac{\sigma^2}{2}\partial_\xi^2 + (r - \frac{\sigma^2}{2})\partial_\xi - r \tag{2.4}$$

with a terminal condition (corresponding to the payoff function) given by:

$$\psi(\xi, T) = (K - e^\xi)^+ \tag{2.5}$$

In the rest of the paper we assume the above form of the Black-Scholes operator eq. (2.4) and terminal condition eq. (2.5).

Formulation on a discrete and finite domain

Since the analytical solution to the above OCP (2.3) is not known one resorts to numerical methods. For a numerical approximation the function space has to be finite and the value function discrete.

We define the problem on a rectangular domain $[L, U] \times [0, T]$ and assume that L and U are chosen properly so that for practical purposes they do not effect the option price. Note that the infinite domain solution is recovered in the limit $L \to -\infty, U \to +\infty$ [14].

The differential operators are approximated by homogeneous *finite differences* on a lattice with $(I + 1) \times (M + 1)$ number of points. We label these points by indices as follows:

$$\xi_i = L + i\Delta\xi, \quad i = 0, \dots, I, \quad \Delta\xi = (U - L)/I \tag{2.6}$$

$$t_m = T - m\Delta t, \quad m = 0, \dots, M, \quad \Delta t = T/M \tag{2.7}$$

The discrete value function on this domain is denoted by:

$$v_i^m = v(\xi_i, t_m), \quad m = 0, \dots, M, \quad i = 0, \dots, I$$

with *boundary values*:

$$v_0^m = \psi(L), \;\; v_I^m = \psi(U), \quad m = 0, \dots, M$$

and *terminal value*:

$$v_i^0 = \psi(\xi_i) \equiv \psi_i, \quad i = 0, \dots, I$$

The time derivative in the Black-Scholes equation (2.1) is approximated by the finite difference:

$$\partial_t v \approx \frac{v_i^{m-1} - v_i^m}{\Delta t}$$

Spatial derivatives in the Black-Scholes operator eq. (2.4) are approximated by finite difference derivatives:

$$\partial_\xi v \approx \theta \frac{v_{i+1}^m - v_{i-1}^m}{2\Delta\xi} + (1 - \theta) \frac{v_{i+1}^{m-1} - v_{i-1}^{m-1}}{2\Delta\xi}$$
$$\partial_\xi^2 v \approx \theta \frac{v_{i+1}^m - 2v_i^m + v_{i-1}^m}{(\Delta\xi)^2} + (1 - \theta) \frac{v_{i+1}^{m-1} - 2v_i^{m-1} + v_{i-1}^{m-1}}{(\Delta\xi)^2}$$

whereas the value function near the constant term is split as follows:

$$v \to \theta v_i^m + (1-\theta) v_i^{m-1}$$

Here $i = 1, \dots, I-1$ and $m = 1, \dots, M$ and θ is a parameter that controls the stability of the proposed difference scheme. The scheme is known to be unconditionally stable for $0.5 \le \theta \le 1$. For the financial oriented reader we refer to [18].

Let $n = I - 1$ and $v^m \in \mathbb{R}^n$ be the discrete value function at time time slice m. In vector notations we write:

$$v^m = \begin{pmatrix} v_1^m \\ v_2^m \\ \vdots \\ v_n^m \end{pmatrix},$$

Then terminal and boundary value vectors $\psi, \phi \in \mathbb{R}^n$ are given by:

$$v^0 = \psi = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}, \qquad \phi = \rho(1-\tilde{q}) \begin{pmatrix} \psi_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where ρ is called *the mesh ratio* and given by:

$$\rho = \frac{\sigma^2 \Delta t}{(\Delta \xi)^2} \tag{2.8}$$

We define also the matrix $\tilde{Q} \in \mathbb{R}^{n \times n}$:

$$\tilde{Q} = \begin{pmatrix} 0 & \tilde{q} & & \\ 1-\tilde{q} & 0 & \ddots & \\ & \ddots & \ddots & \tilde{q} \\ & & 1-\tilde{q} & 0 \end{pmatrix},$$

with the matrix element $\tilde{q}$ is given by:

$$\tilde{q} = \frac{1}{2} + \frac{1}{2}(r - \frac{\sigma^2}{2})\frac{\Delta \xi}{\sigma^2}$$

Then using the Black-Scholes equation eq. (2.1) and above definitions one gets the **Black-Scholes difference equations** as the following sequence of matrix equations:

$$B\ v^{m-1} + A\ v^m - \phi = 0, \quad m = 1, \ldots, M \tag{2.9}$$

where

$$A = \mathbb{1} + \theta\ [r\Delta t\ \mathbb{1} + \rho(\mathbb{1} - \tilde{Q})]$$

$$B = -\mathbb{1} + (1-\theta)\ [r\Delta t\ \mathbb{1} + \rho(\mathbb{1} - \tilde{Q})]$$

with $\mathbb{1} \in \mathbb{R}^{n \times n}$ denoting the identity matrix. Note that the discrete option price converges to the continuous price for $\Delta t \to 0$, $\Delta\xi \to 0$ and $\rho \to 0$ [14].

We note first that the matrix A is positive definite, since the symmetric part of $\mathbb{1} - \tilde{Q}$:

$$\begin{pmatrix} 1 & -\frac{1}{2} & & \\ -\frac{1}{2} & 1 & \ddots & \\ & \ddots & \ddots & -\frac{1}{2} \\ & & -\frac{1}{2} & 1 \end{pmatrix},$$

is symmetric positive definite. In the case $0 \leq \tilde{q} \leq 1$ the matrix A has all its off-diagonal elements nonpositive and belongs to the class of the so called Z-matrices, a crucial property that we shall use in the next section.

In terms of the problem parameters and the spatial lattice spacing the matrix A will be of Z-type if:

$$|r - \frac{\sigma^2}{2}| \leq \frac{\sigma^2}{\Delta\xi}$$

which we shall assume.

Note that in this case $\tilde{q}$ may be interpreted as the "would be" binomial probability of a small upwards change of the stock price.

We note also some properties of matrices A and B which we shall find useful below. The following identity holds:

$$-A^{-1}B = \frac{1}{\theta}A^{-1} - \frac{1-\theta}{\theta}\ \mathbb{1} \tag{2.10}$$

But we can write further A as:

$$A = \alpha(\mathbb{1} - \kappa\tilde{Q}), \quad \alpha = 1 + \theta(r\Delta t + \rho), \quad \kappa = \frac{\theta\rho}{\alpha} < 1 \tag{2.11}$$

and its inverse as:

$$A^{-1} = \frac{1}{\alpha}(\mathbb{1} + \kappa\tilde{Q} + \cdots) \geq \frac{1}{\alpha}\mathbb{1} \tag{2.12}$$

With the discrete Black-Scholes equation (2.9) at hand one can formulate the computation of the American option as the following sequence of (linear) complementarity problems:

$$\begin{aligned} &\text{For} \quad m = 1, \ldots, M \\ &v^m \geq \psi \\ &B\ v^{m-1} + A\ v^m - \phi \geq 0 \\ &(B\ v^{m-1} + A\ v^m - \phi) \wedge (v^m - \psi) = 0 \end{aligned} \tag{2.13}$$

where $\wedge$ denotes the componentwise minimum of two vectors $\in \mathbb{R}^n$.

But the above complementarity problems can be written as a usual LCP sequence. Let $u^m \equiv v^m - \psi, m = 1, \ldots, M$ be the excess value vector and $s^m \equiv B\ v^{m-1} + A\ v^m - \phi$ the *slack* vector. Then for $m = 1, \ldots, M$ the time slices of an American option can be computed by solving the following sequence of LCPs:

$$\begin{aligned} &\text{LCP} \\ &\text{For} \quad m = 1, \ldots, M \\ &Au^m - s^m = b^m \\ &u^m \geq 0, \quad s^m \geq 0, \quad (s^m)^T u^m = 0 \end{aligned} \tag{2.14}$$

where

$$b^m = \phi - B(u^{m-1} + \psi) - A\ \psi, \quad u^o = 0$$

or

$$\begin{aligned} &b^m = b^o - Bu^{m-1} \\ &b^o = \phi - B\ \psi - A\ \psi \\ &u^o = 0 \end{aligned} \tag{2.15}$$

3 Solution of LCP

In the sequel we will investigate the solution to the LCP (2.14). Here we refer to the huge literature on LCP which was summarized by Cottle, Pang and Stone [5].

Since A is a positive definite matrix, it is a so-called P-matrix (a matrix with all its principal minors positive, see [5]) and therefore LCP has a *unique* solution for all right hand sides b^m [15]. Given the existence of the solution to (2.14), we are left with the problem of computing it efficiently.

In fact A is by construction a Z-matrix. In this case the LCP can be solved by pivoting techniques in $O(n^3)$, i.e. polynomial by the method developed by Chandrasekaran [4]. In the following we make use of the fact that if A is a Z-matrix the following statements are equivalent (see [11]):

$$\begin{array}{ll} \text{(a)} & A \ \text{ is a P-matrix} \\ \text{(b)} & A^{-1} \geq 0 \\ \text{(c)} & \text{There is } \ x \geq 0 \ \text{ s.t. } \ Ax > 0 \ \text{ has a solution} \end{array} \tag{3.1}$$

We will use these equivalent properties in the design of a linear complexity algorithm to solve the LCP as stated below.

As pointed out by Dempster and Hutton, for Z-matrices a solution to the LCP can be obtained using the least element property as shown in [8] and a linear programme (LP). In particular, the optimal basis to LP was assumed to contain slack and real basic variables in the following order:

$$\bar{u}^m_{n_b} = (s^m_1, \ldots, s^m_{n_b}, u^m_{n_b+1}, \ldots, u^m_n)^T \tag{3.2}$$

This is a neat assumption which was proven later by [2] in the case of the implicit difference scheme. Here we restate the argument in the general case $\theta \in [0, 1]$.

Theorem 3.1 *The solution to the LCPs (2.14) are unique and the complementary feasible bases are of the structure of eq. (3.2).*

The proof can be found at [2] and is given here for completness and with minor changes.

Note that the proof is constructive. It serves at the same time as an algorithm that finds the solution to LCP.

According to (3.2) and for a fixed time step m we have the following complementary partition of the LCP (2.14):

$$\begin{pmatrix} A_{11} & -\beta e_{n_b-1} & \\ -\gamma e_{n_b-1}^T & \alpha & -\beta e_1^T \\ & -\gamma e_1 & A_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ u_3 \end{pmatrix} - \begin{pmatrix} s_1 \\ s_2 \\ 0 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{3.3}$$

where the time slice index m has been omitted from the vectors to simplify notations. Note also that by definition of A we have $\alpha = 1 + \theta(r\Delta t + \rho)$, $\beta = \theta\rho\tilde{q}$ and $\gamma = \theta\rho(1 - \tilde{q})$.

Then the following property holds:

Lemma 3.2 *For each time step there is a partition of the above form such that:*

$$b_1, \quad b_2 < 0, \quad A_{33}^{-1} b_3 \geq 0 \tag{3.4}$$

for small enough $\Delta\xi$*.*

Note that this lemma was proven earlier for the case $\theta = 1$ [2]. Here we generalise it for any θ. But, first we proceed with the proof of the theorem.

Proof of Theorem 3.1. The complementary basic solution corresponding to the above partition (3.3) with the property (3.4) reads:

$$\begin{aligned} s_1^{\text{old}} &= -b_1 > 0 \\ s_2^{\text{old}} &= -(b_2 + \beta e_1^T u_3^{\text{old}}) \\ u_3^{\text{old}} &= A_{33}^{-1} b_3 \geq 0 \end{aligned}$$

Note that if $s_2^{\text{old}} \geq 0$ then the LCP (2.14) is solved for this m. Else $s_2^{\text{old}} < 0$ and by decreasing n_b with one unit one has:

$$b_1^{\text{new}}, \quad b_2^{\text{new}} < 0, \quad b_3^{\text{new}} = \begin{pmatrix} b_2 \\ b_3 \end{pmatrix}$$

The new complementary basic solution in terms of the old one reads:

$$\begin{aligned} s_1^{\text{new}} &= -b_1^{\text{new}} > 0 \\ s_2^{\text{new}} &= -(b_2^{\text{new}} + \beta z) \\ u_3^{\text{new}} &= \begin{pmatrix} z \\ u_3^{\text{old}} + z\gamma A_{33}^{-1} e_1 \end{pmatrix} \end{aligned} \tag{3.5}$$

where

$$z = \frac{-s_2^{\text{old}}}{\alpha - \beta y} \tag{3.6}$$

with $y = \gamma e_1^T A_{33}^{-1} e_1$.

We show now that $u_3^{\text{new}} \geq 0$. First, we observe that $A_{33}^{-1} e_1 \geq 0$ by virtue of property (3.1b). Furthermore the denominator of (3.6) is positive as a principal minor of A being positive definite:

$$\alpha - \beta\gamma e_1^T A_{33}^{-1} e_1 = \det \begin{pmatrix} \alpha & -\beta e_1^T \\ -\gamma e_1 & A_{33} \end{pmatrix} \det A_{33}^{-1}$$

Also as we noted above $s_2^{\text{old}} < 0$. Hence, $z > 0$ and therefore $u_3^{\text{new}} \geq 0$.

The next step is to check the sign of s_2^{new}. If it is negative then decrease again n_b by one and apply the above procedure. Note that in this case y can be updated using

$$y \leftarrow \frac{\gamma}{\alpha - \beta y}$$

One can continue this way until we find $s_2^{\text{new}} \geq 0$, in which case the LCP is solved ■

This proof gives rise and justifies Algorithm 1.

Corollary 3.3 *The "While do loop" of the Algorithm 1 completes in at most n steps.*

Now we continue with the *proof of lemma 3.2.* To prove the property (3.4) we first prove that:

The right hand side of the first time step has the following sign structure:

$$b^o = \begin{pmatrix} - \\ \oplus \end{pmatrix} \tag{3.7}$$

Algorithm 1 Theoretical form of the algorithm AOPT

for $m = 1, \ldots, M$ **do**

  Start from a complementary partition (3.3), which has the property (3.4)

  Compute $\bar{u}_{n_b}^{\text{old}}$

  if $n_b = 0$ **then**

    stop

  end if

  while $s_{n_b}^{\text{old}} < 0$ **do**

    $n_b := n_b - 1$

    Compute $\bar{u}_{n_b}^{\text{new}}$ according to (3.5-3.6)

    Set $\bar{u}_{n_b}^{\text{old}} := \bar{u}_{n_b}^{\text{new}}$

    if $n_b = 0$ **then**

      stop

    end if

  end while

end for

where by $\oplus$ is denoted the non-negativity of these components.

Indeed, since $u^o = 0$ we have:

$$b^o \equiv b = \phi - (A + B)\psi$$

or by its elements:

$$b_i = -\rho[\psi_i - \tilde{q}\psi_{i+1} - (1 - \tilde{q})\psi_{i-1}] - r\Delta t\psi_i, \quad i = 1, \ldots, n$$

Assume that i_o is the index such that:

$$\psi_{i \le i_o} = K - e^{\xi_i} > 0, \quad \psi_{i > i_o} = 0$$

Then one has

$$b_{i_o+1} = \rho(1 - \tilde{q})\psi_{i_o} \ge 0$$

and therefore $b_{i>i_o} \ge 0$. If i_o is such that $K = e^{\xi_{i_o}}$ then $b_{i \ge i_o} \ge 0$, otherwise b_{i_o} may have either sign.

To analyze the $i < i_o$ case one may define a new vector c_i by:

$$b_i = \rho c_i - r\Delta t\psi_i$$

where

$$c_i = \tilde{q}\psi_{i+1} + (1 - \tilde{q})\psi_{i-1} - \psi_i$$

or

$$c_i = e^{\xi_i}(1 - \cosh\Delta\xi) + e^{\xi_i}(1 - 2\tilde{q})\sinh\Delta\xi$$

The first term is always negative, whereas the second term may be of either sign: for $\tilde{q} \ge \frac{1}{2}$ it is non-positive and therefore $b_{i<i_o} < 0$ too. In case of $\tilde{q} < \frac{1}{2}$ the terms are competing and the sign of $b_{i<i_o}$ may be positive for some indices i near i_o. In any case one can restrict $\Delta\xi$ to be small enough such that $b_{i<i_o} < 0$. For example, a sufficient condition may be the non-positivity of $c_{i<i_o}$. Using the definition of $\tilde{q}$ and a simple algebra yields:

$$\frac{\frac{\Delta\xi}{2}}{\tanh\frac{\Delta\xi}{2}} \le \frac{1}{\left|1 - \frac{2r}{\sigma^2}\right|} \tag{3.8}$$

This condition implies that in the extreme case of high volatility $(\Delta\xi)^2$ should of the order $O(r/\sigma^2)$.

A consequence of the above structure is the property (3.4) for the first time step $m = 1$. Indeed, since $A^{-1} \geq 0$ by virtue of (3.1b) we can make a partition (3.3) with the property (3.4) by taking as b_3 the nonnegative part of the starting right hand side b^o.

Finally let us prove the Lemma for any other time step. The proof is by induction and generalises the arguments given for the implicit scheme [2].

For $m = 1$ the Lemma is true. Let us show it holds also for $m = 2$. Using eqs. (2.15) we get:

$$b^2 = b^1 - Bu^1 \tag{3.9}$$

where u^1 is the solution to the LCPs (2.14) for $m = 1$. Since $A_{33}u_3^1 = b_3^1$ we have:

$$A_{33}^{-1}b_3^2 = u_3^1 - A_{33}^{-1}B_{33}u_3^1 \tag{3.10}$$

Using inequality (2.12) and identity (2.10) we get:

$$-A^{-1}B \geq \frac{1}{\theta}[\frac{1}{\alpha} - (1-\theta)]\mathbb{1}. \tag{3.11}$$

The right hand side is positive for

$$\alpha \leq \frac{1}{1-\theta} \tag{3.12}$$

in which case:

$$A_{33}^{-1}b_3^2 \geq 0. \tag{3.13}$$

Now, from eq. (3.9) we have $b_1^2 < 0$ and

$$b_2^2 = b_2^1 + (1-\theta)\rho\tilde{q}e_1^T u_3^1 \leq= b_2^1 + \theta\rho\tilde{q}e_1^T u_3^1 = -s_2^1 \leq 0 \tag{3.14}$$

otherwise $s_2^1 < 0$ which is in contradiction with the fact that the "While do loop" of the algorithm AOPT was terminated for the first time step. Therefore, the Lemma is proven for $m = 2$.

From the Algorithm AOPT it is clear that the feasible basic solution is updated by positive increments. Therefore one has:

$$u_3^2 \geq A_{33}^{-1}b_3^2 \tag{3.15}$$

from which it follows that:

$$u^2 \geq u^1 \tag{3.16}$$

Until now we have proven the Lemma for $m = 1, 2$ and the following *monotonicity*:

$$u^m \geq u^{m-1}, \quad m = 1, 2 \tag{3.17}$$

Finally let us assume that the Lemma and the above monotonicity are true at the time slice $m-1$ and show that they hold at the time slice m. Again using eqs. (2.15) we get:

$$b^m = b^{m-1} - B(u^{m-1} - u^{m-2}) \tag{3.18}$$

and since $A_{33}u_3^{m-1} = b_3^{m-1}$ we have:

$$A_{33}^{-1}b_3^m = u_3^{m-1} - A_{33}^{-1}B_{33}(u_3^{m-1} - u_3^{m-2}) \tag{3.19}$$

Using the monotonicity at $m-1$, i.e. $u_3^{m-1} \geq u_3^{m-2}$ and inequalities (3.11,3.12) it follows that $A_{33}^{-1}b_3^m \geq 0$.

From eq. (3.18) follows that $b_1^m < 0$ and

$$b_2^m = b_2^{m-1} + (1-\theta)\rho\tilde{q}e_1^T u_3^{m-1} \leq= b_2^m + \theta\rho\tilde{q}e_1^T u_3^{m-1} = -s_2^{m-1} \leq 0 \tag{3.20}$$

otherwise $s_2^{m-1} < 0$ which is in contradiction with the fact that the "While do loop" of the algorithm AOPT was terminated for the time slice $m-1$. Therefore, the Lemma is proven for this time slice.

Since the feasible basic solution is updated by positive increments we have:

$$u_3^m \geq A_{33}^{-1}b_3^m \tag{3.21}$$

Using eq. (3.19) we get:

$$u^m \geq u^{m-1} \tag{3.22}$$

i.e. the monotonicity is proven also for this time slice. ■

Remark 3.4 *The monotonicity property (3.22) that we showed in the discrete case is inherited from the monotonicity property of the continuous value function [16]. It is proven also by [14] in the context of their VI derivation of the LCP.*

Remark 3.5 *We have shown that the right hand side for the* 1^{st} *time slice has a certain sign structure for* $\Delta\xi$ *satisfying the inequality (3.8). In fact, for a zero risk-free interest rate* r *this condition it is not fulfilled. As a consequence the* 1^{st} *time slice right hand side is nonnegative and there are no LCPs to be solved. In this case the option is optimaly held to maturity.*

Remark 3.6 *From lemma 3.2 it is clear that the starting complementary partition may be given by the last complementary partition of the previous time step. This simplifies Algorithm 1 and gives Algorithm 2.*

Remark 3.7 *The restriction (3.12) imposed in the proof requires a closer analysis. Substituting* α *in (3.12) we get a restriction on the mesh ratio:*

$$r\Delta t + \rho \leq \frac{1}{1-\theta} \tag{3.23}$$

which is $\rho \leq 2 - r\Delta t$ *for* $\theta = \frac{1}{2}$*. Using the definition of* ρ *(2.8) we get* $\Delta t \sim (\Delta\xi)^2$*, which means that we are not free to choose* Δt *and* $\Delta\xi$ *independently. However, numerical examples show that the algorithm is valid for any value of* ρ*, although the monotonicity (3.22) is lost for very large* ρ *values. We will get back to this discussion in the next section.*

For illustration we have used Algorithm 2 to evaluate an American put option which expires in one year, i.e. $T = 1$ with a strike price $K = 1$. Risk-free interest rate is assumed to be $r = 10\%$ and volatility of the underlying $\sigma = 20\%$. In Fig. 1 we show the efficient frontier or the exercise price of the option using the Crank-Nicolson (CN) discretization scheme with $\rho = 1$. In Fig. 2 we show the option price which is computed using the implicit discretization scheme.

4 Implementation and run times

Algorithm 2 is easy to implement. Special care is needed for the implementation of A_{33} invertions, which are the most intensive part of the computation. Note that matrix

Algorithm 2 Algorithm AOPT

Find n_b such that $b_1^o, b_2^o < 0$ and $b_3^o \geq 0$

for $m = 1, \ldots, M$ **do**

  $y = \gamma\ e_1^T A_{33}^{-1} e_1,\ z = e_1^T A_{33}^{-1} b_3$

  if $n_b = 0$ **then**

    $s_2 := 0$

  else

    $s_2 = -(b_2 + \beta z)$

  end if

  while $s_2 < 0$ **do**

    $n_b := n_b - 1$

    $z = -s_2/(\alpha - \beta y)$

    if $n_b = 0$ **then**

      $s_2 := 0$

    else

      $s_2 = -(b_2 + \beta z),\ y := \gamma/(\alpha - \beta y)$

    end if

    $u_3^m = A_{33}^{-1} b_3,\ u_1^m := 0, u_2^m := 0$

    $b := b^o - Bu^m$

  end while

end for

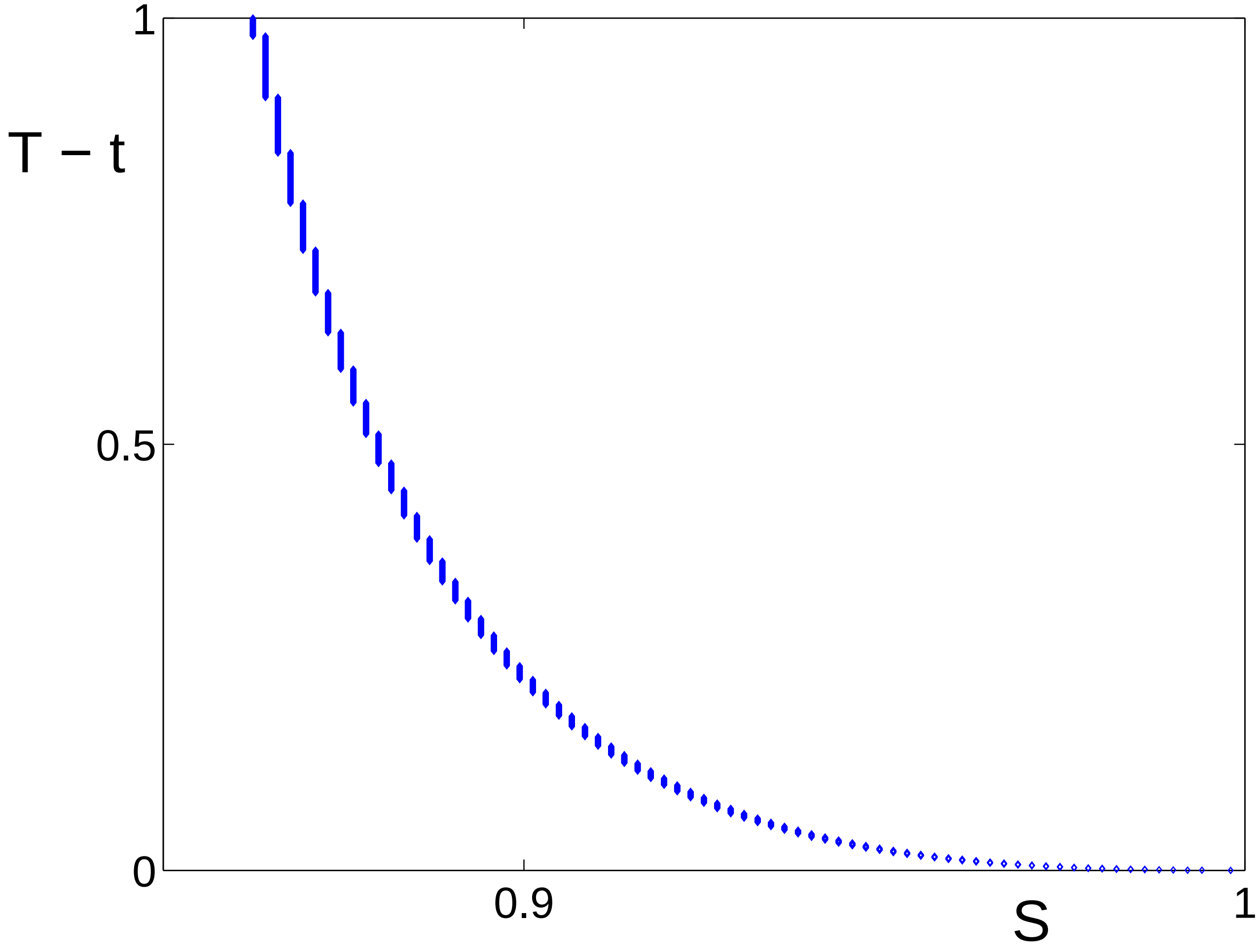

Figure 1: Efficient frontier of an American put option computed by the AOPT algorithm described in the text using the Crank-Nicolson discretization scheme. The option parameters are: interest rate $r = 0.1$, volatility $\sigma = 0.2$, strike price $K = 1$ and maturity $T = 1$. The discretization parameters are: $L = -4, U = 4, I = 4000, M = 10000$

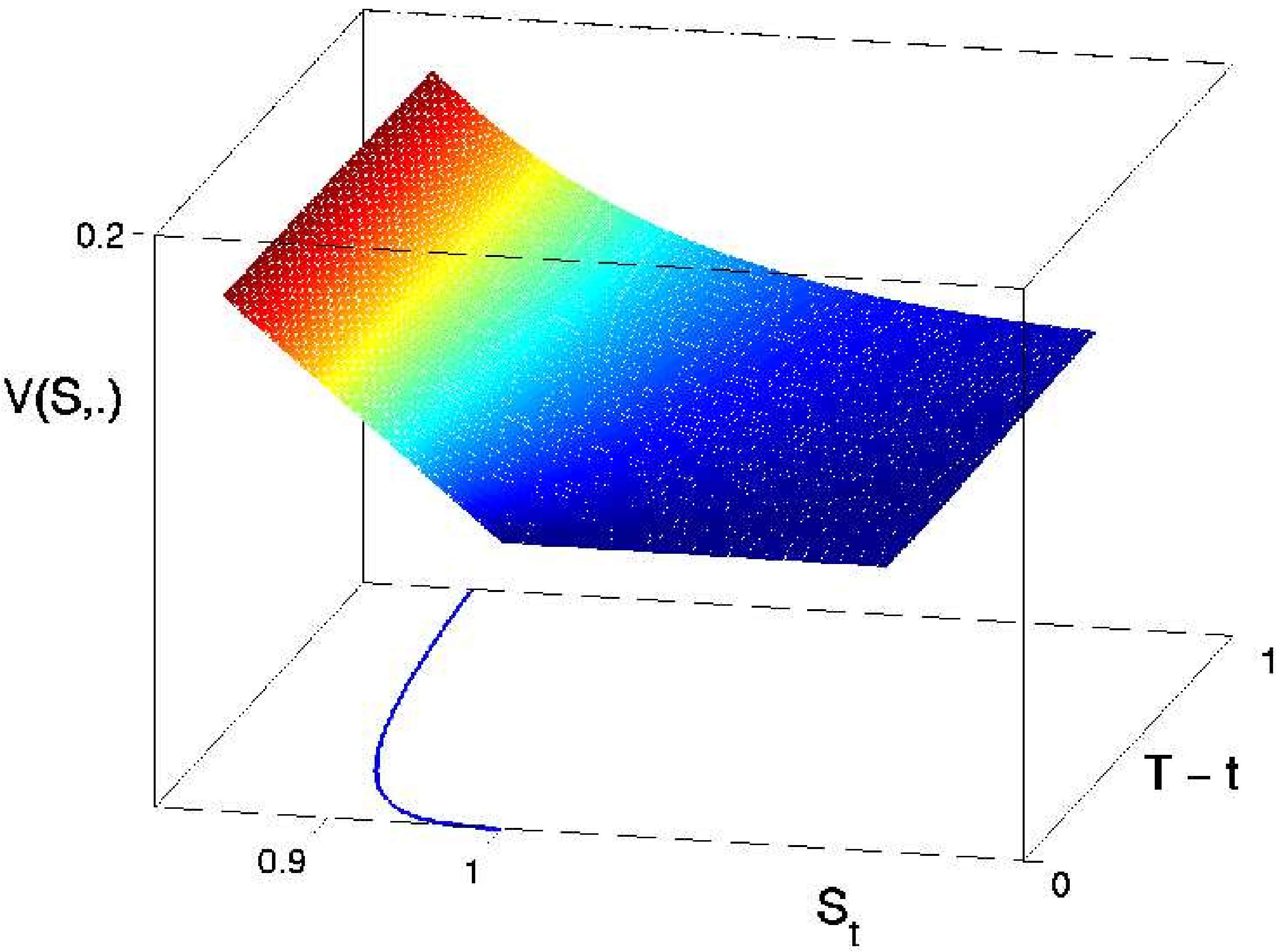

Figure 2: Value function of an American put option computed by the AOPT algorithm described in the text using the implicit discretization scheme. The option parameters are: interest rate $r = 0.1$, volatility $\sigma = 0.2$, strike price $K = 1$ and maturity $T = 1$. The discretization parameters are: $L = -.3, U = .7, I = 4000, M = 100$

elements are time and stock price independent. Since A is tridigonal its LU-decomposition can be written as:

$$\begin{pmatrix} \alpha & -\beta & & \\ -\gamma & \alpha & \ddots & \\ & \ddots & \ddots & -\beta \\ & & -\gamma & \alpha \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ & & l_n & 1 \end{pmatrix} \begin{pmatrix} u_1 & -\beta & & \\ & u_2 & \ddots & \\ & & \ddots & -\beta \\ & & & u_n \end{pmatrix} \tag{4.1}$$

We computed this only once at the begining and stored arrays $l_2, \ldots, l_n$ and $u_1, \ldots, u_n$. Then for the A_{33} invertions inside the time loop, depending on the value of n_b, we used only $l_2, \ldots, l_{n-n_b}$ and $u_1, \ldots, u_{n-n_b}$ elements. The lower and upper bidiagonal matrices are inverted using forward and backward substitutions respectively.

We compared the run times of the new algorithm to those of PSOR algorithm [7], which is widely used for American option pricing. The PSOR routine is given here as Algorithm 3, which can be called by the pricing algorithm.

Algorithm 3 PSOR algorithm

Given tol $\in \mathbb{R}_+$

for $k = 1, 2, \ldots$ **do**

  for $i = 1, n$ **do**

    $r_i^{(k+1)} = b_i - \alpha u_i^{(k)} + \beta u_{i+1}^{(k)} + \gamma u_{i-1}^{(k+1)}$

    $u_i^{(k+1)} = \max\{0, u_i^{(k)} + \omega r_i^{(k+1)}/\alpha\}$

    $s_i^{(k+1)} = -r_i^{(k+1)} + \alpha(u_i^{(k+1)} - u_i^{(k)})$

  end for

  Stop if $||u^{(k+1)} - u^{(k)}||_\infty < \text{tol}$

end for

Note that boundary conditions $u_0 = u_{n+1} = 0$ apply and ω is the overlaxation parameter with values in the interval $(0, 2)$. The algorithm has the fastest convegence for an optimal value, ω_{opt} [1]. Sometimes it is difficult to estime it from theoretical considerations and very often ω is tuned in order to get a faster convergence. This is a disadvantage for PSOR compared to parameter-free algorithms.

Table 1: Comparison of AOPT and PSOR algorithms.
Run times in seconds on a 3 GHz Intel Pentium 4 processor

$M = 1000$

I	ρ	ω	PSOR	AOPT
2000	10	1.5	3.56	1.17
3000	22.5	1.6	8.26	1.81
4000	40	1.7	15.26	2.43
5000	62.5	1.75	27.58	3.02

$M = 4000$

I	ρ	ω	PSOR	AOPT
4000	10	1.5	28	17
6000	22.5	1.8	130	26
8000	40	1.85	195	35
10000	62.5	1.85	312	45

In Table 1 we compare directly the new algorithm to the PSOR algorithm. Option parameters are $T = 1$, $K = 1$, $r = 10\%$ and $\sigma = 20\%$. The domain of the logarithmic price is the closed interval $[-1, 3]$. The firgues show that AOPT always wins. Moreover, as the accuracy is increased AOPT algorithm runs much faster than PSOR. For example, in the upper panel of Table 1 ($M = 1000$) the gain increases from a factor 2 for $I = 2000$ to a factor 9 for $I = 5000$. In the lower panel ($M = 4000$) the gain increases from 50% for $I = 4000$ to a factor 7 for $I = 10000$.

Note that we have taken ρ that violates the restriction (3.12) imposed in the proof of Lemma 3.2. In this case one should modify Algorithm 2 to include an earlier termination if $A_{33}^{-1}b_3^m < 0$ before the next LCP is solved. If this does not happen then the conditions of the Theorem 3.1 apply and the algorithm is valid. If such termination occurs one has to select discretisation parameters which correspond to a smaller mesh ratio ρ and restart the algorithm. We have not experienced a single termination of this type up to date. This hints that Lemma 3.2 should hold without restriction (3.12).

However, we caution in the use of very large ρ or coarse grids along the time direction. It is well known that in this case the CN scheme produces an oscillating solution [18]. Indeed, we observe that for such values of ρ monotonicity property (3.22) is lost. This contradicts the regularity properties of the option price [14]. Hence, one should check the monotonicty of the option price during the run. In case it is violated this is a sign that we are using a large ρ. As seen from Table 1 we could run the CN scheme without problem up to $\rho = 62.5$. We encountered monotonicity violation for $\rho = 90$ and $M = 1000$ only at the second time step (going backwards from maturity) and only at the underlying price $S = 1$. The recipe suggested by [18] is to increase the number of time step. Indeed, by taking $M = 4000$ we observe no monotonicity violation.

5 Complexity and accuracy of the algorithm

The most expensive part of the computation is the A_{33} invertion. As stated above the LU decomposition is performed at the begining and stored. Then the lower and upper bidiagonal matrices are inverted using forward and backward substitutions. Such proccesses need $3(n - n_b) - 2$ multiplications or divisions. There are three such invertions and operations inside *While do* loop. They cost no more than $9(n - n_b)$ multiplications. Adding here $3n$ multiplications to update the right hand side b, the whole algorithm does not cost more than $(12n - 9n_b)M$ operations in total, where n_b stands for the last updated n_b. Hence, Algorithm 2 has a complexity which is linear in both n and M or linear in the number of grid points. Our experiments support the theoretical complexity as can be seen from run times in Table 1. For example, whenever the number of spatial grid points I is doubled the run time is doubled as well.

The situation for the PSOR algorithm is more subtle. The algorithm performs $6n$ multiplications/divisions per iteration. If k_{mach} is the number of iterations needed for the algorithm to converge to machine percision, then the cost of the PSOR is $6nk_{mach}$ operations. The overall cost of the option pricing is then $6nMk_{mach}$, which is still linear in the number of grid points but linear also in k_{mach}. However, the timings in Table 1 suggest that if the number of spatial grid points I is doubled the run time of PSOR increases by more than a factor of four. In Fig. 3 we show the run times as a function of the normalized I^2. The lower line in the figure corresponds to the $M = 1000$ data in Table 1, whereas the upper line to $M = 4000$ data. Except an anomaly in the second point of the upper line, this graph suggests that run times of PSOR algorithm grow as a power law which is consistent to a quadratic law. This behaviour is in contrast to the new algorithm, which has only linear dependence on I.

Having a fast algorithm is one thing, but having an accurate solution is another. It is well known that tree methods are fast but only first order accurate. The same is true for the explicit scheme, which is mathematically equivalent to tree methods for small Δt. In contrast, the CN scheme is second order accurate in both variables [18]:

$$\text{Accuracy}(\theta = \frac{1}{2}) \sim O[(\Delta t)^2] + O[(\Delta \xi)^2]$$

Hence, this is the scheme of choice. In order to measure the accuracy of the difference scheme we must know the exact option price. For zero interest rate the American put option is optimally held to maturity. Therefore, its price can be computed exactly using the Black-Scholes formula. We measured the error as the inifinte norm of the deviation from the exact price.

In Table 2 we show run times of different discretization schemes for about the same accuracy. Option parameters are the same as in Table 1 but with $r = 0$. We see that the CN scheme wins for moderately large ρ.

Note that we implemented directly the explicit scheme, since the solution of LCP is trivial in this case. The complexity of the explicit scheme is $5nM$ operations. Hence, the explicit scheme runs faster than CN scheme. Nevertheless, to reach the same acuracy the

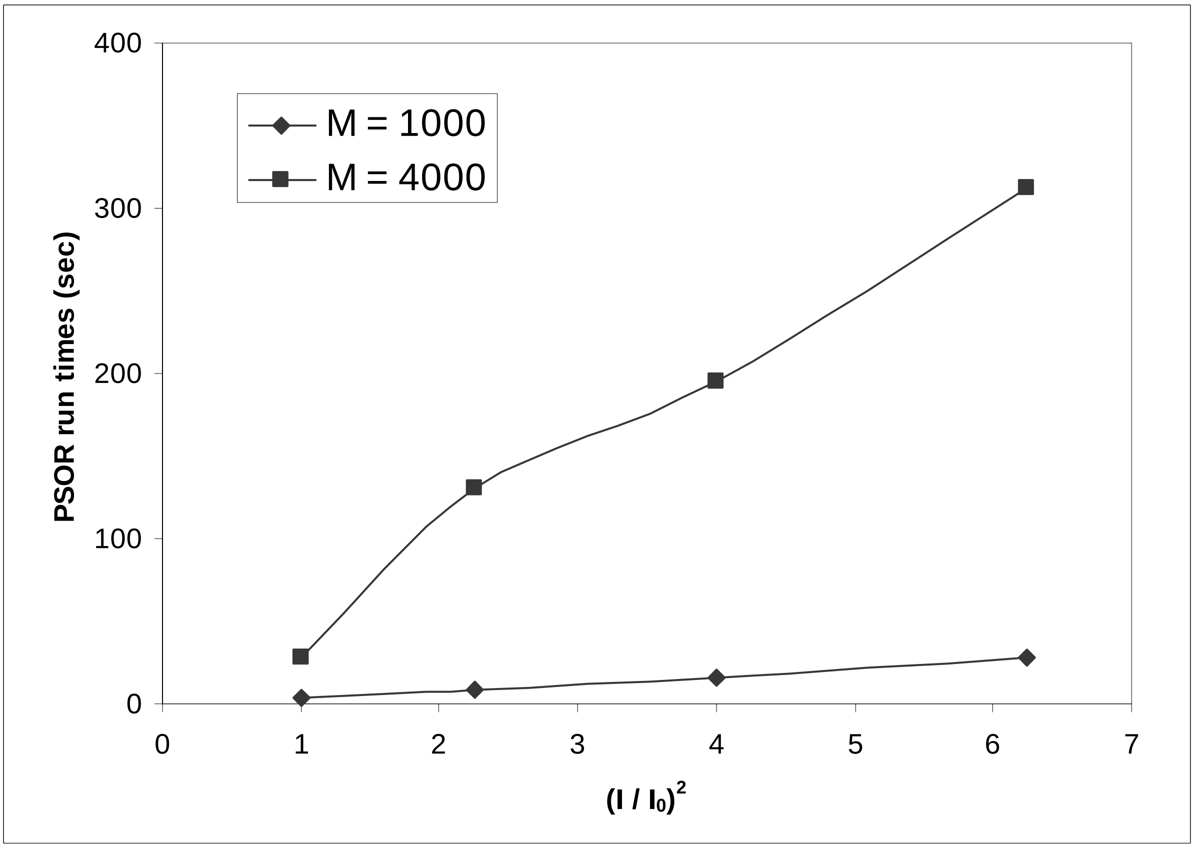

Figure 3: Run times of PSOR algorithm as a function of the spatial grid points squared. Note normalization with $I_0 = 2000$ for $M = 1000$ and $I_0 = 4000$ for $M = 4000$. Lines are drawn to guide the eye.

Table 2: Accuracy of difference schemes.

Run times in seconds on a 3 GHz Intel Pentium 4 processor

$I = 2000$

θ	ρ	Error	Run time (sec)
0.5	50	1.03×10^{-6}	0.11
0.5	40	1.02×10^{-6}	0.13
0	1	2.11×10^{-6}	0.79
0.5	10	1.06×10^{-6}	1.98
1	1	2.06×10^{-6}	7

$I = 4000$

θ	ρ	Error	Run time (sec)
0.5	50	3.22×10^{-7}	2.82
0.5	40	3.23×10^{-7}	3.95
0	1	6.15×10^{-7}	10.6
0.5	10	3.24×10^{-7}	24
1	1	5.56×10^{-7}	71.39

explicit scheme is slower than the CN scheme. As can be seen from Table 2 the explicit scheme is a factor of 7 slower than the CN scheme for $I = 2000$. This factor is about 4 for $I = 4000$.

We have shown that American options can be evaluated accurately by algorithms which scale linearly with the number of grid points. On the other hand such an application represents an imprtant class of structural LCP problems that can be solved in linear time. Coming to this result has required a long time of interdisciplinary research in both optimization and PDEs.

Experience shows that there is no black-box solution to PDE numerical solutions. For example, Crank-Nicolson scheme is second order accurate but caution is in order for large time steps. This is demonstrated once more in the evaluation of American options, where the monotonicity of the option price may be violated. Other second order discretisation schemes, like multi-level schemes exist [18]. Whether they are more robust and offer more flxibility in choosing the time step size is a question that should be investigated in the future.

References

[1] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, Ch. Romine and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd. ed., SIAM, 1994, Philadelphia

[2] A. Borici and H.-J. Lüthi, *Pricing American Options by Fast Solutions of the Linear Complementarity Problem*, in Computational Methods in Decision-Making, Economics and Finance (Eds. E. Kontoghiorghes , B. Rustem and S. Siokos), pp. 325 -338, 2002

[3] J.M.Borwein and M.A.H. Dempster, *The linear order complementarity problem*, Math. of Operations Research 14 (1989) 534-558

[4] R. Chandrasekaran, *A special case of the complementary pivot problem*, Opssearch (1970) 263-268

[5] R.W. Cottle and J-S. Pang and R.E. Stone, *The Linear Complementarity Problem*, Academic Press, Inc. 1992.

[6] J. C. Cox, S. Ross and M. Rubinstein, *Option Pricing: A Simplified Approach*, Journal of Financial Economics, 7(3):229-263, 1979.

[7] C.W. Cryer, *The solution of a quadratic programme using systematic overrelaxation*, SIAM J. Control and Opt. 9 (1971) 385-392

[8] M.A.H. Dempster and J.P.Hutton, *Pricing American stock options by linear programming*, Mathematical Finance 9 (1999) 229-254

[9] M.A.H. Dempster, J.P. Hutton & D.G. Richards, *LP valuation of exotic American options exploiting structure*, J. Computat. Finance 2 (1998) 61-84

[10] J.N. Dewynne, S.D. Howison, I. Rupf and P. Wilmott, *Some mathematical results in the pricing of American Options*, Europ. J. Appl. Math. 4, 381-398, 1993

[11] M. Fiedler and V. Ptak, *On matrices with non-positive off-diagonal elements and positive principal minors*, Czechoslovak Math. J. 12 (1962) 382-400

[12] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, 1989

[13] J. Huang and J-S. Pang, *Option pricing and linear complementarity*, J. Computat. Finance 2 (1998) 31-60

[14] P. Jaillet, D. Lamberton, B. Lapeyre, *Variational Inequalities and the pricing of American Options*, Acta Applicandae Mathematicae, 21 263-289, 1990

[15] K.G. Murty, *On the number of solutions to the complementarity problem and spanning properties of complementary cones*, Linear Alg. and its Appl. 5 (1972) 65-108

[16] R. Myneni, *The pricing of the American option*, Annals of Appl. Probability 2 (1992) 1-23

[17] L.C.G. Rogers and D. Talay, editors of *Numerical Methods in Finance*, Cambidge University Press 1997.

[18] D. Tavella and K. Randall, *Pricing financial instruments: the finite difference method*, 2000, John Wiley & Sons, Inc.