

Penalty and front-fixing methods for the numerical solution of American option problems.

Bjørn Fredrik Nielsen, Ola Skavhaug and Aslak Tveito

January 31, 2000

Abstract

In this paper we introduce two methods for the efficient and accurate numerical solution of Black-Scholes models of American options; A penalty method and a front-fixing scheme.

In the penalty approach the free and moving boundary is removed by adding a small, and continuous penalty term to the Black-Scholes equation. Then the problem can be solved on a fixed domain and thus removing the difficulties associated with a moving boundary. To gain insight in the accuracy of the method, we apply it to similar situations where the approximate solutions can be compared with analytical solutions. For explicit, semi-implicit and fully-implicit numerical schemes, we prove that the numerical option values generated by the penalty method mimics the basic properties of the analytical solution of the American option problem.

In the front-fixing method we apply a change of variables to transform the American put problem into a nonlinear parabolic differential equation posed on a fixed domain. We propose both an implicit and an explicit scheme for solving this latter equation.

Finally, the performance of the schemes are illustrated through a series of numerical experiments.

1 Introduction

Analytical solutions of Black-Scholes models of American option problems are seldom available, and hence such derivatives must be priced by numerical techniques. The problem of solving the American option problem numerically has during the last decade been subject for intensive research, cf. e.g. [1, 2, 4].

Elementary introductions to this topic can be found in e.g. [10, 13, 14, 15]. In this paper we introduce two schemes for solving the free and moving boundary value problem arising in such models; A front-fixing scheme and a penalty method. In both cases we derive problems, in terms of nonlinear parabolic differential equations, posed on fixed domains. Thus, significantly simplifying the numerical solution of the American put problem.

The penalty method for solving option problems was introduced by Zvan, Forsyth and Vetzal in [19]. Our objective is to derive a refinement of their approach which is easy to generalize to any American type of option. We do this by adding a term to the partial differential equation assuring that the solution will stay in the proper state space but also altering the exact solution as little as possible. For explicit, semi-implicit a fully implicit numerical schemes, we derive conditions that assures that the approximate option values satisfies the basic properties of the analytical solution of the problem. In addition, the performance of the schemes are illustrated through a series of numerical experiments. In particular, for a simple model problem, the examples indicate that the approximations generated by the penalty method converge towards the correct solution as the penalty term tend towards zero.

The front-fixing method has been applied successfully to a wide range of problems arising in physics, cf. [5] and references therein. The basic idea is to remove the moving boundary by a transformation of the involved variables. In this paper we show how this technique can be applied to the American put problem. Furthermore, we present an implicit and an explicit scheme for solving the resulting nonlinear parabolic equation. It should be mentioned that a similar approach has been studied by Zhu, Ren and Xu in [17]. They apply a singularity-separating method to derive an equation for the difference between the value of an American and an European option. Thereafter, this latter problem is mapped onto a fixed domain. In [18, 16] this approach is generalized to more advanced pricing problems for various derivatives. In contrast to their work we focus on the transformation of the moving boundary onto a stationary domain. Furthermore, we use the computational results obtained by the front-fixing method as a reference solution for studying the convergence properties of our penalty schemes.

In addition to penalty, singularity-separating and front-fixing methods for solving option problems several schemes have been proposed. Among these are the Brennan and Schwartz algorithm [3, 9], the projected SOR scheme [15], the binomial method [8] and Monte Carlo simulation techniques [7, 13, 14].

The outline of the paper is as follows. The next section contains the Black-Scholes model for American put problems. In Section 3 we define the Front-Fixing Method and the associated explicit and implicit numerical

schemes. The Penalty Method is introduced in Section 4 for two simple model problems. More precisely, in this section we present such methods, along with their convergence properties and numerical experiments, for an ordinary differential equation and a two point boundary value problem. Finally, Section 5 contains the derivation of the penalty method, and the resulting numerical schemes, for solving American put problems. This section also contains several numerical experiments illustrating the performance of our algorithms.

2 The mathematical model

Suppose that at time t the price of an asset A is S . The American early exercise constraint leads to the following mathematical model for the value $P = P(S, t)$ of an American put option to sell A ;

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + rS \frac{\partial P}{\partial S} - rP = 0 \quad \text{for } S > \bar{S}(t) \text{ and } 0 \leq t < T, \quad (1)$$

$$P(S, T) = \max(E - S, 0) \quad \text{for } S \geq 0, \quad (2)$$

$$\frac{\partial P}{\partial S}(\bar{S}(t), t) = -1, \quad (3)$$

$$P(\bar{S}(t), t) = E - \bar{S}(t), \quad (4)$$

$$\lim_{S \rightarrow \infty} P(S, t) = 0, \quad (5)$$

$$\bar{S}(T) = E, \quad (6)$$

$$P(S, t) = E - S \quad \text{for } 0 \leq S < \bar{S}(t), \quad (7)$$

where $\bar{S}(t)$ represents the free (and moving) boundary, see e.g. [7], [10] or [15]. Here, σ , r and E are given parameters representing the volatility of the underlying asset, the interest rate and the exercise price of the option, respectively. Note that, since early exercise is permitted, the value P of the option must satisfy

$$P(S, t) \geq \max(E - S, 0) \quad \text{for all } S \geq 0 \text{ and } 0 \leq t \leq T, \quad (8)$$

cf. [15]. Mathematical models of this kind, involving a moving boundary, are frequently referred to as moving boundary problems, cf. [5] and Figure 1.

3 A front-fixing method

The basic idea behind the front-fixing method is to remove the moving boundary in the American option problem by a change of variables. It turns out

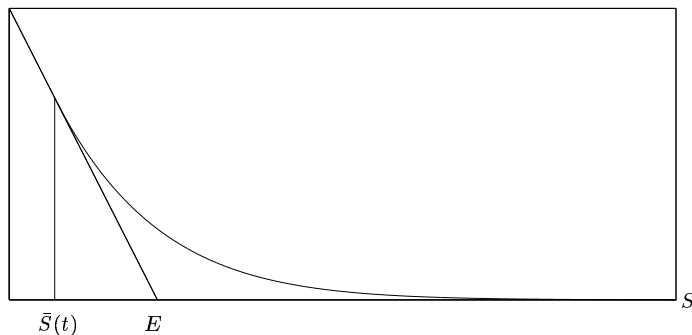


Figure 1: The figure shows a typical solution of the American put problem (1)-(7) at time $t < T$. Here, E represents the exercise price and $\bar{S}(t)$ is the moving boundary.

that this approach leads to a nonlinear problem posed on a fixed domain. In this latter formulation of the problem the position of the boundary is given but some of the boundary conditions remain unknown and must consequently be computed.

We want to solve the problem (1)-(7) using a front-fixing method, i.e. by a transformation of the involved variables. To this end, define

$$x = S/\bar{S}(t) \quad \text{or} \quad S = x\bar{S}(t), \quad (9)$$

and

$$p(x, t) = P(S, t) = P(x\bar{S}(t), t). \quad (10)$$

Notice that we have $x \in [1, \infty)$ for $S \in [\bar{S}(t), \infty)$. Our goal is to derive, from (1)-(6), a set of equations for $p(x, t)$ for $x \geq 1$ and $0 \leq t \leq T$. That is, to obtain a problem posed on a fixed domain.

The final condition (2) for p takes the form

$$\begin{aligned} p(x, T) &= P(S, T) = P(x\bar{S}(T), T) = \max(E - x\bar{S}(T), 0) \\ &= \max(E - xE, 0) = E \max(1 - x, 0) = 0 \quad \text{for } x \geq 1, \end{aligned} \quad (11)$$

where we have used (6). Next, we derive the boundary conditions. Differentiating (10) with respect to x gives

$$\frac{\partial p}{\partial x} = \frac{\partial P}{\partial S} \frac{\partial S}{\partial x} = \bar{S}(t) \frac{\partial P}{\partial S}, \quad (12)$$

and thus (3) implies that

$$\frac{\partial p}{\partial x}(1, t) = \bar{S}(t) \frac{\partial P}{\partial S}(\bar{S}(t), t) = -\bar{S}(t). \quad (13)$$

From (4), (5) and (9) we find that

$$p(1, t) = P(\bar{S}(t), t) = E - \bar{S}(t), \quad (14)$$

and

$$\lim_{x \rightarrow \infty} p(x, t) = \lim_{x \rightarrow \infty} P(x\bar{S}(t), t) = 0. \quad (15)$$

A partial differential equation for $p(x, t)$ is derived from (1) which governs $P(S, t)$. In order to do this, we need to express $\partial P/\partial t$, $\partial P/\partial S$ and $\partial^2 P/\partial S^2$ in terms of p and its derivatives. From (12), we have

$$\frac{\partial P}{\partial S} = \frac{1}{\bar{S}(t)} \frac{\partial p}{\partial x} \quad (16)$$

Differentiating (12) with respect to x gives

$$\frac{\partial^2 p}{\partial x^2} = \bar{S}(t) \frac{\partial^2 P}{\partial S^2} \frac{\partial S}{\partial x} = \bar{S}^2(t) \frac{\partial^2 P}{\partial S^2},$$

or

$$\frac{\partial^2 P}{\partial S^2} = \frac{1}{\bar{S}^2(t)} \frac{\partial^2 p}{\partial x^2}. \quad (17)$$

By differentiating (10) with respect to t , we get

$$\frac{\partial p}{\partial t} = \frac{\partial P}{\partial t} + \frac{\partial P}{\partial S} \frac{\partial S}{\partial t} = \frac{\partial P}{\partial t} + \frac{\partial P}{\partial S} x \bar{S}'(t),$$

so (16) yields

$$\frac{\partial P}{\partial t} = \frac{\partial p}{\partial t} - x \frac{\bar{S}'(t)}{\bar{S}(t)} \frac{\partial p}{\partial x}. \quad (18)$$

Hence, it follows from (1),(9),(16),(17) and (18) that $p(x, t)$ must satisfy

$$\frac{\partial p}{\partial t} - x \frac{\bar{S}'(t)}{\bar{S}(t)} \frac{\partial p}{\partial x} + \frac{1}{2} \sigma^2 x^2 \frac{\partial^2 p}{\partial x^2} + rx \frac{\partial p}{\partial x} - rp = 0.$$

To summarize, it follows from (11), (13), (14) and (15) that the two

unknowns p and \bar{S} are governed by the following system

$$\frac{\partial p}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 p}{\partial x^2} + x \left(r - \frac{\bar{S}'(t)}{\bar{S}(t)} \right) \frac{\partial p}{\partial x} - rp = 0 \quad \text{for } x > 1 \text{ and } 0 \leq t < T, \quad (19)$$

$$p(x, T) = 0 \quad \text{for } x \geq 1, \quad (20)$$

$$\frac{\partial p}{\partial x}(1, t) = -\bar{S}(t), \quad (21)$$

$$p(1, t) = E - \bar{S}(t), \quad (22)$$

$$\lim_{x \rightarrow \infty} p(x, t) = 0, \quad (23)$$

$$\bar{S}(T) = E. \quad (24)$$

If p and \bar{S} are computed by solving (19)-(24), then the value P of the American option is given by

$$P(S, t) = \begin{cases} p(S/\bar{S}(t), t) & \text{for } S/\bar{S}(t) \geq 1, \\ E - S & \text{for } 0 \leq S/\bar{S}(t) < 1. \end{cases} \quad (25)$$

3.1 An implicit scheme

In order to solve the system (19)-(24) numerically, we introduce x_∞ which is a large value of x where we impose the boundary condition¹ (5). That is, we put

$$p(x_\infty, t) = 0. \quad (26)$$

Next, for given positive integers M and N , we define

$$\begin{aligned} \Delta x &= \frac{x_\infty - 1}{M + 1}, & \Delta t &= \frac{T}{N + 1}, \\ x_j &= 1 + j\Delta x \quad \text{for } j = 0, \dots, M + 1, \\ t_n &= n\Delta t \quad \text{for } n = 0, \dots, N + 1. \end{aligned}$$

Our goal is to define an implicit method suitable for computing

$$p_j^n \approx p(x_j, t_n), \quad \text{for } j = 0, 1, \dots, M + 1 \text{ and } n = N, N - 1, \dots, 0,$$

¹Clearly, by applying a suitable change of variables we can transform the problem onto a finite domain. Thus avoiding the domain truncation parameter x_∞ in the discrete system. However, the resulting equation will in this case involve one or more unbounded coefficients.

and the associated front-position

$$\bar{S}^n \approx \bar{S}(t_n) \quad \text{for } n = N, N-1, \dots, 0.$$

Note that the final conditions (20) and (24) give

$$p_j^{N+1} = 0, \quad j = 0, 1, \dots, M+1, \quad (27)$$

and

$$\bar{S}^{N+1} = E. \quad (28)$$

The boundary conditions (22) and (26) imply that

$$p_0^n = E - \bar{S}^n \quad \text{for } n = N, N-1, \dots, 0, \quad (29)$$

and

$$p_{M+1}^n = 0 \quad \text{for } n = N, N-1, \dots, 0. \quad (30)$$

A finite difference approximation of (21) is given by

$$\frac{p_1^n - p_0^n}{\Delta x} = -\bar{S}^n \quad \text{for } n = N, N-1, \dots, 0,$$

which by (29) gives

$$p_1^n = E - (1 + \Delta x)\bar{S}^n \quad \text{for } n = N, N-1, \dots, 0 \quad (31)$$

Note that $\bar{S}'(t) \leq 0$, and consequently an upwind scheme can be applied to discretize the transport term of (19).

An implicit-upwind finite difference scheme for (19) is given by

$$\begin{aligned} & \frac{p_j^{n+1} - p_j^n}{\Delta t} + \frac{1}{2}\sigma^2 x_j^2 \frac{p_{j-1}^n - 2p_j^n + p_{j+1}^n}{(\Delta x)^2} + \\ & x_j \left(r - \frac{\bar{S}^{n+1} - \bar{S}^n}{\Delta t \bar{S}^n} \right) \frac{p_{j+1}^n - p_j^n}{\Delta x} - r p_j^n = 0, \end{aligned} \quad (32)$$

for $j = 1, \dots, M$ and $n = N, N-1, \dots, 0$. Here, p^{n+1} and \bar{S}^{n+1} are known and we want to compute p^n and \bar{S}^n . From (32) it follows that

$$\beta_j^n p_{j-1}^n + \alpha_j^n p_j^n + \gamma_j^n p_{j+1}^n = b_j, \quad (33)$$

and thus the system (38)-(41) can be written on the form

$$F(p^n, \bar{S}^n) = A(\bar{S}^n)p^n - f(\bar{S}^n) = 0, \quad (44)$$

where $p^n = (p_2^n, p_3^n, \dots, p_M^n)$. We will solve this nonlinear problem by Newton's method. To this end, let $y = (p_2^n, \dots, p_M^n, \bar{S}^n)$, and define the iteration

$$y_{k+1} = y_k - J^{-1}(y_k)F(y_k), \quad (45)$$

where J is the Jacobian of F . Having computed p_2^n, \dots, p_M^n and \bar{S}^n by (45) we apply the formulas (29), (30) and (31) to compute p_0^n, p_1^n and p_{M+1}^n .

3.2 An upwind explicit scheme

Now we want to define an upwind explicit scheme for the front-fixing method discussed above. That is, an upwind explicit numerical method for solving the problem posed in equations (19)-(24). If we apply the same notation as in section 3.1 the scheme is defined as follows

$$\begin{aligned} & \frac{p_j^{n+1} - p_j^n}{\Delta t} + \frac{1}{2}\sigma^2 x_j^2 \frac{p_{j-1}^{n+1} - 2p_j^{n+1} + p_{j+1}^{n+1}}{(\Delta x)^2} + \\ & x_j \left(r - \frac{\bar{S}^{n+1} - \bar{S}^n}{\Delta t \bar{S}^{n+1}} \right) \frac{p_{j+1}^{n+1} - p_j^{n+1}}{\Delta x} - r p_j^{n+1} = 0, \end{aligned} \quad (46)$$

for $j = 1, \dots, M$ and $n = N, N-1, \dots, 0$. Here, p^{n+1} and \bar{S}^{n+1} are given and the goal is to compute p^n and \bar{S}^n . The discrete final condition and boundary conditions are given in (27)-(31).

Some simple algebraic manipulations show that this problem can be written on the form

$$p_j^n - D_j^{n+1} \bar{S}^n = A_j p_{j-1}^{n+1} + B_j p_j^{n+1} + C_j p_{j+1}^{n+1}, \quad (47)$$

for $j = 1, \dots, M$ and $n = N, N-1, \dots, 0$, where

$$\begin{aligned} A_j &= \frac{1}{2}\sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2}, \\ B_j &= 1 - \sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} - x_j \left(r - \frac{1}{\Delta t} \right) \frac{\Delta t}{\Delta x} - r \Delta t, \\ C_j &= \frac{1}{2}\sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} + x_j \left(r - \frac{1}{\Delta t} \right) \frac{\Delta t}{\Delta x}, \\ D_j^n &= \frac{x_j}{\Delta x} \frac{p_{j+1}^n - p_j^n}{\bar{S}^n}. \end{aligned}$$

Notice that the coefficient D_j^{n+1} in (47) only depends on parameters computed at time step t_{n+1} .

In (47) we can use the boundary condition (31) to find a simple expression for the front-position \bar{S}^n . Put $j = 1$ in (47) and we get

$$p_1^n - D_1^{n+1} \bar{S}^n = A_1 p_0^{n+1} + B_1 p_1^{n+1} + C_1 p_2^{n+1},$$

and hence it follows from (31) that

$$\bar{S}^n = \frac{E - (A_1 p_0^{n+1} + B_1 p_1^{n+1} + C_1 p_2^{n+1})}{D_1^{n+1} + (1 + \Delta x)}.$$

The derivations given above lead to the following algorithm;

1. for $j = 0, 1, \dots, M + 1$ do $p_j^{N+1} = 0$.
2. $\bar{S}^{N+1} = E$.
3. for $n = N + 1, N, \dots, 0$ do $p_{M+1}^n = 0$.
4. for $j = 1, 2, \dots, M$ do

$$\begin{aligned} A_j &= \frac{1}{2} \sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2}, \\ B_j &= 1 - \sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} - x_j \left(r - \frac{1}{\Delta t} \right) \frac{\Delta t}{\Delta x} - r \Delta t, \\ C_j &= \frac{1}{2} \sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} + x_j \left(r - \frac{1}{\Delta t} \right) \frac{\Delta t}{\Delta x}. \end{aligned}$$

5. for $n = N, N - 1, \dots, 0$ do

- a) $D_j^{n+1} = \frac{x_j p_{j+1}^{n+1} - p_j^{n+1}}{\Delta x \bar{S}^{n+1}}$,
- b) $\bar{S}^n = \frac{E - (A_1 p_0^{n+1} + B_1 p_1^{n+1} + C_1 p_2^{n+1})}{D_1^{n+1} + (1 + \Delta x)}$
- c) $p_0^n = E - \bar{S}^n$ and $p_1^n = E - (1 + \Delta x) \bar{S}^n$.
- d) for $j = 2, 3, \dots, M$ do

$$p_j^n = A_j p_{j-1}^{n+1} + B_j p_j^{n+1} + C_j p_{j+1}^{n+1} + D_j^{n+1} \bar{S}^n.$$

3.3 A numerical experiment

Now we turn our attention to two simple numerical experiments illustrating the performance of the implicit and explicit front-fixing schemes derived above. More precisely, we solved our (transformed) model problem (19)-(24) with the following set of parameters

$$\begin{aligned}r &= 0.1, \\ \sigma &= 0.2, \\ E &= 1, \\ T &= 1, \\ x_\infty &= 2.\end{aligned}$$

Figure 2 shows the numerical solution, with discretization parameters $\Delta t = \Delta x = 0.001$, computed by the implicit method described in Section 3.1. The implementation was done within the Diffpack framework [6, 11]. Notice that the approximate option values generated by this algorithm will be used as a reference solution for testing the performance of the penalty method derived below, cf. Section 5.

This figure also shows the results computed by the explicit scheme presented in Section 3.2. In this case the computations were carried out in Matlab using the discretization parameters $\Delta t = 5.0 \cdot 10^{-6}$ and $\Delta x = 0.001$.

Clearly, the implicit and explicit front-fixing schemes provide almost identical results. In particular, the schemes computed the following front-positions at time $t = 0$,

$$\text{implicit; } \bar{S}^0 = 8.615 \cdot 10^{-1}, \quad \text{explicit; } \bar{S}^0 = 8.62 \cdot 10^{-1}.$$

Notice that, as expected, the time step for the explicit scheme is much smaller than for the implicit method.

Finally, we tested the influence of the domain truncation parameter x_∞ on the computations. By putting $x_\infty = 3$ and running the implicit scheme we got the front-position

$$\bar{S}^0 = 8.616 \cdot 10^{-1}$$

at time $t = 0$. Moreover, the approximate option values generated on this domain were almost identical to the results computed for $x_\infty = 2$ above. Hence, we concluded that $x_\infty = 2$ seems to be a sufficiently large domain truncation parameter for this model problem.

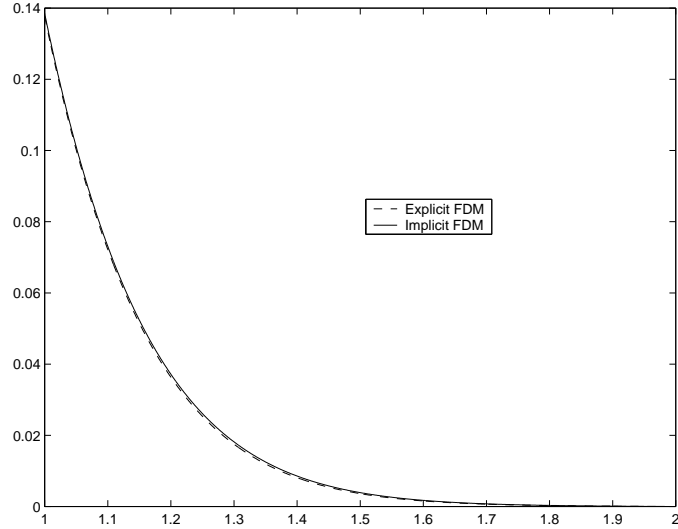


Figure 2: The figure shows the numerical results computed by the front-fixing method derived in Section 3. The solid and dashed lines are plots of the approximate solution of (19)-(24) computed by the explicit and implicit schemes, respectively.

4 Penalty methods

Now we turn our attention to penalty methods for solving free and moving boundary problems. In order to explain our approach, we start by giving two very simple problems chosen to illuminate the key properties of the method.

4.1 An ordinary differential equation

We start by considering a simple ordinary differential equation. Suppose we want to solve the system

$$\begin{aligned} u' &= -u, \\ u(0) &= 2, \end{aligned} \tag{48}$$

with the additional constraint that

$$u(t) \geq 1. \tag{49}$$

The solution of this problem can be computed analytically and is given by

$$u(t) = \begin{cases} 2e^{-t} & \text{for } t \leq \ln 2, \\ 1 & \text{for } t > \ln 2. \end{cases} \tag{50}$$

But suppose we want to solve the initial-value problem (48)-(49) numerically. Then we would have to check, for each time-step, whether the constraint is satisfied or not. Let u_n be a numerical approximation of $u(t_n)$ where $t_n = n\Delta t$, and $\Delta t > 0$ is the time step. We compute a numerical solution of the initial-value problem (48,49) using an explicit finite difference scheme;

$$u_{n+1} = \max((1 - \Delta t)u_n, 1) \text{ for } n \geq 0, \quad (51)$$

where $u_0 = 2$. This corresponds to a Brennan-Schwartz type of algorithm for pricing American put options, cf. [3].

What we would like is to simply solve a differential equation which automatically fulfills the extra requirement. An equation which approximates this property fairly well can be derived by adding an extra term to the equation given in (48). Consider the initial-value problem

$$\begin{aligned} v' &= -v + \frac{\epsilon}{v + \epsilon - 1}, \\ v(0) &= 2, \end{aligned} \quad (52)$$

where $\epsilon > 0$ is a small parameter. Note that initially, $v = 2$, so the penalty term

$$\frac{\epsilon}{v + \epsilon - 1}$$

is of order ϵ . The effect of the penalty term increases as v approaches its asymptotic solution given by

$$\bar{v} = 1. \quad (53)$$

We will show that the solution of the problem (52) satisfies

$$\bar{v} \leq v(t) \leq 2, \quad t \geq 0, \quad (54)$$

$$v'(t) \leq 0, \quad t \geq 0, \quad (55)$$

$$v''(t) \geq 0, \quad t \geq 0. \quad (56)$$

Note that for

$$\bar{v} \leq v \leq 2, \quad (57)$$

we have

$$v(v + \epsilon - 1) \geq \epsilon. \quad (58)$$

Hence, by (52) we have

$$v' = \frac{1}{v + \epsilon - 1}(\epsilon - v(v + \epsilon - 1)) \leq 0. \quad (59)$$

Since $v' \leq 0$ and $v' = 0$ for $v = \bar{v}$, we have proved that (54) and (55) hold. Next we note that

$$v'' = -v' \left(1 + \frac{\epsilon}{(v + \epsilon - 1)^2} \right), \quad (60)$$

and thus

$$v'' \geq 0. \quad (61)$$

4.2 A finite difference scheme

Next we consider a finite difference scheme for the initial value problem (52). Let v_n be a numerical approximation of $v(t_n)$ and consider the following scheme

$$v_{n+1} = (1 - \Delta t)v_n + \frac{\Delta t \epsilon}{v_n + \epsilon - 1} \quad \text{for } n \geq 1, \quad (62)$$

where $v_0 = 2$. We want to show that for sufficiently small Δt , the numerical solution satisfies

$$\bar{v} \leq v_n \leq 2 \quad n \geq 0, \quad (63)$$

$$v_{n+1} \leq v_n \quad \text{for } n \geq 0, \quad (64)$$

$$v_n \xrightarrow{n \rightarrow \infty} \bar{v}. \quad (65)$$

We assume that

$$\Delta t \leq \frac{\epsilon}{1 + \epsilon}. \quad (66)$$

Define

$$f(v) = (1 - \Delta t)v + \frac{\Delta t \epsilon}{v + \epsilon - 1}, \quad (67)$$

and note that

$$v_{n+1} = f(v_n).$$

We assume that

$$\bar{v} \leq v_n \leq 2. \quad (68)$$

Then

$$v_n(v_n + \epsilon - 1) \geq \epsilon, \quad (69)$$

and

$$v_{n+1} = v_n \left[1 - \Delta t + \frac{\Delta t \epsilon}{v_n(v_n + \epsilon - 1)} \right] \leq v_n \left[1 - \Delta t + \frac{\Delta t \epsilon}{\epsilon} \right] = v_n. \quad (70)$$

Note that

$$\begin{aligned} f'(v) &= 1 - \Delta t - \frac{\epsilon \Delta t}{(v + \epsilon - 1)^2} \\ &\geq 1 - \Delta t - \frac{\epsilon \Delta t}{\epsilon^2} \\ &= 1 - \Delta t - \frac{\Delta t}{\epsilon} \\ &\geq 1 - \frac{\epsilon}{1 + \epsilon} - \frac{1}{1 + \epsilon} \\ &= 0, \end{aligned}$$

where we have used the assumption (66). Hence, since f is an increasing function

$$v_{n+1} = f(v_n) \geq f(\bar{v}) = 1 - \Delta t + \frac{\Delta t \epsilon}{\epsilon} = \bar{v} = 1,$$

and thus (63) and (64) follows by induction on n . Finally we want to prove (65). Note first that

$$v_n \geq 1,$$

and consider

$$\begin{aligned} v_{n+1} - 1 &= v_n - 1 - \Delta t v_n + \frac{\Delta t \epsilon}{v_n + \epsilon - 1} \\ &= (v_n - 1)(1 - \Delta t) + \Delta t \left(\frac{\epsilon}{v_n + \epsilon - 1} - 1 \right) \\ &\leq (1 - \Delta t)(v_n - 1). \end{aligned}$$

By induction on n we have

$$0 \leq v_n - 1 \leq (1 - \Delta t)^n (v_0 - 1) = (1 - \Delta t)^n \xrightarrow{n \rightarrow \infty} 0,$$

hence

$$v_n \rightarrow 1 \text{ as } n \rightarrow \infty.$$

4.2.1 Experiments

Table 1 contains the discrete L_∞ error,

$$\|e\|_\infty = \max_n |u(t_n) - v_n|,$$

associated with the scheme (62) for solving (48)-(49). In these experiments we have computed the discrete solution in the time interval $[0, 2]$ with time steps

$$\Delta t = \frac{\epsilon}{1 + \epsilon},$$

see (66). Clearly, these results indicate that the approximations generated by the penalty method converges towards the correct solution as ϵ (and consequently Δt) tends to zero. Notice that the error (roughly) is of order ϵ .

Table 1: Numerical results generated by the penalty method presented in sections 4.1 and 4.2.

ϵ	Δt	$\ e\ _\infty$
10^{-1}	$9.09 \cdot 10^{-2}$	$6.35 \cdot 10^{-2}$
10^{-2}	$9.90 \cdot 10^{-3}$	$2.17 \cdot 10^{-2}$
10^{-3}	$9.99 \cdot 10^{-4}$	$3.79 \cdot 10^{-3}$
10^{-4}	$1.00 \cdot 10^{-4}$	$5.81 \cdot 10^{-4}$

4.3 A two point boundary value problem

To provide further insight into the penalty method, we consider the stationary problem; Find $\bar{S} \in \mathbb{R}$ and $u = u(S)$ such that

$$u(S) = 1 - S, \quad 0 \leq S \leq \bar{S}, \quad (71)$$

$$u''(S) = 1, \quad \bar{S} < S < 2, \quad (72)$$

$$u(\bar{S}) = 1 - \bar{S}, \quad (73)$$

$$u'(\bar{S}) = -1, \quad (74)$$

$$u(2) = 0. \quad (75)$$

This is a free boundary problem of the same flavor as the American put problem, cf. equations (1)-(7). However, it is a stationary problem, and hence easier to solve by analytical methods. The solution is given by

$$\bar{S} = 2 - \sqrt{2}, \quad (76)$$

$$u(S) = \begin{cases} 1 - S, & 0 \leq S \leq \bar{S}, \\ (\sqrt{2} - 1)(S - 1) + \frac{1}{2}(S - 2)^2, & \bar{S} < S \leq 2, \end{cases} \quad (77)$$

and is plotted in Figure 3. Note that the solution has the property that

$$u(S) \geq (1 - S) \quad \text{for all } S \in (0, 2). \quad (78)$$

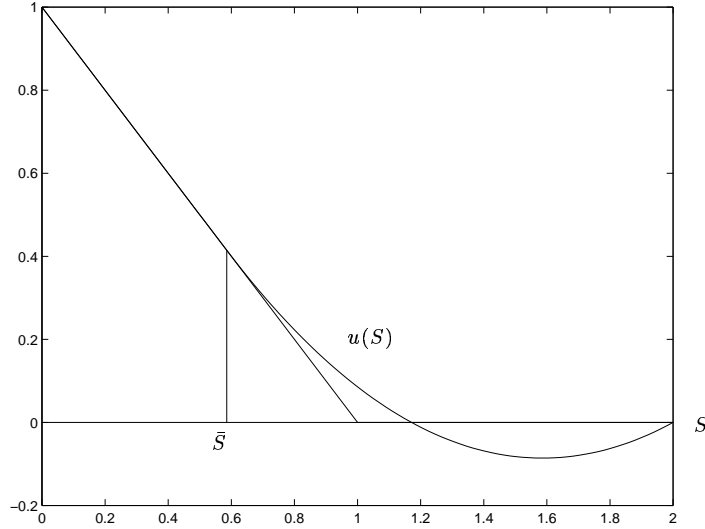


Figure 3: The solution of (71)-(75).

4.3.1 A penalty method

We want to derive a two-point boundary value problem with a solution that approximates the solution of (71) – (75) and in addition satisfies the extra requirement (78). To this end we consider the problem

$$v''(S) = 1 - \frac{\epsilon}{v + \epsilon - (1 - S)}, \quad 0 < S < 2, \quad (79)$$

$$v(0) = 1, \quad (80)$$

$$v(2) = 0, \quad (81)$$

where again $\epsilon > 0$ is a small parameter. We refer to this problem as the penalty formulation of the original problem (71) – (75). We discretize this equation by a finite difference method

$$v_{j-1} - 2v_j + v_{j+1} = (\Delta S)^2 - \frac{\epsilon(\Delta S)^2}{v_j + \epsilon - (1 - S_j)}, \quad j = 1, \dots, M, \quad (82)$$

where

$$\begin{aligned} v_j &\approx v(S_j) \quad \text{for } j=1, \dots, M, \\ v_0 &= 1, \\ v_{M+1} &= 0, \\ S_j &= j\Delta S, \\ \Delta S &= \frac{2}{M+1}, \end{aligned}$$

and M represents the number of inner mesh points in the domain $[0, 2]$. In the experiments below we solve this system of non-linear equations by Newton's method.

4.3.2 A front-fixing method

As for the American put problem we can solve (71) – (75) by front-fixing method. The change of variable

$$x = \frac{2 - S}{2 - \bar{S}}, \quad \text{i.e. } S = 2 - x(2 - \bar{S})$$

leads to problem

$$w''(x) = \frac{1}{(2 - \bar{S})^2}, \quad 0 < x < 1, \quad (83)$$

$$w(0) = 0, \quad (84)$$

$$w(1) = 1 - \bar{S}, \quad (85)$$

$$w'(1) = 2 - \bar{S}, \quad (86)$$

where $w(x) = u(S)$. Notice that (83)-(86) is posed on a fixed domain and that computing \bar{S} is a part of solving the problem. Therefore, a finite difference discretization of (83)-(86) will lead to a system of non-linear equations. However, we will not pursue this approach any further in this paper. Instead we turn our attention to some experiments with the penalty method discussed above.

4.3.3 Experiments

We will test the convergence properties of the penalty scheme derived in Section 4.3.1 using the discrete L_1 , L_2 and L_∞ norms. More precisely, for a discrete function g , defined on the mesh

$$(x_0, x_1, \dots, x_j, \dots, x_{M+1}),$$

we define the norms

$$\|g\|_1 = \Delta x \left[\frac{|g_0| + |g_{M+1}|}{2} + \sum_{j=1}^M |g_j| \right], \quad (87)$$

$$\|g\|_2 = \left(\Delta x \left[\frac{(g_0)^2 + (g_{M+1})^2}{2} + \sum_{j=1}^M (g_j)^2 \right] \right)^{1/2}, \quad (88)$$

$$\|g\|_\infty = \max_j |g_j|. \quad (89)$$

In Table 2 we have measured the error

$$e_j = u(S_j) - v_j, \quad j = 0, \dots, M + 1,$$

in these norms. Here, $u(S)$ is the analytical solution of (71)-(75), given in (76) and (77), and $\{v_j\}_{j=0}^{M+1}$ is the approximation computed by the scheme (82). Clearly, for this problem the penalty method provides satisfactory results (the error seems to be of order ϵ).

Table 2: Computational results obtained by the penalty method applied to the two point boundary value problem (71) – (74).

ΔS	ϵ	L_1	L_2	L_∞
$2 \cdot 10^{-3}$	$1.0 \cdot 10^{-1}$	$1.24 \cdot 10^{-1}$	$1.01 \cdot 10^{-1}$	$1.07 \cdot 10^{-1}$
$2 \cdot 10^{-3}$	$1.0 \cdot 10^{-2}$	$2.36 \cdot 10^{-2}$	$2.06 \cdot 10^{-2}$	$2.34 \cdot 10^{-2}$
$2 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$3.67 \cdot 10^{-3}$	$3.36 \cdot 10^{-3}$	$3.95 \cdot 10^{-3}$
$2 \cdot 10^{-3}$	$1.0 \cdot 10^{-4}$	$5.17 \cdot 10^{-4}$	$4.81 \cdot 10^{-4}$	$5.77 \cdot 10^{-4}$

5 A penalty method for Black-Scholes models

In this section we will modify the analysis presented above such that it can be applied to Black-Scholes models of American put options. As in Section 4 the original problem (1)-(7) is approximated by adding a penalty term to the equation (1). Thereby obtaining a non-linear parabolic partial differential equation posed on a fixed domain.

More precisely, let $0 < \epsilon \ll 1$ be a small regularization parameter and

consider the following initial-boundary value problem

$$\frac{\partial V_\epsilon}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V_\epsilon}{\partial S^2} + rS \frac{\partial V_\epsilon}{\partial S} - rV_\epsilon \quad (90)$$

$$+ \frac{\epsilon C}{V_\epsilon + \epsilon - q(S)} = 0, \quad S \geq 0, \quad t \in [0, T),$$

$$V_\epsilon(S, T) = \max(E - S, 0), \quad (91)$$

$$V_\epsilon(0, t) = E, \quad (92)$$

$$V_\epsilon(S, t) = 0 \quad \text{as } S \rightarrow \infty, \quad (93)$$

where $C \geq rE$ is a positive constant² and

$$q(S) = E - S, \quad (94)$$

see (8). Note again that the penalty term

$$\frac{\epsilon C}{V_\epsilon + \epsilon - q(S)}$$

is of order ϵ if $V_\epsilon = V_\epsilon(S, t) \gg q(S)$, and that it increases towards C as $V_\epsilon \rightarrow q(S)$.

Our goal is to define numerical methods for solving (90)-(93) and to prove that the approximate option values generated by the schemes satisfy a discrete version of (8). We will consider explicit, semi-implicit and fully implicit schemes.

5.1 An upwind explicit finite difference scheme

Clearly, we can only discretize (90)-(93) on a limited interval for the value S of the underlying asset. Thus, we introduce a parameter S_∞ (preferable a large number) representing the endpoint of discretization with respect to S . More precisely, we will discretize (90)-(93) in the domain

$$0 \leq S \leq S_\infty \quad \text{and} \quad 0 \leq t \leq T.$$

²We will show below why C should be larger or equal to rE .

Let, for given positive integers M and N ,

$$\begin{aligned}
\Delta S &= \frac{S_\infty}{M+1}, \quad \Delta t = \frac{T}{N+1}, \\
S_j &= j\Delta S, \quad j = 0, \dots, M+1, \\
t_n &= n\Delta t, \quad n = 0, \dots, N+1, \\
q_j &= q(S_j), \quad j = 0, \dots, M+1, \\
V_{\epsilon,j}^{N+1} &= \max(E - S_j, 0), \quad j = 1, \dots, M, \\
V_{\epsilon,0}^n &= E, \quad n = 0, \dots, N+1, \\
V_{\epsilon,M+1}^n &= 0, \quad n = 0, \dots, N+1 \\
V_{\epsilon,j}^n &\approx V_\epsilon(S_j, t_n), \quad j = 1, \dots, M \text{ and } n = 0, \dots, N.
\end{aligned}$$

For the sake of simplicity, we will omit the ϵ subscript in the discrete case and simply write V_j^n for $V_{\epsilon,j}^n$.

The discrete equations are derived by applying an upwind differencing of the transport term and a standard explicit time-stepping scheme for (90)-(93), i.e.

$$\begin{aligned}
&\frac{V_j^n - V_j^{n-1}}{\Delta t} + \frac{1}{2}\sigma^2 S_j^2 \frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{(\Delta S)^2} \\
&+ rS_j \frac{V_{j+1}^n - V_j^n}{\Delta S} - rV_j^n + \frac{\epsilon C}{V_j^n + \epsilon - q_j} = 0,
\end{aligned} \tag{95}$$

for $j = 1, \dots, M$ and $n = N+1, N, \dots, 1$. If we define the function

$$\begin{aligned}
f(V_-, V, V_+, q, S) &= [\alpha\sigma^2 S^2] V_- \\
&+ \left[1 - 2\alpha\sigma^2 S^2 - \frac{\Delta t}{\Delta S} rS - r\Delta t \right] V \\
&+ \left[\alpha\sigma^2 S^2 + \frac{\Delta t}{\Delta S} rS \right] V_+ \\
&+ \frac{\epsilon C \Delta t}{V + \epsilon - q},
\end{aligned} \tag{96}$$

where

$$\alpha = \frac{1}{2} \frac{\Delta t}{(\Delta S)^2},$$

then (95) can be written on the form

$$V_j^{n-1} = f(V_{j-1}^n, V_j^n, V_{j+1}^n, q_j, S_j), \quad j = 1, \dots, M \text{ and } n = N+1, \dots, 1. \tag{97}$$

With this notation at hand we are ready to start analyzing our explicit scheme. We start of by showing that the function f is increasing in the variables V_- , V and V_+ .

Lemma 1 *For all $S, r \geq 0$ the partial derivatives $\partial f/\partial V_-$ and $\partial f/\partial V_+$ of f are non-negative,*

$$\frac{\partial f}{\partial V_-}, \frac{\partial f}{\partial V_+} \geq 0.$$

Moreover,

$$\frac{\partial f}{\partial V} \geq 0 \quad \text{for all } V \geq q,$$

provided that Δt satisfies

$$\Delta t \leq \frac{(\Delta S)^2}{\sigma^2 S_\infty^2 + r S_\infty (\Delta S) + r (\Delta S)^2 + \frac{C}{\epsilon} (\Delta S)^2}. \quad (98)$$

Proof. Clearly,

$$\frac{\partial f}{\partial V_-} = \alpha \sigma^2 S^2 \geq 0, \quad \frac{\partial f}{\partial V_+} = \alpha \sigma^2 S^2 + \frac{\Delta t}{\Delta S} r S \geq 0,$$

for all $S, r \geq 0$.

Next,

$$\frac{\partial f}{\partial V} = 1 - 2\alpha \sigma^2 S^2 - \frac{\Delta t}{\Delta S} r S - r \Delta t - \frac{\epsilon C \Delta t}{(V + \epsilon - q)^2},$$

and for $V \geq q$ it follows that

$$\begin{aligned} \frac{\partial f}{\partial V} &\geq 1 - \frac{\Delta t}{(\Delta S)^2} \sigma^2 S^2 - \frac{\Delta t}{\Delta S} r S - r \Delta t - \frac{\epsilon C \Delta t}{\epsilon^2} \\ &= 1 - \Delta t \left[\frac{\sigma^2 S^2}{(\Delta S)^2} + \frac{r S}{\Delta S} + r + \frac{C}{\epsilon} \right] \geq 0, \end{aligned}$$

provided that Δt satisfies (98). ■

As mentioned above, we prove that the approximate option values generated by our explicit scheme fulfills a discrete analogue to (8).

Theorem 2 *For all $C \geq rE$, $S_\infty \geq E$ and all Δt satisfying (98) the approximate option values $\{V_j^n\}$ generated by the scheme (97) satisfy*

$$V_j^n \geq \max(E - S_j, 0), \quad (99)$$

for $j = 0, \dots, M + 1$ and $n = N + 1, \dots, 0$.

Proof. By definition, $V_j^{N+1} = \max(E - S_j, 0)$ for $j = 0, \dots, M + 1$, and hence (99) holds for $n = N + 1$. Furthermore, $V_0^n = E = \max(E - S_0, 0)$ and $V_{M+1}^n = 0 = \max(E - S_{M+1}, 0)$ for $n = N + 1, \dots, 0$, provided that $S_\infty \geq E$.

Next, we will prove that if (99) holds for n then it must also be valid for $n - 1$. Let $q(S)$ be the function defined in (94) and notice that

$$\max(E - S_j, 0) = \max(q_j, 0).$$

If (99) holds for n then it follows from the definition (97) of our scheme and Lemma 1 that

$$V_j^{n-1} = f(V_{j-1}^n, V_j^n, V_{j+1}^n, q_j, S_j) \geq f(q_{j-1}, q_j, q_{j+1}, q_j, S_j). \quad (100)$$

Notice that, cf. the definition (94) of q ,

$$q_{j-1} = q_j + \Delta S \text{ and } q_{j+1} = q_j - \Delta S.$$

Next, from the definition (96) of f we find that

$$\begin{aligned} V_j^{n-1} &\geq f(q_{j-1}, q_j, q_{j+1}, q_j, S_j) = q_j - r\Delta t q_j - \frac{\Delta t}{\Delta S} r S_j \Delta S + \frac{\epsilon C \Delta t}{q_j + \epsilon - q_j} \\ &= q_j - r\Delta t q_j - r\Delta t S_j + C \Delta t. \end{aligned}$$

Recall that $q_j = E - S_j$, cf. (94), and consequently

$$\begin{aligned} V_j^{n-1} &\geq q_j - r\Delta t E + r\Delta t S_j - r\Delta t S_j + C \Delta t \\ &= q_j + (C - rE)\Delta t \geq q_j, \end{aligned}$$

provided that $C \geq rE$.

If (98) holds and $\{V_j^n\}$ satisfies (99) it follows directly from (97) and (96) that

$$V_j^{n-1} \geq 0.$$

Hence,

$$V_j^{n-1} \geq \max(q_j, 0),$$

and we conclude by induction that the theorem must hold. ■

5.2 An implicit and a semi-implicit scheme

We observed above that the explicit scheme puts severe restrictions on the time steps leading to unacceptable computing times. This is particularly so in the case of multi-asset options, cf. [12]. In the present section we will derive an implicit and a semi-implicit scheme. For both schemes we assume that

$$C \geq rE. \quad (101)$$

Under this mild assumption it turns out that the implicit scheme is stable whereas the semi-implicit scheme is stable if the additional condition

$$\Delta t \leq \frac{\epsilon}{rE}, \quad (102)$$

is satisfied. Note that this condition is a significantly milder restriction than (98). In order to avoid solution of non-linear algebraic systems, we find the use of a semi-implicit scheme to be an attractive alternative.

Using the notation introduced above, we consider the following scheme,

$$\begin{aligned} \frac{V_j^{n+1} - V_j^n}{\Delta t} + \frac{1}{2}\sigma^2 S_j^2 \frac{V_{j-1}^n - 2V_j^n + V_{j+1}^n}{(\Delta S)^2} \\ + rS_j \frac{V_{j+1}^n - V_j^n}{\Delta S} - rV_j^n + \frac{\epsilon C}{V_j^{n+1/2} + \epsilon - q_j} = 0. \end{aligned} \quad (103)$$

Here, we put $V_j^{n+1/2} = V_j^n$ in the fully implicit scheme, and $V_j^{n+1/2} = V_j^{n+1}$ in the semi-implicit scheme. The scheme (103) can be rearranged as

$$(1 + r\Delta t + 2\alpha_j + \beta_j)V_j^n = V_j^{n+1} + (\alpha_j + \beta_j)V_{j+1}^n + \alpha_j V_{j-1}^n + \frac{\epsilon\Delta t C}{V_j^{n+1/2} + \epsilon - q_j}, \quad (104)$$

where

$$\alpha_j = \frac{1}{2} \frac{\Delta t}{(\Delta S)^2} S_j^2 \sigma^2, \quad \beta_j = r \frac{\Delta t}{\Delta S} S_j. \quad (105)$$

Our aim is to show that

$$V_j^n \geq \max(q_j, 0), \quad \forall j, n. \quad (106)$$

We do this in two steps; first we show that

$$V_j^n \geq q_j, \quad \forall j, \quad (107)$$

and next we show that

$$V_j^n \geq 0. \quad (108)$$

In order to prove (107), we introduce

$$u_j^n = V_j^n - q_j. \quad (109)$$

The scheme for $\{u_j^n\}$ then reads

$$\begin{aligned} (1 + r\Delta t + 2\alpha_j + \beta_j)u_j^n &= u_j^{n+1} + (\alpha_j + \beta_j)u_{j+1}^n \\ &+ \alpha_j u_{j-1}^n + \frac{\epsilon\Delta t C}{u_j^{n+1/2} + \epsilon} - r\Delta t E, \end{aligned} \quad (110)$$

where $u_j^{n+1/2} = u_j^n$ in the fully implicit case, and $u_j^{n+1/2} = u_j^{n+1}$ in the semi-implicit case.

Define

$$u^n = \min_j u_j^n, \quad (111)$$

and let k be an index such that

$$u_k^n = u^n. \quad (112)$$

For $j = k$, it follows from (110) that

$$\begin{aligned} (1 + r\Delta t + 2\alpha_k + \beta_k)u^n &\geq u_k^{n+1} + (\alpha_k + \beta_k)u^n \\ &+ \alpha_k u^n + \frac{\epsilon\Delta t C}{u_k^{n+1/2} + \epsilon} - r\Delta t E, \end{aligned} \quad (113)$$

or

$$(1 + r\Delta t)u^n \geq u_k^{n+1} + \frac{\epsilon\Delta t C}{u_k^{n+1/2} + \epsilon} - r\Delta t E. \quad (114)$$

Let us now consider the fully implicit case. Then (114) takes the form

$$(1 + r\Delta t)u^n - \frac{\epsilon\Delta t C}{u^n + \epsilon} + r\Delta t E \geq u_k^{n+1} \geq u^{n+1}. \quad (115)$$

If we assume that

$$u^{n+1} \geq 0, \quad (116)$$

we have

$$F(u^n) \geq 0, \quad (117)$$

where

$$F(u) = (1 + r\Delta t)u - \frac{\epsilon\Delta tC}{u + \epsilon} + r\Delta tE. \quad (118)$$

Since

$$F(0) = \Delta t(rE - C) \leq 0, \quad (119)$$

cf. (101), and

$$F'(u) = 1 + r\Delta t + \frac{\epsilon\Delta tC}{(u + \epsilon)^2} > 0, \quad (120)$$

it follows from (117) that

$$u^n \geq 0. \quad (121)$$

Consequently, by induction on n , it follows from (109) that

$$V_j^n \geq q_j, \quad \forall j \quad (122)$$

for $n = N + 1, N, N - 1, \dots, 0$.

Next we consider the semi-implicit scheme and we assume that (102) holds. It follows from (114) that

$$(1 + r\Delta t)u^n \geq \frac{u_k^{n+1}(u_k^{n+1} + \epsilon) + \epsilon\Delta tC - r\Delta tE(u_k^{n+1} + \epsilon)}{u_k^{n+1} + \epsilon} \quad (123)$$

We assume that $u^{n+1} \geq 0$, and thus $u_k^{n+1} \geq 0$. Let

$$G(u) = u(u + \epsilon) + \epsilon\Delta tC - r\Delta tE(u + \epsilon). \quad (124)$$

Then

$$G(0) = \Delta t\epsilon(C - Er) \geq 0, \quad (125)$$

cf. (101), and

$$G'(u) = 2u + \epsilon - r\Delta tE, \quad (126)$$

so $G'(u) \geq 0$ for $u \geq 0$ provided that (102) holds. Hence we have

$$u_j^n \geq 0, \quad (127)$$

and thus by (109)

$$V_j^n \geq q_j, \quad \forall j,$$

for $n = N + 1, N, N - 1, \dots, 0$.

Next we consider (108), i.e. we want to show that

$$V_j^n \geq 0. \quad (128)$$

As above, we define

$$V^n = \min_j V_j^n, \quad (129)$$

and let k be an index such that

$$V_k^n = V^n. \quad (130)$$

It follows from (104) that

$$\begin{aligned} (1 + r\Delta t + 2\alpha_k + \beta_k)V^n &\geq V^{n+1} + (\alpha_k + \beta_k)V^n + \alpha_k V^n \\ &\quad + \frac{\epsilon\Delta t C}{V_k^{n+1/2} + \epsilon - q_k}, \end{aligned} \quad (131)$$

or

$$(1 + r\Delta t)V^n \geq V^{n+1} + \frac{\epsilon\Delta t C}{V_k^{n+1/2} + \epsilon - q_k}. \quad (132)$$

Since we have just seen that

$$V_k^{n+1/2} \geq q_k, \quad (133)$$

both in the fully implicit and the semi-implicit case, it follows from (132) that

$$(1 + r\Delta t)V^n \geq V^{n+1}, \quad (134)$$

and then it follows by induction on n that

$$V_j^n \geq 0, \quad \forall j, \quad (135)$$

$n = N + 1, N, N - 1, \dots, 0$.

Theorem 3 *Suppose (101) holds, then the numerical solution computed by the fully implicit scheme (103) satisfies the bound*

$$V_j^n \geq \max(E - S_j, 0) \quad \forall j \quad (136)$$

$n = N + 1, N, N - 1, \dots, 0$.

Similarly, if (101) and (102) hold, the numerical solution computed by the semi-implicit version of (103) satisfies the bound (136).

5.3 Numerical experiments

To provide further insight we will test the proposed penalty schemes on the model problem discussed in section 3.3. The main purpose of these experiments is to study the convergence properties of the method and to illustrate numerically that the inequality constraint, imposed by the possibility of early exercise, is fulfilled, i.e. that the discrete analogue to (8) hold.

Consider the discrete L_1 , L_2 and L_∞ norms defined in Section 4.3.3. Clearly, these norms are only capable of measuring the convergence of the approximate option values generated by the schemes. Assume for a moment that we are working on a hedging problem for a portfolio. In such cases we might be interested in computing the Greeks, involving the derivatives of the option value function, associated with the portfolio, cf. e.g. [15]. Hence, the convergence properties of the discrete derivatives, implicitly defined by our methods, are also of interest. Consequently, we will also measure the convergence properties of our schemes in the discrete first order Sobolev H^1 norm, and study whether or not the discrete derivatives converge in L_1 sense,

$$\begin{aligned} \|g(\cdot, t_n)\|_{H^1} &= \left[\|g(\cdot, t_n)\|_2^2 + \frac{1}{\Delta x} \sum_{j=1}^{M+1} (g_j^n - g_{j-1}^n)^2 \right]^{1/2}, \\ |g(\cdot, t_n)|_1 &= \sum_{j=1}^{M+1} \frac{|g_j^n - g_{j-1}^n|}{\Delta x} \Delta x = \sum_{j=1}^{M+1} |g_j^n - g_{j-1}^n|. \end{aligned}$$

Notice that $|\cdot|_1$ only defines a semi-norm on the set of discrete functions defined on the mesh.

We solve our model problem using the same set of parameters as in Section 3.3 and $C = rE$, cf. theorems 2 and 3. For a decreasing sequence of ϵ -values $\epsilon_0, \epsilon_1, \dots, \epsilon_I$,

$$\epsilon_0 = \frac{1}{10} \quad \text{and} \quad \epsilon_i = \frac{\epsilon_{i-1}}{10} \quad \text{for } i = 1, \dots, I,$$

we compare the approximate option values generated by (97) (and the approximate option values generated by the implicit and semi-implicit versions of (103)) with the results computed by the implicit front-fixing method in Section 3.3. That is, since no analytical solution of the problem is available, we use the approximate option values generated by the implicit front-fixing scheme on a fine mesh as a reference solution. More precisely, let P_f represent the discrete solution obtained by the front-fixing method. Then we compute the error $e_{\epsilon_i}(\cdot, \cdot)$, associated with the scheme (97) (and the error associated the implicit and semi-implicit versions of (103))

$$e_{\epsilon_i}(x_j, t_n) = e_j^n = P_f(x_j, t_n) - V_{\epsilon_i, j}^n$$

for $j = 0, 1, \dots, M + 1$ and $n = 0, 1, \dots, N + 1$.

In the tables below we present the L_1 , L_2 , L_∞ and H^1 norms of $e_{\epsilon_i}(\cdot, t_0)$ for each value of ϵ_i . In addition we compute the $|\cdot|_1$ semi-norm of $e_{\epsilon_i}(\cdot, t_0)$. Finally, we also test if Theorem 2 and Theorem 3 hold, i.e. we compute

$$\phi = \min_{j, n} (V_{\epsilon_i, j}^n - q_j)$$

for each value of ϵ_i .

5.3.1 Case I

As mentioned above, we use the same model parameters as in Section 3.3 (replacing $x_\infty = 2$ with $S_\infty = 2$). The implementations of the schemes are based on the C++ class library Diffpack [11].

The results reported in tables 3, 4 and 5 have been computed as follows

Table 3; We used the upwind explicit finite difference scheme (95) with discretization parameters $\Delta S = 1.0 \cdot 10^{-3}$ and Δt computed according to (98).

Table 4; These are the results generated by the fully implicit version of the scheme (103). In these computations we applied the discretization parameters $\Delta S = \Delta t = 1.0 \cdot 10^{-3}$.

Table 5; This table contains the numbers generated by the semi-implicit version of the finite difference scheme (103). We applied the discretization parameters $\Delta S = 1.0 \cdot 10^{-3}$ and $\Delta t = 5.0 \cdot 10^{-4}$. Hence, condition (102) is satisfied for all values of ϵ used in these experiments.

We observe that the three schemes generate results which are consistent with the approximate option values provided by the implicit front-fixing method.

More precisely, it seems like the estimated option values provided by the penalty schemes converge towards the reference solution as $\epsilon \rightarrow 0$. This is the case, not only for the L_1, L_2, L_∞ norms, but also for the H^1 norm and the $|\cdot|_1$ semi-norm taking the discrete derivatives into consideration. Furthermore, the constraint imposed by the possibility of early exercise hold, i.e. $\phi = 0$ in all three tables (Notice that, due to the final condition $V_{\epsilon_i, j}^{N+1} = q_j$, ϕ will always satisfy $\phi \leq 0$. Consequently, if the conditions of theorems 2 and 3 hold then ϕ must be zero).

Notice that the computational efficiency of the schemes differ significantly. Due to the severe restriction (98) on the time-steps, the explicit scheme is much slower than the fully-implicit and semi-implicit methods. Moreover, the semi-implicit scheme is significantly faster than the fully-implicit method. Recall that in the fully-implicit case we must solve a system of nonlinear equations at each time-step whereas for the semi-implicit scheme it is sufficient to solve a tridiagonal linear system at each time-step. In these experiments the solution of the nonlinear problems in the fully-implicit case required 3 – 4 Newton-iterations (at an average).

Table 3: The penalty method applied to the American put problem, explicit time stepping, $\phi = 0.0$ for all values of ϵ .

ϵ	L_1	L_2	L_∞	H_1	$ \cdot _1$	CPU-time
10^{-1}	$2.50 \cdot 10^{-2}$	$2.61 \cdot 10^{-2}$	$4.23 \cdot 10^{-2}$	$1.00 \cdot 10^{-1}$	$9.39 \cdot 10^{-3}$	129.5s
10^{-2}	$5.04 \cdot 10^{-3}$	$6.31 \cdot 10^{-3}$	$1.32 \cdot 10^{-2}$	$4.05 \cdot 10^{-2}$	$1.60 \cdot 10^{-3}$	129.5s
10^{-3}	$6.22 \cdot 10^{-4}$	$9.49 \cdot 10^{-4}$	$2.50 \cdot 10^{-3}$	$1.10 \cdot 10^{-2}$	$1.21 \cdot 10^{-4}$	129.6s
10^{-4}	$1.18 \cdot 10^{-4}$	$1.51 \cdot 10^{-4}$	$3.02 \cdot 10^{-4}$	$2.50 \cdot 10^{-3}$	$6.29 \cdot 10^{-6}$	130.2s

Table 4: The penalty method applied to the American put problem, implicit time stepping, $\phi = 0.0$ for all values of ϵ .

ϵ	L_1	L_2	L_∞	H_1	$ \cdot _1$	CPU-time
10^{-1}	$2.50 \cdot 10^{-2}$	$2.61 \cdot 10^{-2}$	$4.23 \cdot 10^{-2}$	$1.00 \cdot 10^{-1}$	$9.38 \cdot 10^{-3}$	7.8s
10^{-2}	$5.03 \cdot 10^{-3}$	$6.30 \cdot 10^{-3}$	$1.32 \cdot 10^{-2}$	$4.05 \cdot 10^{-2}$	$1.60 \cdot 10^{-3}$	7.8s
10^{-3}	$6.19 \cdot 10^{-4}$	$9.45 \cdot 10^{-4}$	$2.49 \cdot 10^{-3}$	$1.10 \cdot 10^{-2}$	$1.21 \cdot 10^{-4}$	7.8s
10^{-4}	$1.20 \cdot 10^{-4}$	$1.54 \cdot 10^{-4}$	$2.99 \cdot 10^{-4}$	$2.50 \cdot 10^{-3}$	$6.32 \cdot 10^{-6}$	8.4s

5.3.2 Case II

For the upwind explicit finite difference method, presented in Section 5.1, we showed that if (98) hold then the scheme satisfies a discrete analogue to the

Table 5: The penalty method applied to the American put problem, semi-implicit time stepping, $\phi = 0.0$ for all values of ϵ .

ϵ	L_1	L_2	L_∞	H_1	$ \cdot _1$	CPU-time
10^{-1}	$2.50 \cdot 10^{-2}$	$2.61 \cdot 10^{-2}$	$4.23 \cdot 10^{-2}$	$1.00 \cdot 10^{-1}$	$9.39 \cdot 10^{-3}$	2.8s
10^{-2}	$5.03 \cdot 10^{-3}$	$6.31 \cdot 10^{-3}$	$1.32 \cdot 10^{-2}$	$4.05 \cdot 10^{-2}$	$1.60 \cdot 10^{-3}$	2.8s
10^{-3}	$6.21 \cdot 10^{-4}$	$9.48 \cdot 10^{-4}$	$2.49 \cdot 10^{-3}$	$1.10 \cdot 10^{-2}$	$1.21 \cdot 10^{-4}$	2.8s
10^{-4}	$1.19 \cdot 10^{-4}$	$1.52 \cdot 10^{-4}$	$3.01 \cdot 10^{-4}$	$2.49 \cdot 10^{-3}$	$6.31 \cdot 10^{-6}$	2.8s

early exercise constraint (8), cf. Theorem 2. We tested the necessity of this condition by increasing the time-step size used in Case I by 1.5%. The scheme broke down. In particular, for $\epsilon = 0.1$ we observed that $\phi = -3.93 \cdot 10^6$.

The restriction (102) on Δt for the semi-implicit scheme is milder. However, by choosing $\Delta t = 1.0 \cdot 10^{-3}$ and $\epsilon = 5.0 \cdot 10^{-5}$, and hence violating (102), we observed that $\phi = -2.72 \cdot 10^{-8}$. Thus leading to unacceptable results.

Recall that we proved theorems 2 and 3 assuming that the constant C in the penalty term is larger or equal to the product of the interest rate r and the exercise price E of the option. We tried to replace $C = rE$ by $C = 0.9 \cdot rE$ in the experiments reported in Case I for the fully implicit scheme. This led to approximate option values not satisfying the lower bound (136). The scheme became unstable and $\phi \approx -1.0 \cdot 10^{-6}$.

Finally, it should be mentioned that all computations reported in this paper have been carried out on a dual Dell Workstation 410 with two PIII 600 MHz microprocessors and 1GB ram, running the Linux operating system.

References

- [1] K. Amin and A. Khanna. Convergence of american option values from discrete to continuous-time financial models. *Mathematical Finance*, 4:289–304, 1994.
- [2] J. Barraquand and T. Pudet. *Pricing of American Path-Dependent Contingent Claims*. Digital Research Laboratory, Paris, 1994.
- [3] M. Brennan and E. Schwartz. The valuation of american put options. *Journal of Finance*, 32:449–462, 1977.
- [4] M. Broadie and J. Detemple. American option valuation: New bounds, approximations, and comparison of existing methods. *Review of Financial Studies*, 1994.

- [5] J. Crank. *Free and moving boundary problems*. Oxford University Press, 1984.
- [6] Diffpack. <http://www.nobjects.no/>.
- [7] D. Duffie. *Dynamic asset pricing theory*. Princeton University Press, 1996.
- [8] J. C. Hull. *Options, futures and other derivatives*. Prentice Hall, 1997.
- [9] P. Jaillet, D. Lamberton, and B. Lapeyre. Variational-inequalities and the pricing of american options. *ACTA Applicandae Mathematicae*, 21(3):263–289, 1990.
- [10] Y. K. Kwok. *Mathematical models of financial derivatives*. Springer-Verlag, 1998.
- [11] H. P. Langtangen. *Computational partial differential equations. Numerical methods and Diffpack programming*. Springer-Verlag, 1999.
- [12] B. F. Nielsen, O. Skavhaug, and A. Tveito. Penalty methods for the numerical solution of american multi-asset option problems. Technical report, Department of Informatics, University of Oslo, In preparation.
- [13] S. H. Ross. *An introduction to mathematical finance*. Cambridge University Press, 1999.
- [14] P. Wilmott. *Derivatives, The theory and practice of financial engineering*. John Wiley & Sons, 1998.
- [15] P. Wilmott, J. Dewynne, and S. Howison. *Option Pricing, Mathematical models and computation*. Oxford Financial Press, 1993.
- [16] Y. I. Zhu and A. Abifaker. Applications of the singularity-separating method to double moving barrier options with rebates. Technical report, University of North Carolina at Charlotte, 1999.
- [17] Y. I. Zhu, H. Ren, and H. Xu. Improved effectiveness evaluating american options by the singularity-separating method. Technical report, University of North Carolina at Charlotte, 1997.
- [18] Y. I. Zhu and Y. Sun. The singularity-separating method for two-factor convertible bonds. *The Journal of Computational Finance*, 3(1):91–110, 1999.

- [19] R. Zvan, P. A. Forsyth, and K. R. Vetzal. Penalty methods for american options with stochastic volatility. *Journal of Computational and Applied Mathematics*, 91:199–218, 1998.