

To appear in the *International Journal of Computer Mathematics*
Vol. 00, No. 00, Month 20XX, 1–19

Moving Boundary Transformation for American Call Options with Transaction Cost: **Finite Difference** Methods and Computing

V.N. Egorova^{a *}, S.-H. Tan^b, C.-H. Lai^{b *}, R. Company^a and L. Jódar^a

^a*Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera s/n, Valencia, Spain;* ^b*University of Greenwich, Park Row, Greenwich, London, UK*

(v2.3 released April 27, 2015)

The pricing of American call option with transaction cost is a free boundary problem. Using a new transformation method the boundary is made to follow a certain known trajectory in time. The new transformed problem is solved by various finite difference methods, such as explicit and implicit schemes. Broyden's and Schubert's methods are applied as a modification to Newton's method in the case of nonlinearity in the equation. An Alternating Direction Explicit (ADE) method with second order accuracy in time is used as an example in this paper to demonstrate the technique. Numerical results demonstrate the efficiency and the rate of convergence of the methods.

Keywords: nonlinear PDE; free boundary; transformation; finite difference methods; Newton-like method; alternating direction explicit method

2010 AMS Subject Classification:

1. Introduction

Pricing American option is a problem often studied in the field of computational finance. The well-known Black-Scholes (BS) model [5] provides an easy computable pricing formula of European option in an idealistic market. Such assumptions are not realistic [6], [24], and do not take into account, for instance, transaction costs, investor's preferences, feedback and illiquid markets effects. This has motivated the development of new nonlinear models to take account of various realistic trading environment.

Leland [24] proposed a BS formula with an augmented volatility due to transaction cost. Authors in [3], [16] presented an adjusted volatility which depends on the sign of the gamma of the option to control effectively the hedging risk and transaction cost. These ideas led to a nonlinear BS equation for European options. Barles and Soner [4] proposed a more complicated nonlinear model by assuming that investor's preferences are characterized by an exponential utility function. Later risk adjusted pricing methodology (RAPM) was proposed by Kratka [19] and revisited by Jandačka and Ševčovič in [18]. Note that all the above mentioned nonlinear models are consistent with the original BS equation in the case when the additional parameters are vanishing.

In general there is no closed form solution for nonlinear American or European option pricing problem. Therefore numerical methods are usually employed to solve them. For European options numerical methods have been developed by several authors in recent years [10], [14], [31], etc. For American options the main difficulty is the existence of the

*Corresponding author. Email: egorova.vn@gmail.com

*Corresponding author. Email: C.H.Lai@greenwich.ac.uk

unknown optimal stopping boundary. One way to overcome this difficulty is to present it as a nonlinear complementarity problem (NCP) arising from the discretisation of the free boundary problem. In [15] and [25] the penalty approach is proposed to solve the NCP by approximating it using an algebraic system of nonlinear equations containing a power penalty term.

A common alternative approach to NCP that is able to remember the free boundary while solving the problem is the so-called front-fixing method [11], based on the transformation of the original equation into a new one defined on a fixed domain. The unknown free boundary is calculated as an additional unknown function involved in the PDE problem. Although free boundary problems originated in physics, this technique has been used in computational finance since 1998 [36].

For the case of American options with constant volatility various front-fixing transformations have been studied in [7], [21], [29], [32]. Ševčovič proposed a fixed domain transformation for nonlinear American option pricing problem [33]. Further, this method was studied in some recent papers (see [2], [12]). Since the transformed equation contains a strong convective term the operator splitting method is used to overcome numerical difficulties. Moreover, in order to close up the system of equations that determines the value of a new function an additional equation for the free boundary position is required.

In this paper a new transformation in the framework of a front-fixing method is proposed. Under the transformation the free boundary is replaced by time-dependent known boundary. In the resulting equation there is no reaction term and the convection term is simplified in a such way that the operator splitting technique is not required. This ensured a single numerical scheme is suitable for the entire equation. The connection between the transformed boundary conditions with the transformed option price and the free boundary does not require additional information.

The proposed formulation of the nonlinear problem allows the use of a versatile numerical treatment. In this paper an explicit Euler and alternating direction explicit (ADE) method [13], [30] together with implicit methods were studied. Dealing with implicit methods one has to solve nonlinear system. In this paper Newton's method with suitable modifications to improve its efficiency and in saving computational cost [22] were examined.

The paper is organized as follows. Section 2 presents the transformation of the nonlinear models of Barles and Soner [4] and RAPM [18] for American option pricing. Section 3 presents the discretisation of the transformed option pricing problem and the related numerical methods. Section 4 provides the numerical results and their comparison. In Section 5 conclusion and discussion of the results are presented.

2. Transformation of the Nonlinear American Option Pricing Problem

The transformation of the free boundary American option pricing problem into another nonlinear PDE problem, such that the free boundary is written in terms of another variable with a known moving boundary, is presented.

Nonlinear American call option pricing models may be presented as the free boundary PDE problem

$$C_\tau = \frac{\tilde{\sigma}^2}{2} S^2 C_{SS} + (r - q) S C_S - r C, \quad 0 \leq S < B(\tau), \quad 0 < \tau \leq T, \quad (1)$$

where the adjusted volatility function is given by

$$\tilde{\sigma}^2 = \tilde{\sigma}^2(\tau, S, C_{SS}). \quad (2)$$

Here $\tau = T - t$ denotes the time to maturity T , S is the asset price, $C(S, \tau)$ is the option price, $B(\tau)$ is the unknown early exercise boundary, σ_0 is the implied volatility of the asset, r is the risk free interest rate, q is the continuous dividend yield and E is the strike price.

The boundary and initial conditions for an American call option problem are (see [35])

$$C(S, 0) = \max(S - E, 0), \quad (3)$$

$$\frac{\partial C}{\partial S}(B(\tau), \tau) = 1, \quad (4)$$

$$C(B(\tau), \tau) = B(\tau) - E, \quad (5)$$

$$C(0, \tau) = 0, \quad (6)$$

$$B(0) = \begin{cases} E, & r \leq q, \\ \frac{r}{q}E, & r > q. \end{cases} \quad (7)$$

It is well known that if there is no dividend payment ($q = 0$), then the optimal strategy is to exercise option at the maturity (see [35], [17]). In that case the American call becomes an European call. Due to this reason $q > 0$ [17] is used in the problem defined in (1)-(7).

In the following study, two nonlinear models with different adjusted volatility functions (2) are considered. First the strong nonlinear RAPM model, where $\tilde{\sigma}^2$ is a cubic root function. Second the widely used Barles and Soner's model in which the adjusted volatility function is obtained through the solution of an ordinary differential equation.

Under the RAPM model the volatility σ is a function of the asset price (S) and the second derivative of the option price ($\frac{\partial^2 C}{\partial S^2}$), i.e.

$$\tilde{\sigma}^2 = \sigma_0^2 \left(1 + \mu \left(S \frac{\partial^2 C}{\partial S^2} \right)^{\frac{1}{3}} \right), \quad (8)$$

where σ_0^2 is a constant historical volatility of the asset and μ is a nonnegative constant.

Barles and Soner introduced a nonlinear Black-Scholes equation with an adjusted volatility [4] which is a function of the second derivative of the price itself, i.e.

$$\tilde{\sigma}^2 = \sigma_0^2 (1 + \Psi(e^{r\tau} a^2 S^2 C_{SS})), \quad (9)$$

where $a = \mu\sqrt{\gamma N}$, γ is the risk aversion factor and N is the number of options to be sold. The function Ψ is the solution of the nonlinear singular initial-value problem

$$\Psi'(A) = \frac{\Psi(A) + 1}{2\sqrt{A\Psi(A) - A}}, \quad A \neq 0, \quad \Psi(0) = 0. \quad (10)$$

From the Theorem 1.1 of [9] it is known that $\Psi(A)$ is an increasing function mapping

the real line onto the interval $] - 1, +\infty]$ and is implicitly defined by

$$A = \left(-\frac{\operatorname{arcsinh} \sqrt{\Psi}}{\sqrt{\Psi+1}} + \sqrt{\Psi} \right)^2, \quad \Psi > 0; \quad (11)$$

$$A = -\left(\frac{\operatorname{arcsin} \sqrt{-\Psi}}{\sqrt{\Psi+1}} + \sqrt{-\Psi} \right)^2, \quad -1 < \Psi < 0. \quad (12)$$

The case for Barles and Soner's model is a slightly complicated one in terms of numerical implementation. By using numerical examples it is able to demonstrate that the proposed method may be used to handle other models with nonconstant volatility.

Taking advantages of Landau transformation [23] with modifications in the exponential factors like those described in [10], it is possible to remove the reaction term and partially the convection term by using the transformation given below.

$$x = e^{(r-q)\tau} \frac{S}{B(\tau)}, \quad V(x, \tau) = \frac{e^{r\tau}}{E} C(S, \tau), \quad S_f(\tau) = \frac{B(\tau)}{E}. \quad (13)$$

Using transformation (13) the equation (1) takes the form

$$V_\tau = \frac{\sigma^2}{2} x^2 V_{xx} + \frac{S'_f}{S_f} x V_x, \quad 0 \leq x < e^{(r-q)\tau}, \quad 0 < \tau \leq T, \quad (14)$$

where

$$\sigma^2 = \sigma^2(\tau, x, V_{xx}) = \tilde{\sigma}^2(\tau, S, C_{SS}), \quad (15)$$

with new initial and boundary conditions

$$S_f(0) = \max\left(\frac{r}{q}, 1\right), \quad (16)$$

$$V(x, 0) = \max(x S_f(0) - 1, 0), \quad (17)$$

$$V(0, \tau) = 0, \quad (18)$$

$$V(e^{(r-q)\tau}, \tau) = e^{r\tau} (S_f(\tau) - 1), \quad (19)$$

$$V_x(e^{(r-q)\tau}, \tau) = e^{q\tau} S_f(\tau). \quad (20)$$

Note that the transformation described in (13) transformed the original free boundary value problem to a known moving boundary problem. In the case $r > q$ the computational domain increases with respect to time, otherwise it decreases.

In the problem (14) - (20) there are two sources of nonlinearity. First, the additional unknown function (free boundary) in the equation (14). The method to handle this problem relies on the choice of the finite difference method and is explained in section 3.

Second, the volatility σ is nonlinear. With the moving domain transformation (13) argument of the function σ in RAPM model changes and is given below.

$$\sigma^2 = \sigma_0^2 \left(1 + \mu \left(x V_{xx} \frac{e^{-q\tau}}{S_f(\tau)} \right)^{\frac{1}{3}} \right). \quad (21)$$

Table 1. CPU-time (sec) for linear and binary search.

Number of calls	10 ³	10 ⁴
Linear search	5.022	47.600
Binary search	0.062	0.707

For Barles and Soner's model σ is transformed to

$$\sigma^2 = \sigma_0^2 (1 + \Psi(a^2 Ex^2 V_{xx})). \quad (22)$$

3. Numerical Methods

This section begins with an algorithm computing the implicitly adjusted volatility function Ψ given by Barles and Soner's model. It follows with a description of three finite difference methods of solving the transformed problem described in equation (14). These finite difference methods include an explicit Euler method, an alternating direction explicit method and an implicit method.

Note that the numerical evaluation of the adjusted volatility function for the RAPM model is straight forward as an explicit function of the volatility is defined. In the case of Barles and Soner's model the volatility function is given in terms of the solution Ψ of the ODE (10). It is well known that MATLAB built-in solver for ODE [1], [34] may be used, or in some other cases simply take $\Psi(A) = A$. One way to avoid the additional errors due to the numerical solution of the ODE is to make use of the implicit solution (11)-(12), given in [9], through an interpolation procedure as proposed in the numerical algorithm below.

Algorithm 1: Computing Ψ using Barles and Soner's model.

Data: $L :=$ Desired range of Ψ ; $N_\Psi :=$ Number of discrete points for Ψ ;
 $\Delta\Psi = L/(N_\Psi - 1)$;

Result: $\Psi(A)$

- 1 Discretise Ψ as Ψ_j , $j = 1, 2, \dots, N_\Psi$;
 - 2 Compute a_j at Ψ_j , using formulae (11)-(12);
 - 3 Compute the argument A of Ψ at every spatial finite difference node using (22);
 - 4 For all values of A : Search J such that $A \in [a_J, a_{J+1}]$;
 - 5 Obtain an approximate value of $\Psi(A)$ using interpolation.
-

There exists many search algorithms in literature. It is very important to choose an appropriate one. The **simplest** and widely used **one** is the linear search. It is a method for finding a particular value (key) in an array that checks each element in sequence until the desired element is found or the list is exhausted [20]. The cost of the worst case is proportional to the number of elements in the array. Since function $\Psi(A)$ is increasing, the list of search is also an increasing sequence, i.e. it is sorted. Therefore binary search can be used. In each step, the algorithm compares the key value with the middle element of the list. If the values match, then a matching element has been found. Otherwise, if the search key is less than the middle element, then the algorithm repeats its action on the sub-array to the left of the middle element or, if the search key is greater, on the sub-array to the right. Binary search takes logarithmic time (see [20], p. 414).

Both algorithms were tested in this paper. For the search of just one element (i.e. calculate one value of σ), both algorithms work with similar speed. The difference becomes

noticeable when the procedure repeats several times. Results of the tests are presented in the Table 1. Number of calls there means the number of repetition of the procedure.

In the finite difference methods described below for the solutions of (14)-(20), the temporal axis takes a uniform partition with the time step $k = \frac{T}{N}$. Each time level is denoted as $\tau^n = nk$, $0 \leq n \leq N$. Let the right boundary of the domain be denoted as $x_{\max}^n = e^{(r-q)\tau^n}$. The spatial step size h_n and grid point x_j^n at time level τ^n are defined as

$$h_n = \frac{e^{(r-q)nk}}{M}, \quad x_j^n = jh_n, \quad 0 \leq j \leq M, \quad (23)$$

where M is the number of spatial grid points.

Denote the approximate value of the solution $V(x, \tau)$ at the point x_j^n and time τ^n as $u_j^n \approx V(x_j^n, \tau^n)$ and the approximate value of the free boundary as $S_f^n \approx S_f(\tau^n)$. As the problem itself has a moving boundary which means that the spatial finite difference mesh would have to be rearranged at every time level. A typical spatial mesh with grid points (x_j^n, τ^n) at time $\tau = nk$ does not remain as a grid point at time $\tau = (n+1)k$. Fig. 1 illustrates the nodal points (x_j^n, τ^n) as black dots and those corresponding nodes as white dots and the spatial finite difference mesh remains unchanged, and the corresponding moveable nodes in black dots at the next time level. The pair (x_j^n, τ^{n+1}) is known as a non-grid point for this purpose.

3.1 Explicit Euler method

Let the approximation at the non-grid point (x_j^n, τ^{n+1}) be denoted as $\tilde{u}_j^{n+1} \approx V(x_j^n, \tau^{n+1})$. In order to calculate the value at a new time level an orthogonal 4-point stencil $u_{j-1}^n, u_j^n, u_{j+1}^n$, and \tilde{u}_j^{n+1} is used following the forward difference approximation of the temporal derivative of V at (x_j^n, τ^n) is given by

$$\frac{\partial V}{\partial \tau}(x_j^n, \tau^n) \approx \frac{\tilde{u}_j^{n+1} - u_j^n}{k}. \quad (24)$$

Once the approximations $\{\tilde{u}_j^{n+1}\}$ are computed $\{u_j^{n+1}\}$ may be obtained by using a Lagrange interpolation from the resulting data $\{\tilde{u}_j^{n+1}\}$.

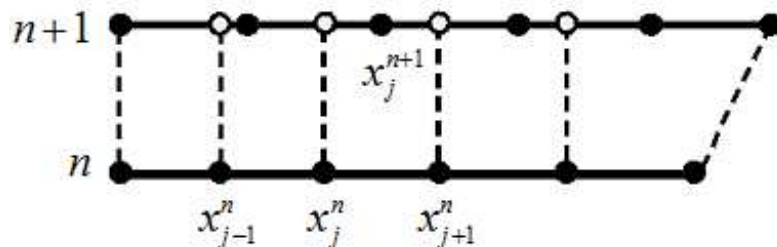


Figure 1. Moving grid. Black points are from a mesh and white points are out of the grid.

For the numerical solution of the problem (14)-(20) an explicit finite difference scheme based on a central difference scheme for the spatial derivatives and a forward difference

scheme, as described in (24), for the temporal derivative is constructed as follows

$$\frac{\tilde{u}_j^{n+1} - u_j^n}{k} = \frac{\sigma_{j,n}^2}{2} (x_j^n)^2 \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{h_n^2} + \frac{S_f^{n+1} - S_f^n}{k S_f^n} x_j^n \frac{u_{j+1}^n - u_{j-1}^n}{2h_n}, \quad (25)$$

where $\sigma_{j,n}^2 = \sigma(x_j^n, \tau^n)$ is computed by using either the RAPM or the Barles and Soner's model. For the RAPM model σ^2 is calculated as follows

$$\sigma_{j,n}^2 = \sigma_0^2 \left(1 + \mu \left(x_j^n \frac{e^{-qnk}}{S_f^n} \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{h_n^2} \right) \right). \quad (26)$$

In the case of Barles and Soner's model

$$\sigma_{j,n}^2 = \sigma_0^2 \left(1 + \Psi \left(E a^2 (x_j^n)^2 \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{h_n^2} \right) \right). \quad (27)$$

From the boundary conditions (19), (20) one obtains

$$u_0^n = 0, \quad u_M^n = e^{r\tau^n} (S_f^n - 1), \quad (28)$$

$$\frac{u_M^n - u_{M-1}^n}{h_n} = e^{q\tau^n} S_f^n \Rightarrow u_{M-1}^n = e^{q\tau^n} S_f^n \left(e^{(r-q)\tau^n} - h^n \right) - e^{r\tau^n}. \quad (29)$$

The initial conditions are discretised as follows

$$S_f^0 = \max \left(\frac{r}{q}, 1 \right), \quad u_j^0 = \max (S_f^0 x_j^0 - 1, 0), \quad j = 0, \dots, M. \quad (30)$$

Using the scheme (25) for $j = M - 1$ and equation (29) at the $(n + 1)$ -th time level, the expression for S_f^{n+1} takes the form

$$S_f^{n+1} = \varrho(u^n, S_f^n) = \frac{u_{M-1}^n + \frac{\sigma^2(M-1,n)k}{2h_n^2} (x_{M-1}^n)^2 (u_{M-2}^n - 2u_{M-1}^n + u_M^n) - x_{M-1}^n \frac{u_M^n - u_{M-2}^n}{2h_n} + e^{r\tau^n}}{e^{q\tau^n} (e^{(r-q)\tau^n} - h_{n+1}) - x_{M-1}^n \frac{u_M^n - u_{M-2}^n}{2h_n S_f^n}}. \quad (31)$$

Assembling all these ideas leads to the following algorithm.

Remark 1. Linear interpolation is used in order to preserve the second order accuracy of the approximations of the spatial derivatives.

Remark 2. With the interpolation one needs to guarantee that new grid point $x_j^{n+1} \in [x_j^n, x_{j+1}^n]$. In the case $r > q$ if

$$jh_n \leq jh_{n+1} < (j + 1)h_n, \quad (33)$$

one has from the definition (23),

$$0 \leq j \left(e^{(r-q)k} - 1 \right) < 1, \quad \forall j. \quad (34)$$

Algorithm 2: Explicit Euler method

Data: Initial values using (17);
Result: Solution at $\tau = T$;

```

1 n=0;
2 while n < N do
3   Compute  $\sigma_{j,n}^2$  for  $j = 1, \dots, M - 1$  using Alg. 1;
4   Compute  $S_f^{n+1}$  by (31);
5   Compute  $u_0^{n+1}, u_{M-1}^{n+1}, u_M^{n+1}$  at the boundary points using the boundary
   conditions (28), (29);
6   for  $j = 1, \dots, M - 2$  do
7     Obtain  $\tilde{u}_j^{n+1}$ ;
8   end
9   Construct new uniform grid: for  $j = 0, \dots, M$  do
10     $x_j^{n+1} = jh_{n+1}$ ;
11  end
12  Interpolation:

```

$$u_j^{n+1} = \tilde{u}_j^{n+1} + \frac{x_j^{n+1} - x_j^n}{h_n} (\tilde{u}_{j+1}^{n+1} - \tilde{u}_j^{n+1}); \quad (32)$$

```

    n=n+1;
13 end

```

The inequality (34) is guaranteed if

$$M(e^{(r-q)k} - 1) < 1, \quad (35)$$

which occurs if k satisfies

$$k < \frac{\ln(1 + h_0)}{r - q}. \quad (36)$$

In the case $r < q$ domain is decreasing and $k < \frac{\ln(1+h_0)}{q-r}$ is a sufficient condition for $x_j^n \in [x_j^{n+1}, x_{j+1}^{n+1})$. In the case when $r = q$ the moving boundary is fixed with $x_M^n = 1$ for all n and the interpolation is not necessary.

3.2 An Alternating Direction Explicit (ADE)

An Alternating Direction Explicit method combines the advantages of simplicity of an explicit method and the unconditional stability of implicit scheme for the linear case (see [26]). The numerical solution is calculated as the average of two solutions using explicit scheme known as the right direction solution $\{R_j^n\}$ and the left direction solution $\{L_j^n\}$. The algorithm of the ADE method for the moving domain problem (14)-(20) is given as follows.

Algorithm 3: Alternating direction explicit method.

Data: Initial values: $L_j^0 = R_j^0 = u_j^0$;
Result: Solution at $\tau = T$;

```

1 n=0;
2 while n < N do
3    $S_f^{n+1} = \varrho(u^n, S_f^n)$ ;
4   Set the boundary conditions:  $\tilde{R}_0^{n+1} = \tilde{L}_0^{n+1} = 0$ ;
5   Compute right direction solution:
6   for j = 1, ..., M - 1 do
7     
$$\frac{\tilde{R}_j^{n+1} - R_j^n}{k} = \frac{\sigma_{j,n}^2 (x_j^n)^2}{2} \frac{\tilde{R}_{j-1}^{n+1} - \tilde{R}_j^{n+1} - R_j^n + R_{j+1}^n}{h_n^2} + \frac{S_f^{n+1} - S_f^n}{k S_f^n} x_j^n \frac{R_{j+1}^n - \tilde{R}_{j-1}^{n+1}}{2h_n}. \quad (37)$$

8   end
9   Set the boundary conditions:  $R_M^{n+1} = L_M^{n+1} = e^{r\tau^{n+1}} (S_f^{n+1} - 1)$ ;
    $\tilde{R}_{M-1}^{n+1} = \tilde{L}_{M-1}^{n+1} = R_M^{n+1} - S_f^{n+1} e^{q\tau^{n+1}} (e^{(r-q)\tau^{n+1}} - x_{M-1}^n)$ ;
10  Compute left direction solution:
11  for j = M - 1, ..., 1 do
12    
$$\frac{\tilde{L}_j^{n+1} - L_j^n}{k} = \frac{\sigma_{j,n}^2 (x_j^n)^2}{2} \frac{L_{j-1}^n - L_j^n - \tilde{L}_j^{n+1} + \tilde{L}_{j+1}^{n+1}}{h_n^2} + \frac{S_f^{n+1} - S_f^n}{k S_f^n} x_j^n \frac{\tilde{L}_{j+1}^{n+1} - L_{j-1}^n}{2h_n}. \quad (38)$$

13  end
14  Construct new uniform grid:
15  for j = 0, ..., M do
16     $x_j^{n+1} = j h_{n+1}$ ;
17  end
18  Interpolate the data  $R_j^{n+1}$  and  $L_j^{n+1}$ ,  $j = 1, \dots, M - 1$  onto the new grid;
19 end
20 for j = 0, ..., M do
21    $u_j^{n+1} = \frac{R_j^{n+1} + L_j^{n+1}}{2}$ ;
22 end
23 n=n+1;
```

3.3 Implicit numerical methods

Both implicit and explicit numerical methods have advantages and disadvantages (see introduction of [8]). In previous subsections two explicit numerical methods are discussed and in this subsection implicit schemes are discussed. The discretisation of the nonlinear equation described in (14) leads to a system of nonlinear equations. The most popular and widely used method for solving nonlinear systems is the so-called Newton's method. This method is iterative and requires to calculate Jacobian of the nonlinear system every iteration which is time consuming. There exists various modifications of the method [22].

A fully implicit scheme for the equation (14) using the same notation as in (27) takes the form

$$\frac{\tilde{u}_j^{n+1} - u_j^n}{k} = \frac{\sigma_{j,n+1}^2}{2} (x_j^n)^2 \frac{\tilde{u}_{j-1}^{n+1} - 2\tilde{u}_j^{n+1} + \tilde{u}_{j+1}^{n+1}}{h_n^2} + \frac{S_f^{n+1} - S_f^n}{kS_f^{n+1}} x_j^n \frac{\tilde{u}_{j+1}^{n+1} - \tilde{u}_{j-1}^{n+1}}{2h_n} \quad (39)$$

for $j = 1, \dots, M-2$. Since the left boundary is fixed at $x_0^n = 0$, and the right boundary is given by $x_M^n = e^{(r-q)\tau^n}$, the last three points are non-equidistant. Let $\tilde{h}_n = x_M^{n+1} - x_{M-1}^n$. Taylor's series expansion is used to obtain a discretization of the second derivative on the non-uniform grid:

$$V_{xx}(x_{M-1}^n, \tau^{n+1}) \approx 2 \left(\frac{\tilde{u}_{M-2}^{n+1}}{h_n(h_n + \tilde{h}_n)} + \frac{u_M^{n+1}}{\tilde{h}_n(h_n + \tilde{h}_n)} - \frac{\tilde{u}_{M-1}^{n+1}}{h_n \tilde{h}_n} \right). \quad (40)$$

Using (40) and a central difference for the first derivative, the implicit scheme (39) for $j = M-1$ takes the following form:

$$\begin{aligned} \frac{\tilde{u}_{M-1}^{n+1} - u_{M-1}^n}{k} = & \sigma^2(M-1, n+1)(x_{M-1}^n)^2 \left(\frac{\tilde{u}_{M-2}^{n+1}}{h_n(h_n + \tilde{h}_n)} + \frac{u_M^{n+1}}{\tilde{h}_n(h_n + \tilde{h}_n)} - \frac{\tilde{u}_{M-1}^{n+1}}{h_n \tilde{h}_n} \right) \\ & + \frac{S_f^{n+1} - S_f^n}{kS_f^{n+1}} x_{M-1}^n \frac{u_M^{n+1} - \tilde{u}_{M-2}^{n+1}}{2h_n}. \end{aligned} \quad (41)$$

Boundary conditions are discretised as follows

$$u_M^{n+1} = e^{r\tau^{n+1}} (S_f^{n+1} - 1), \quad \frac{u_M^{n+1} - \tilde{u}_{M-1}^{n+1}}{\tilde{h}_n} = e^{q\tau^{n+1}} S_f^{n+1}. \quad (42)$$

Define the following coefficients

$$a_{-1}(j, n + 1) = -\frac{k}{h_n^2}(x_j^n)^2 \frac{\sigma_{j,n+1}^2}{2} + \left(1 - \frac{S_f^n}{S_f^{n+1}}\right) \frac{x_j^n}{2h_n}, \quad (43)$$

$$a_0(j, n + 1) = 1 + \frac{k}{h_n^2}(x_j^n)^2 \sigma_{j,n+1}^2, \quad (44)$$

$$a_{+1}(j, n + 1) = -\frac{k}{h_n^2}(x_j^n)^2 \frac{\sigma_{j,n+1}^2}{2} - \left(1 - \frac{S_f^n}{S_f^{n+1}}\right) \frac{x_j^n}{2h_n}, \quad (45)$$

$$\tilde{a}_{-1}(n + 1) = -\frac{k}{h_n(h_n + \tilde{h}_n)}(x_{M-1}^n)^2 \sigma^2(M - 1, n + 1) \quad (46)$$

$$+ \left(1 - \frac{S_f^n}{S_f^{n+1}}\right) \frac{x_{M-1}^n}{h_n + \tilde{h}}, \quad (47)$$

$$\tilde{a}_0(n + 1) = 1 + \frac{k}{h_n \tilde{h}_n}(x_{M-1}^n)^2 \sigma^2(M - 1, n + 1), \quad (48)$$

$$\tilde{a}_{-1}(n + 1) = -\frac{k}{h_n(h_n + \tilde{h}_n)}(x_{M-1}^n)^2 \sigma^2(M - 1, n + 1) \quad (49)$$

$$- \left(1 - \frac{S_f^n}{S_f^{n+1}}\right) \frac{x_{M-1}^n}{h_n + \tilde{h}}. \quad (50)$$

The fully implicit scheme can be expressed in the matrix form

$$A_{n+1}U_{n+1} = B_n, \quad (51)$$

where

$$A_{n+1} = \begin{pmatrix} a_0(1, n + 1) & a_{+1}(1, n + 1) & 0 & 0 & \dots & 0 \\ a_{-1}(2, n + 1) & a_0(2, n + 1) & a_{+1}(2, n + 1) & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \tilde{a}_{-1}(n + 1) & \tilde{a}_0(n + 1) & \tilde{a}_{+1}(n + 1) & 0 \\ 0 & \dots & 0 & -1 & 1 & -e^{q\tau^{n+1}}\tilde{h}_n \\ 0 & \dots & 0 & 0 & -1 & e^{r\tau^{n+1}} \end{pmatrix} \quad (52)$$

$$U_{n+1} = \begin{pmatrix} \tilde{u}_1^{n+1} \\ \tilde{u}_1^{n+1} \\ \vdots \\ \tilde{u}_{M-1}^{n+1} \\ u_M^{n+1} \\ S_f^{n+1} \end{pmatrix}, \quad B_n = \begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{M-1}^n \\ 0 \\ e^{r\tau^{n+1}} \end{pmatrix}. \quad (53)$$

Newton's method is applied to solve the nonlinear system (51). At each time level an initial guess is required for the iterative process in the Newton's method and may be chosen as the **approximate** solution at the previous time level. The stopping criterion is chosen to be the norm of the increment becomes smaller than the tolerance ϵ .

Algorithm 4: Newton's method.

Data: Initial conditions U_0 ;
Result: U_N ;

```

1 n=0;
2 while n < N do
3   Newton loop: Initial approximation  $U_{n+1} := U_n$ ;  $\|e\| :=$  big value;
4   while  $\|e\| > \epsilon$  do
5      $G := A_{n+1}U_{n+1} - B_n$ ;
6     Compute Jacobian:  $J(G)$ ;
7      $e = \{J(G)\}^{-1}G$ ;
8      $U_{n+1} = U_{n+1} - e$ ;
9   end
10  Construct new uniform mesh at the new time level;
11  Calculate values in the new points using linear (or quadratic) interpolation;
12  n=n+1;
13 end

```

There are many modifications of Newton's method mainly to improve the efficiency and robustness of the method. One type of modification aims to avoid the computations of Jacobian every iteration in order to reduce the total computational time. These methods are collectively known as Newton-like methods. For instance, the main idea of Broyden's method is to calculate an approximate Jacobian iteratively using simple matrix vector multiplications as given below.

$$J_k = J_{k-1} + \frac{\Delta G_k - J_{k-1} \Delta u_k}{\|\Delta u_k\|^2} \Delta u_k^T, \quad (54)$$

where k is the number of current Newton's iteration, $\Delta G_k = G_k - G_{k-1}$, $\Delta u_k = u_k - u_{k-1}$. The initial value J_0 has to be calculated by a standard procedure to avoid instability. Since $\Delta G_k - J_{k-1} \Delta u_k \approx G_{k+1}$, (54) can be presented in the following form

$$J_k = J_{k-1} + \frac{G_{k+1}}{\|\Delta u_k\|^2} \Delta u_k^T. \quad (55)$$

Unfortunately, if the Jacobian has a given sparsity structure, as it occurs in the present study, Broyden's approximation breaks the structure and introduces non-zero values to those zero components. Schubert's method [28] is widely used for sparse matrices because it preserves the sparsity of the Jacobian. Although it has good properties, it is sensitive to the problem under consideration and size of the matrix. Indeed, our problem is not well conditioned for the Schubert's methods as it is shown in Table 5. Therefore, a modification of the method is proposed in order to overcome these computational difficulties. Instead of taking the squared norm in denominator power one was used in all the tests. This modification is denoted as "Schubert-1" method. Numerical tests show that it ensures the convergence of Schubert's algorithm.

In order to overcome the drawback of Broyden's method all of the matrix elements outside the tri-diagonal band were "frozen" at zeros in the numerical tests. This modification preserves the structure of the matrix in the same way as Schubert's method does.

The proposed modification is known as "frozen-Broyden":

$$J_k(i, j) = \begin{cases} J_k(i, j), & j - 1 \leq i \leq j + 1, \\ 0, & \text{otherwise.} \end{cases}$$

Spectral analysis confirms quality of the proposed methods by the numerical examples provided in Section 4.

4. Numerical examples

This section is devoted to several numerical tests and a comparison of the explicit and implicit methods as described above. Convergence rate and computational costs for the numerical solution of Barles and Soner's model for American options are presented.

Example 1. An American call option pricing problem in the transformed form (14)-(20) with the parameters:

$$r = 0.1, q = 0.05, T = 1, \sigma_0 = 0.2, E = 10, \quad (56)$$

was tested.

Barles and Soner's model with $a = 0.05$ was chosen in the test. In this example the numerical convergence rate in terms of root mean square error (RMSE) (see [27], p. 385) of the proposed methods are presented. The RMSE may be computed by the following formula:

$$RMSE_h = \sqrt{\frac{\sum (u^*(x_i, T) - u_h(x_i, T))^2}{M}}, \quad (57)$$

where $u^*(x_i, T)$ is a "true value" of function $V(x_i, T)$ and $u_h(x_i, T)$ is calculated value in the point (x_i, τ^N) . Here the "true value" is understood as the numerical solution on a refined grid with step sizes $h = 5 \cdot 10^{-3}$ and $k = 10^{-5}$.

The spatial convergence rate of the approximate solutions are calculated for different step sizes h at a fixed time step k by using the following formula

$$\gamma(h_1, h_2) = \frac{\ln(RMSE_{h_1}) - \ln(RMSE_{h_2})}{\ln(h_1) - \ln(h_2)}. \quad (58)$$

In Table 2 the results and comparison are presented. The time step is fixed at $k = 0.0001$ to guarantee stability of all numerical solutions. For implicit method the tolerance was chosen as $\epsilon = 10^{-4}$.

From Table 2, taking the mean value of all combinations of h_1 and h_2 , one obtains

$$\gamma_{expl} = 1.512, \quad \gamma_{impl} = 1.191, \quad \gamma_{ADE} = 1.734, \quad \gamma_{NL} = 1.713. \quad (59)$$

An analogous formula to (58) can be used for convergence rate in time. RMSE and computational time for a fixed spatial step are presented in Table 3.

Using data from Table 3, the convergence rate in time can be calculated as follows

$$\gamma_{expl} = 0.627 \quad \gamma_{impl} = 0.733, \quad \gamma_{ADE} = 1.789, \quad \gamma_{NL} = 0.691. \quad (60)$$

Table 2. RMSE with respect to CPU-time for different h_0 and fixed $k = 0.0001$.

h_0	0.08	0.04	0.02	0.01
Explicit method				
RMSE	0.04984	0.02629	0.01232	0.00464
CPU-time, s	15.810	27.566	51.476	99.434
ADE method				
RMSE	0.16816	0.08172	0.02099	0.00620
CPU-time, s	15.129	27.776	53.865	104.247
Implicit method				
RMSE	0.04984	0.02355	0.00958	0.00445
CPU-time, s	34.099	60.030	112.728	257.880
Newton-like method				
RMSE	0.11376	0.06026	0.01389	0.00471
CPU-time, s	33.869	58.141	107.561	315.505

Table 3. RMSE with respect to CPU-time for different k and fixed $h_0 = 0.01$.

k	0.001	0.0005	0.0002	0.0001
Explicit method				
RMSE	0.01713	0.01373	0.00675	0.00464
CPU-time, s	11.470	21.361	50.598	99.434
ADE method				
RMSE	0.38763	0.08152	0.01839	0.00620
CPU-time, s	10.829	21.199	52.684	104.247
Implicit method				
RMSE	0.02122	0.01528	0.00639	0.00445
CPU-time, s	39.318	52.327	129.045	257.880
Newton-like method				
RMSE	0.02127	0.01524	0.00815	0.00471
CPU-time, s	38.650	52.277	128.371	255.347

Note that the main part of the computational time is pertained for the calculation of $\Psi(A)$. For the implicit methods it has to be calculated on each iteration of Newton's method. Thus, their computational costs may be noticeably reduced by choosing another model.

Next example presents a study of the free boundary for both RAPM and Barles and Soner's models with various values of the respective transaction cost parameters R and a .

Example 2. Let the problem (1)-(7) under RAPM model with the parameters (56), fixed transaction cost $C_{tr} = 0.01$ and various risk premium measure $R = 5, 15, 40, 70, 100$ to be considered. The coefficient $\mu = 3 \left(\frac{C_{tr}^2 R}{2\pi} \right)^{1/3}$, according to [18]. Figure 2 shows the variation of the normalised free boundary $S_f(\tau)$ depending on the parameter R .

In Figure 3 there are numerical results for Barles and Soner's model for various a . The difference between values for $a = 0$ and $a = 0.01$ is inappreciable.

In next example the validity of the proposed explicit scheme (25) is discussed. Explicit scheme uses information from the previous time level to compute a solution at the current moment. For nonlinear equations with coefficients depending on the solution one has two alternatives: if we take values from the current time level to compute the coefficients the scheme would not be explicit and we have to use any iterative solver for this problem. It increases computational time. Another alternative is to take values from the previous

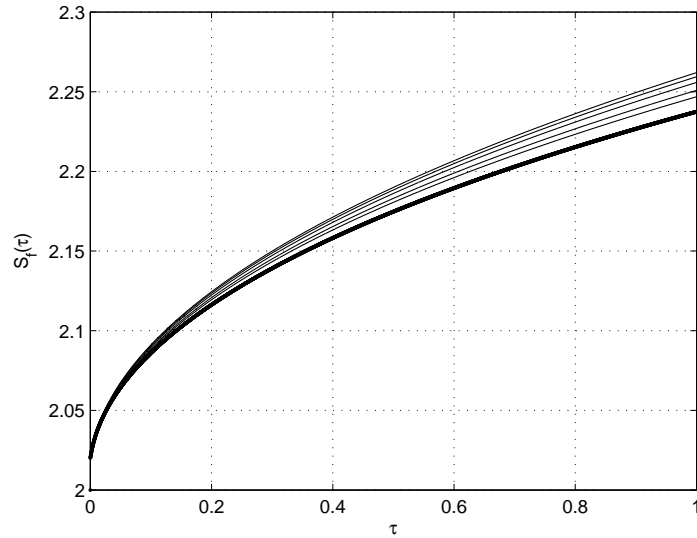


Figure 2. A comparison of the free boundary $S_f(\tau)$ for RAPM model for various risk premium measures $R = 5, 15, 40, 70, 100$ with the corresponding free boundary for $R = 0$ (bold line).

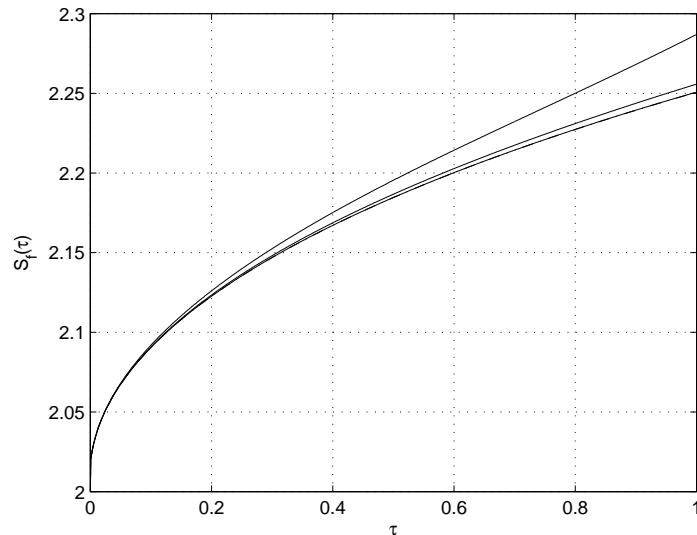


Figure 3. A comparison of the free boundary $S_f(\tau)$ for Barles and Soner's model for $a = 0, 0.01, 0.07, 0.13$.

time level as we used, and the coefficients may be inaccurate.

Example 3. The transformed American call option pricing problem under Barles and Soner's model (14)-(20) with parameters (56) and $a = 0.05$ is considered. Figure 4 demonstrates the difference between the solutions obtained by both alternatives for fixed h_0 and various k . One can see that the difference presents orders no bigger than $O(k)$ that is the order of approximation of the explicit forward in time scheme (25).

Moreover, the series of tests was provided to insure this statement. For fixed k the maximum value of the difference between the solutions is calculated. The results are collected in Table 4.

In order to study stability of the proposed explicit method we compare solutions for

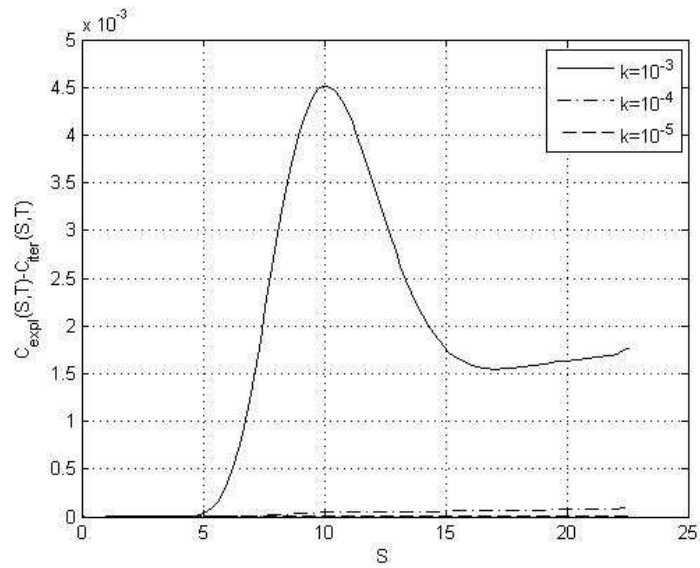


Figure 4. Difference between solutions by explicit method and iterative explicit method with $h_0 = 10^{-2}$ and various k .

Table 4. The maximum distance between the solution of the problem (14) by explicit and iterative explicit method.

h_0	0.01	0.02	0.04
$k = 10^{-3}$	$1.3548 \cdot 10^{-4}$	$7.5463 \cdot 10^{-4}$	$9.5728 \cdot 10^{-5}$
$k = 10^{-4}$	$9.9036 \cdot 10^{-6}$	$1.2605 \cdot 10^{-5}$	$1.0068 \cdot 10^{-5}$
$k = 10^{-5}$	$1.0936 \cdot 10^{-6}$	$1.2995 \cdot 10^{-6}$	$1.6136 \cdot 10^{-6}$

the problem with the parameters (56) for fixed $h = 10^{-2}$ and various $k = 10^{-4}$ and $k = 2.6 \cdot 10^{-3}$ (Figures 5 and 6 correspondingly). As one can see, the numerical solution as shown in Figure 6 is unstable.

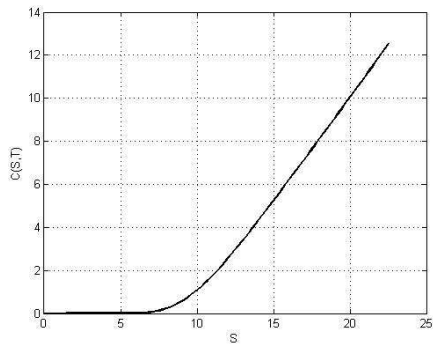


Figure 5. Numerical solution with $k = 10^{-4}$.

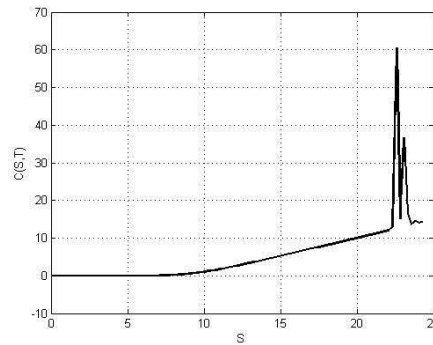


Figure 6. Numerical solution with $k = 2.6 \cdot 10^{-3}$

Next example is used to examine the validity of the proposed modifications in the class of Newton-like methods.

Example 4. Well known Newton-like methods developed by Broyden and Schubert as well as proposed modifications are used to approximate the Jacobian of the problem of Example 1 with $h = 0.01$, $k = 0.001$ and $k = 0.0001$.

In order to demonstrate the viability of the modifications to well known Broyden's and Schubert's methods described in Section 3.3, the spectral radius is used. Let a approximation of Jacobian by any method be denoted as J_{approx} . Since matrices supposed to be approximation of the original Jacobian J , then matrix $J_{approx}^{-1}J$ should be close to identity matrix. Spectral radius of matrix is used to check this fact.

In Table 5 maximum, minimum and mean value of spectral radius of matrices $J_{approx}^{-1}J$ are presented. Further tests, **performed but not presented in the paper**, show that Broyden's method **fails** for smaller step sizes **and solution is unstable**.

Table 5. Spectral radius of matrix $J_{approx}^{-1}J$, where J_{approx} is calculated by various methods.

	Min	Max	Mean
$h = 0.01, k = 0.001$			
frozen-Broyden	1.00239414	1.01232560	1.00648229
Schubert-1	1.00000712	1.12629296	1.00054588
Broyden	1.00015443	1.00103625	1.00066090
Schubert		fail	
$h = 0.01, k = 0.0001$			
frozen-Broyden	1.00044224	1.00372329	1.00095909
Schubert-1	1.00020364	1.00333878	1.00059102
Broyden	1.00024579	1.00377139	1.00075563
Schubert		fail	

5. Conclusion

American option pricing with transaction cost is well known to be a free boundary problem. A new suitable transformation is proposed leading to a problem with known moving domain that is solved numerically. Explicit Euler scheme, ADE method, fully implicit method based on Newton and Newton-like methods are described and compared with their convergence rates and computational costs analysed.

Modifications of Broyden's and Schubert's methods to improve qualitative properties are proposed. Thus, the so-called "frozen-Broyden" preserves tridiagonal structure of approximated Jacobian. New "Schubert-1" technique produces convergent results for a wide range of step sizes. All Newton-like methods are compared in the sense of spectral radius. The results confirm viability of the proposed modifications.

For explicit methods coefficients from the previous known time level are used. It does not influence on the accuracy of the solution and does not require additional computations of iterative procedure. This fact is confirmed by numerical examples.

The techniques described have been applied to the RAPM and Barles and Soner's models. Its versatility allows to be used for any nonlinear model with nonconstant volatility.

Acknowledgements

This work has been partially supported by the European Union in the FP7- PEOPLE-2012-ITN program under Grant Agreement Number 304617 (FP7 Marie Curie Action, Project Multi-ITN STRIKE-Novel Methods in Computational Finance) and the Ministerio de Economía y Competitividad Spanish grant MTM2013-41765-P.

References

- [1] J. Ankudinova, M. Ehrhardt, Fixed domain transformations and split-step Finite Difference schemes for nonlinear Black-Scholes equations for American Options. *Nonlinear Models in Mathematical Finance: New Research Trends in Option Pricing*. Nova Science Publishers, Inc. New York (2008) 243–283.
- [2] J. Ankudinova, M. Ehrhardt, On the numerical solution of nonlinear Black-Scholes equations, *J. Computers and Mathematics with Applications* 56 (2008) 799–812.
- [3] M. Avellaneda A. Lévy and A. Parás, Pricing and hedging derivative securities in markets with uncertain volatilities, *J. Applied Mathematical Finance* 2 (1995) 73–88.
- [4] G. Barles, H. M. Soner, Option pricing with transaction costs and a nonlinear Black-Scholes equation, *J. Finance Stochastics* 2 (1998) 369–397.
- [5] F. Black and M.S. Scholes, The pricing of options and corporate liabilities, *J. Political Economy* 81 (1973) 637–654.
- [6] P. Boyle and T. Vorst, Option replication in discrete time with transaction costs, *J. Finance* 47 (1973) 271–293.
- [7] R. Company, V.N. Egorova, L. Jódar, Constructing positive reliable numerical solution for American call options: a new front-fixing approach, *J. Computational and Applied Mathematics*, 291 (2016) 422–431.
- [8] R. Company, L. Jódar, E. Ponsoda, C. Ballester, Numerical analysis and simulation of option pricing problems modeling illiquid markets, *Computers & Mathematics with Applications*, 59-8 (2010) 2964–2975.
- [9] R. Company, E. Navarro, J.R. Pintos and E. Ponsoda, Numerical solution of linear and nonlinear Black-Scholes option pricing equations, *J. Computers & Mathematics with Applications*, 56 (2008) 813–821.
- [10] R. Company, L. Jódar and J.R. Pintos, Consistent stable difference schemes for nonlinear Black-Scholes equations modelling option pricing with transaction costs, *ESAIM: Mathematical Modelling and Numerical Analysis*, 43 (2009) 1045–1061
- [11] J. Crank, *Free and Moving Boundary problems*. Oxford University Press (1984).
- [12] E. Dremkova, M. Ehrhardt, A high-order compact method for nonlinear Black-Scholes option pricing of American options, *International Journal of Computer Mathematics*, 88 (2011) 2782–2797.
- [13] D.J. Duffy, *Unconditionally Stable and Second-Order Accurate Explicit Finite Difference Schemes Using Domain Transformation: Part I One-Factor Equity Problems*, (2009), SSRN.
- [14] B. Düring, M. Fournier and A. Jüngel, Convergence of a high order compact finite difference scheme for a nonlinear BlackScholes equation, *ESAIM: Mathematical Modelling and Numerical Analysis*, 38 (2004) 359–369.
- [15] P. Heider, Numerical methods for nonlinear Black-Scholes equations, *J. Applied Mathematical Finance*, 17 (2010) 59–81
- [16] T. Hoggard, A.E. Whalley and P. Wilmott, Hedging option portfolios in the presence of transaction costs. *J. Advances in Futures and Options Research*, 7 (1994) 217–35.
- [17] J. Hull, A. White, Valuing derivative securities using the explicit finite difference method. *J. Financial and Quantitative Analysis*, 25 (1990) 87–100.
- [18] M. Jandačka, D. Ševčovič, On the risk adjusted pricing methodology based valuation of vanilla options and explanation of the volatility smile, *J. Applied Mathematics*, 3(2005) 235–258.
- [19] M. Kratka, No mystery behind the smile, *J. Risk*, 9 (1998) 67–71.
- [20] D. Knuth, *The Art of Computer Programming 3* (3rd ed.), Addison-Wesley, 1997.
- [21] Y.-K. Kwok, *Mathematical Models of Financial Derivatives*, Springer, Berlin, 2nd edition (2008).
- [22] C.-H. Lai, An application of quasi-Newton methods for the numerical solution of interface problems, *J. Advances in Engineering Software*, 28 (1997) 333–339.
- [23] H.G. Landau, Heat conduction in a melting solid, *J. Quarterly Applied Mathematics*, 8 (1950) 81–95.
- [24] M. Leland, Option pricing and replication with transaction costs, *J. Finance*, 5 (1985) 1283–1301.
- [25] D.C. Lesmana, S. Wang, An upwind finite difference method for a nonlinear Black-Scholes equation governing European option valuation, *Applied Mathematics and Computation*, 219 (2013) 8818–8828.
- [26] S. Leung, S. Osher, An alternating direction explicit (ADE) scheme for time-dependent evolution equations, *Technical report UCLA*, 2005.
- [27] G.R. Liu, *Mesh free methods: moving beyond the finite element method*, CRC Press, 2002.
- [28] E. Marwil, Convergence results for Schubert’s method for solving sparse nonlinear equations, *SIAM J. Numerical Analysis*, 16-4 (1979) 588–604.

- [29] B.F. Nielsen, O. Skavhaug, A. Tvelto, Penalty and front-fixing methods for the numerical solution of American option problems, *J. Computational Finance*, 5 (2002).
- [30] G. Pealat, D.J. Duffy, The alternating direction explicit (ADE) method for one-factor problems, *Wilmott Magazine*, (2011) 54–60.
- [31] D. M. Pooley, P. A. Forsyth, K. R. Vetzal, Numerical convergence properties of option pricing PDEs with uncertain volatility, *J. Numerical Analysis*, 23 (2003).
- [32] D. Ševčovič, Analysis of the free boundary for the pricing of an American call option, *European Journal of Applied Mathematics*, 12-1 (2001) 25–37.
- [33] D. Ševčovič, An iterative algorithm for evaluating approximations to the optimal exercise boundary for a nonlinear Black-Scholes equation, *J. Canadian Applied Mathematics Quarterly*, 15-1 (2007) 77–97.
- [34] D. Ševčovič, Transformation methods for evaluating approximations to the optimal exercise boundary for linear and nonlinear Black-Scholes equations. *Nonlinear Models in Mathematical Finance: New Research Trends in Option Pricing*. Nova Science Publishers, Inc. New York (2008) 173–218.
- [35] P. Wilmott, S. Howison, J. Dewynne, *The mathematics of financial derivatives*. Cambridge University Press, Cambridge, UK, 1995.
- [36] L. Wu, Y.-K. Kwok, A front-fixing method for the valuation of American option, *J. Financial Engineering*, 6-2 (1997) 83-97.

Author's Accepted Manuscript