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

Article *in* Journal of Computational Finance · September 2001

DOI: 10.21314/JCF.2002.084 · Source: CiteSeer

Ola Skavhaug
Simula Research Laboratory

Aslak Tveito
Simula Research Laboratory

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

Bjørn Fredrik Nielsen
Simula Research Laboratory, Martin Linges v 17, Fornebu, PO Box 134, 1325 Lysaker, Norway

Ola Skavhaug
Simula Research Laboratory, Martin Linges v 17, Fornebu, PO Box 134, 1325 Lysaker, Norway

Aslak Tveito
Simula Research Laboratory, Martin Linges v 17, Fornebu, PO Box 134, 1325 Lysaker, Norway

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, continuous penalty term to the Black–Scholes equation. The problem can then be solved on a fixed domain, thus removing the difficulties associated with a moving boundary. To gain insight into 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 mimic the basic properties of the analytical solution to 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 is illustrated using a series of numerical experiments.

1 Introduction

Analytical solutions of Black–Scholes models of American option problems are seldom available, so such derivatives must be priced by numerical techniques. The problem of solving the American option problem numerically has been the subject of intensive research during the last decade (eg, Amin and Khanna, 1994; Barraquand and Pudet, 1994; Broadie and Detemple, 1996). Elementary introductions to the topic can be found in, for example, Kwok (1998), Ross (1999), Wilmott (1998), Wilmott, Dewynne and Howison (1993). In this paper we introduce two schemes – a front-fixing scheme and a penalty method – for solving the free and moving boundary value problem that arises in such models. 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 (1998). Our objective here is to derive a refinement of their approach that is easy to generalize to any American type of option. We do this by adding a term to the partial differential equation which assures that the solution will stay in the proper state space while altering the exact solution as little as possible. For explicit, semi-implicit and fully implicit numerical schemes we derive conditions which assure that the approximate option values satisfy the basic properties of the analytical solution of the problem. The performance of the schemes is 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 tends to zero.

The front-fixing method has been applied successfully to a wide range of problems arising in physics (see Crank, 1984, and references therein). The basic idea is to remove the moving boundary by a transformation of the variables involved. 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 (1997). They apply a singularity-separating method to derive an equation for the difference between the value of an American and a European option and then map the latter problem on to a fixed domain. In Zhu and Abifaker (1999) and Zhu and Sun (1999) 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 on to 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.

Several schemes for solving option problems have been proposed in addition to penalty, singularity-separating and front-fixing methods. Among these are the Brennan and Schwartz algorithm (Brennan and Schwartz, 1977; Jaillet, Lamberton and Lapeyre, 1990), the projected SOR scheme (Wilmott, Dewynne and Howison, 1993), the binomial method (Hull, 1997) and Monte Carlo simulation techniques (Duffie, 1996; Ross, 1999; Wilmott, 1998).

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 a simple model problem. In this section we present such methods, along with their convergence properties and numerical experiments, for an ordinary differential equation. 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:

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + rS\frac{\partial P}{\partial S} - rP = 0 \qquad \text{for } S > S(t) \text{ and } 0 \le t \le T \tag{1}$$

$$P(S, T) = \max(E - S, 0) \qquad \text{for } S \ge 0 \tag{2}$$

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

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

$$\lim_{S \to \infty} P(S, t) = 0 \tag{5}$$

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

$$P(S, t) = E - S \qquad \text{for } 0 \le S < \bar{S}(t) \tag{7}$$

where $\bar{S}(t)$ represents the free (and moving) boundary (see, eg, Duffie, 1996, Kwok, 1998, or Wilmott, Dewynne and Howison, 1993). 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. We assume that no dividend is paid during the life span of the contract.[1] Note that, since early exercise is permitted, the value, P, of the option must satisfy

$$P(S, t) \ge \max(E - S, 0) \qquad \text{for all } S \ge 0 \text{ and } 0 \le \mathrm{t} \le T \tag{8}$$

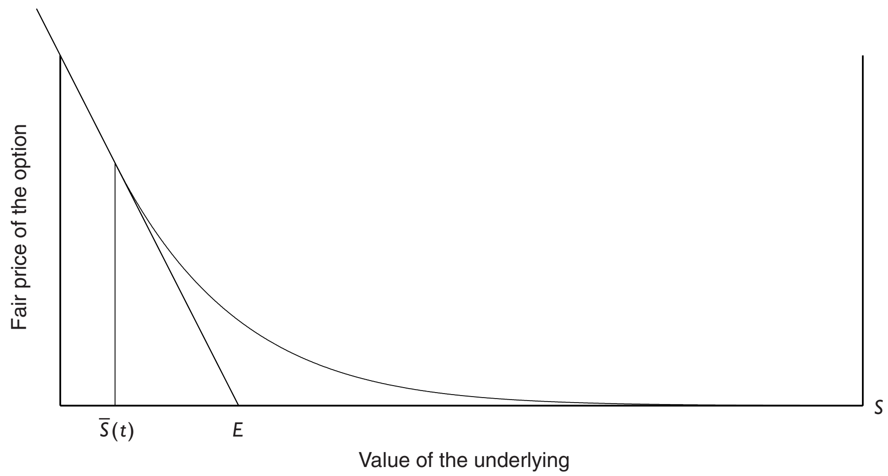
(compare Wilmott, Dewynne and Howison, 1993). Mathematical models of this kind, involving a moving boundary, are frequently referred to as moving boundary problems (see Crank, 1984, and Figure 1).

3 A front-fixing method

The basic idea behind the front-fixing method is to remove the moving boundary in the American option problem through a change of variables. It turns out that this approach leads to a nonlinear problem posed on a fixed domain. In this formulation of the problem the position of the boundary is given but some of the boundary conditions remain unknown and must therefore be computed.

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

FIGURE 1 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



$$x = \frac{S}{\bar{S}(t)} \quad \text{or} \quad S = x\bar{S}(t) \tag{9}$$

and

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

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

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

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

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

and thus (3) implies that

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

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

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

and

$$\lim_{x\to\infty} p(x,t) = \lim_{x\to\infty} P\big(x\bar{S}(t),t\big) = 0 \tag{15}$$

A partial differential equation for $p(x,t)$ is derived from (1), which governs $P(S,t)$. 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

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

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

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

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

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

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:

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

$$p(x, T) = 0 \qquad \text{for } x \ge 1 \tag{20}$$

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

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

$$\lim_{x\to\infty} p(x, t) = 0 \tag{23}$$

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

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

$$P(S, t) = \begin{cases} p\big(S/\bar{S}(t), t\big) & \text{for } S/\bar{S}(t) \ge 1 \\ E - S & \text{for } 0 \le S/\bar{S}(t) < 1 \end{cases} \tag{25}$$

3.1 An implicit scheme

To solve the system (19)–(24) numerically, we introduce x_∞, which is a large value of x where we impose the boundary condition (5).[2] That is, we put

$$p(x_\infty, t) = 0 \tag{26}$$

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

$$\Delta x = \frac{x_\infty - 1}{M + 1}, \qquad \Delta t = \frac{T}{N + 1}$$

$$x_j = 1 + j\Delta x \qquad \text{for } j = 0, \ldots, M + 1$$

$$t_n = n\Delta t \qquad \text{for } n = 0, \ldots, N + 1$$

Our goal is to define an implicit method suitable for computing

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

and the associated front position

$$\overline{S}^n \approx \overline{S}(t_n) \qquad \text{for } n = N, N-1, \ldots, 0$$

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

$$p_j^{N+1} = 0, \qquad j = 0, 1, \ldots, M+1 \tag{27}$$

and

$$\overline{S}^{N+1} = E \tag{28}$$

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

$$p_0^n = E - \overline{S}^n \qquad \text{for } n = N, N-1, \ldots, 0 \tag{29}$$

and

$$p_{M+1}^n = 0 \qquad \text{for } n = N, N-1, \ldots, 0 \tag{30}$$

A finite difference approximation of (21) is given by

$$\frac{p_1^n - p_0^n}{\Delta x} = -\overline{S}^n \qquad \text{for } n = N,\ N-1, \ldots, 0$$

which, by (29), gives

$$p_1^n = E - (1 + \Delta x)\overline{S}^n \qquad \text{for } n = N, N-1, \ldots, 0 \tag{31}$$

Note that, since $\overline{S}'(t) \geq 0$, we apply a centered scheme to discretize the transport term of (19).

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

$$\frac{p_j^{n+1} - p_j^n}{\Delta t} + \frac{1}{2}\sigma^2 x_j^2 \frac{p_{j-1}^n - 2p_j^n + p_{j+1}^n}{(\Delta x)^2} +$$

$$x_j\left(r - \frac{\overline{S}^{n+1} - \overline{S}^n}{\Delta t\, \overline{S}^n}\right)\frac{p_{j+1}^n - p_{j-1}^n}{2\Delta x} - rp_j^n = 0 \tag{32}$$

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

$$\beta_j^n p_{j-1}^n + \alpha_j^n p_j^n + \gamma_j^n p_{j+1}^n = b_j \tag{33}$$

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

$$\alpha_j^n = 1 + \frac{\Delta t}{(\Delta x)^2}\sigma^2 x_j^2 + r\Delta t \tag{34}$$

$$\beta_j^n = \frac{-\Delta t}{2(\Delta x)^2}\sigma^2 x_j^2 + \frac{\Delta t}{2\Delta x}x_j\left(r - \frac{\bar{S}^{n+1} - \bar{S}^n}{\Delta t \bar{S}^n}\right) \tag{35}$$

$$\gamma_j^n = \frac{-\Delta t}{2(\Delta x)^2}\sigma^2 x_j^2 - \frac{\Delta t}{2\Delta x}x_j\left(r - \frac{\bar{S}^{n+1} - \bar{S}^n}{\Delta t \bar{S}^n}\right) \tag{36}$$

$$b_j^n = p_j^{n+1} \tag{37}$$

Putting $j = 1$ in (33) we get

$$\gamma_1^n p_2^n = b_1^n - \beta_1^n(E - \bar{S}^n) - \alpha_1^n[E - (1 + \Delta x)\bar{S}^n] \tag{38}$$

where we have used (29) and (31). Putting $j = 2$ in (33) leads to

$$\alpha_2^n p_2^n + \gamma_2^n p_3^n = b_2^n - \beta_2^n[E - (1 + \Delta x)\bar{S}^n] \tag{39}$$

By putting $j = M$ in (33) and incorporating (30) we find that

$$\beta_M^n p_{M-1}^n + \alpha_M^n p_M^n = b_M^n \tag{40}$$

Finally, for $j = 3, 4, \ldots, M - 1$, we have the equations

$$\beta_j^n p_{j-1}^n + \alpha_j^n p_j^n + \gamma_j^n p_{j+1}^n = b_j \tag{41}$$

At each time step $t_n = n\Delta t$ we now have M unknowns given by $p_2^n, p_3^n, \ldots, p_M^n$ and $\bar{S}^n$, and M equations given by (38), (39), (40) and (41). We want to write this system in a more compact form – that is, in a form more suitable for applying Newton's method. If we define the matrix $A = A(\bar{S}^n) \in \mathbb{R}^{M,M-1}$ by

$$A(\bar{S}^n) = \begin{pmatrix} \gamma_1^n & & & & & \\ \alpha_2^n & \gamma_2^n & & & & \\ \beta_3^n & \alpha_3^n & \gamma_3^n & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \beta_{M-1}^n & \alpha_{M-1}^n & \gamma_{M-1}^n \\ & & & & \beta_M^n & \alpha_M^n \end{pmatrix} \tag{42}$$

and the mapping $f = f(\bar{S}^n)$: $\mathbb{R} \to \mathbb{R}^M$ by

$$f(\bar{S}^n) = \begin{bmatrix} b_1^n - \beta_1^n(E - \bar{S}^n) - \alpha_1^n[E - (1+\Delta x)\bar{S}^n] \\ b_2^n - \beta_2^n[E - (1+\Delta x)\bar{S}^n] \\ b_3^n \\ \vdots \\ b_M^n \end{bmatrix} \tag{43}$$

the system (38)–(41) can now be written as

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

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

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

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

3.2 A centered explicit scheme

Now we want to define a centered explicit scheme for the front-fixing method discussed above. Specifically, we need a centered 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:

$$\frac{p_j^{n+1} - p_j^n}{\Delta t} + \frac{1}{2}\sigma^2 x_j^2 \frac{p_{j-1}^{n+1} - 2p_j^{n+1} + p_{j+1}^{n+1}}{(\Delta x)^2} +$$

$$x_j\left(r - \frac{\bar{S}^{n+1} - \bar{S}^n}{\Delta t\,\bar{S}^{n+1}}\right)\frac{p_{j+1}^{n+1} - p_{j-1}^{n+1}}{2\Delta x} - rp_j^{n+1} = 0 \tag{46}$$

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 as

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

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

$$A_j = \frac{1}{2}\sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} - x_j\left(r - \frac{1}{\Delta t}\right)\frac{\Delta t}{2\Delta x}$$

$$B_j = 1 - \sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} - r\Delta t$$

$$C_j = \frac{1}{2}\sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} + x_j\left(r - \frac{1}{\Delta t}\right)\frac{\Delta t}{2\Delta x}$$

$$D_j^n = \frac{x_j}{2\Delta x}\frac{p_{j+1}^n - p_{j-1}^{n+1}}{\bar{S}^n}$$

Note that the coefficient D_j^{n+1} in (47) depends only 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 $\bar{S}^n$. Putting $j = 1$ in (47), 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

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

The derivations given above lead to the following algorithm:

1. for $j = 0, 1, \ldots, M+1$, do $p_j^{N+1} = 0$;

2. $\bar{S}^{N+1} = E$;

3. for $n = N+1, N, \ldots, 0$, do $p_{M+1}^n = 0$;

4. for $j = 1, 2, \ldots, M$, do

$$A_j = \frac{1}{2}\sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} - x_j\left(r - \frac{1}{\Delta t}\right)\frac{\Delta t}{2\Delta x}$$

$$B_j = 1 - \sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} - r\Delta t$$

$$C_j = \frac{1}{2}\sigma^2 x_j^2 \frac{\Delta t}{(\Delta x)^2} + x_j\left(r - \frac{1}{\Delta t}\right)\frac{\Delta t}{2\Delta x}$$

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

a. $D_j^{n+1} = \dfrac{x_j}{2\Delta x} \dfrac{p_{j+1}^{n+1} - p_{j-1}^{n+1}}{\bar{S}^{n+1}}$

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

c. $p_0^n = E - \bar{S}^n$ and $p_1^n = E - (1 + \Delta x)\bar{S}^n$

d. for $j = 2, 3, \ldots, M$, 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:

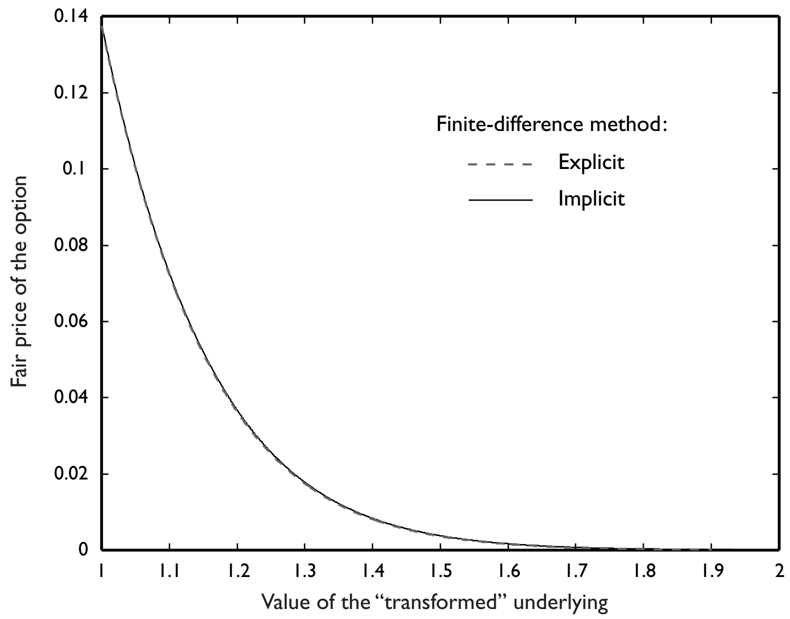
$$r = 0.1$$
$$\sigma = 0.2$$
$$E = 1$$
$$T = 1$$
$$x_\infty = 2$$

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 (Diffpack; Langtangen, 1999). 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 in Section 5.

Figure 2 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 $\Delta t = 5.0 \times 10^{-6}$ 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 $t = 0$:

$$\text{implicit: } \bar{S}^0 = 8.619 \times 10^{-1}, \qquad \text{explicit: } \bar{S}^0 = 8.623 \times 10^{-1}$$

FIGURE 2 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.

Note 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 obtained the front position

$$\overline{S}^0 = 8.616 \times 10^{-1}$$

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

4 Penalty methods

Now we turn our attention to penalty methods for solving free and moving boundary problems. To explain our approach, we present a very simple problem chosen to illuminate the key properties of the method.

4.1 An ordinary differential equation

We consider 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 to this problem can be computed analytically and is given by

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

Suppose, however, that 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:

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

where $u_0 = 2$. This corresponds to a Brennan–Schwartz type of algorithm for pricing American put options (Brennan and Schwartz, 1977).

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

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

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

$$\bar{v} = 1 \tag{53}$$

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

$$\bar{v} \le v(t) \le 2, \qquad t \ge 0 \tag{54}$$

$$v'(t) \le 0, \qquad t \ge 0 \tag{55}$$

$$v''(t) \ge 0, \qquad t \ge 0 \tag{56}$$

Note that for

$$\bar{v} \le v \le 2 \tag{57}$$

we have

$$v(v + \epsilon - 1) \ge \epsilon \tag{58}$$

Hence, by (52), we have

$$v' = \frac{1}{v+\epsilon-1}\left(\epsilon - v(v+\epsilon-1)\right) \le 0 \tag{59}$$

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

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

and thus

$$v'' \ge 0 \tag{61}$$

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 scheme

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

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

$$\bar{v} \le v_n \le 2 \qquad \text{for } n \ge 0 \tag{63}$$

$$v_{n+1} \le v_n \qquad \text{for } n \ge 0 \tag{64}$$

$$v_n \xrightarrow{n\to\infty} \bar{v} \tag{65}$$

We assume that

$$\Delta t \leq \frac{\epsilon}{1+\epsilon} \tag{66}$$

Define

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

and note that

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

We assume that

$$\bar{v} \leq v_n \leq 2 \tag{68}$$

Then

$$v_n(v_n + \epsilon - 1) \geq \epsilon \tag{69}$$

and

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

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) follow 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

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

Hence,

$$v_n \to 1 \quad \text{as} \quad n \to \infty$$

Experiments

Table 1 gives 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 converge towards the correct solution as ϵ (and consequently Δt) tends towards zero. Notice that the error is (roughly) of order ϵ.

TABLE 1 Numerical results generated by the penalty method presented in Sections 4.1 and 4.2.

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

5. A penalty method for Black–Scholes models

In this section we modify the analysis presented above so 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 equation (1). This gives 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

$$\frac{\partial V_\epsilon}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V_\epsilon}{\partial S^2} + rS\frac{\partial V_\epsilon}{\partial S} - rV_\epsilon + \frac{\epsilon C}{V_\epsilon + \epsilon - q(S)} = 0, \qquad S \geq 0,\ t \in [0, t) \tag{71}$$

$$V_\epsilon(S, T) = \max(E - S, 0) \tag{72}$$

$$V_\epsilon(0, t) = E \tag{73}$$

$$V_\epsilon(S, t) = 0 \qquad \text{as } S \to \infty \tag{74}$$

where $C \geq rE$ is a positive constant[3] and

$$q(S) = E - S \tag{75}$$

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 $V_\epsilon \to q(S)$.

Our goal is to define numerical methods for solving (71)–(74) 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 (71)–(74) on a limited interval for the value S of the underlying asset. Thus, we introduce a parameter S_∞ (preferably a large number) that represents the endpoint of discretization with respect to S. More precisely, we will discretize (71)–(74) in the domain

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

Let, for given positive integers M and N,

$$\Delta S = \frac{S_\infty}{M+1}, \qquad \Delta t = \frac{T}{N+1}$$

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

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 (71)–(74), ie,

$$\frac{V_j^n - V_j^{n-1}}{\Delta t} + \frac{1}{2}\sigma^2 S_j^2 \frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{(\Delta S)^2} + rS_j \frac{V_{j+1}^n - V_j^n}{\Delta S} - rV_j^n + \frac{\epsilon C}{V_j^n + \epsilon - q_j} = 0 \tag{76}$$

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

$$\begin{aligned}
f(V_-, V, V_+, q, S) = {} & \left[\alpha\sigma^2 S^2\right] V_- + \left[1 - 2\alpha\sigma^2 S^2 - \frac{\Delta t}{\Delta S} rS - r\Delta t\right] V \\
& + \left[\alpha\sigma^2 S^2 - \frac{\Delta t}{\Delta S} rS\right] V_+ + \frac{\epsilon C \Delta t}{V + \epsilon - q}
\end{aligned} \tag{77}$$

where

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

then (76) can be written as

$$V_j^{n-1} = f(V_{j-1}^n, V_j^n, V_{j+1}^n, q_j, S_j), \qquad j = 1, \dots, M \text{ and } n = N+1, \dots, 1 \tag{78}$$

With this notation at hand we are ready to start analyzing our explicit scheme. We begin 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,

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

provided that Δt satisfies

$$\Delta t \leq \frac{(\Delta S)^2}{\sigma^2 S_\infty^2 + rS_\infty(\Delta S) + r(\Delta S)^2 + \frac{C}{\epsilon}(\Delta S)^2} \tag{79}$$

PROOF Clearly,

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

for all S, $r \geq 0$.

Next,

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

and for $V \geq q$ it follows that

$$\frac{\partial f}{\partial V} \geq 1 - \frac{\Delta t}{(\Delta S)^2}\sigma^2 S^2 - \frac{\Delta t}{\Delta S} rS - r\Delta t - \frac{\epsilon C \Delta t}{\epsilon^2}$$

$$= 1 - \Delta t\left[\frac{\sigma^2 S^2}{(\Delta S)^2} + \frac{rS}{\Delta S} + r + \frac{C}{\epsilon}\right] \geq 0$$

provided that Δt satisfies (79). □

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

THEOREM 2 *For all $C \geq rE$, $S_\infty \geq E$ and all Δt satisfying (79), the approximate option values $\{V_j^n\}$ generated by the scheme (78) satisfy*

$$V_j^n \geq \max(E - S_j, 0) \tag{80}$$

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

PROOF By definition, $V_j^{N+1} = \max(E - S_j, 0)$ for $j = 0, \ldots, M+1$, and hence (80) 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 $n = N+1, \ldots, 0$, provided that $S_\infty \geq E$.

Next, we prove that if (80) holds for n, it must also be valid for $n - 1$. Let $q(S)$ be the function defined in (75) and notice that

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

If (80) holds for n, it follows from definition (78) of our scheme and Lemma 1 that

$$V_j^{n-1} = f(V_{j-1}^n, V_j^n, V_{j+1}^n, q_j, S_j) \geq f(q_{j-1}, q_j, q_{j+1}, q_j, S_j) \tag{81}$$

Notice that, compared with the definition (75) of q,

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

Next, from the definition (77) of f we find that

$$V_j^{n-1} \geq f\left(q_{j-1}, q_j, q_{j+1}, q_j, S_j\right) = q_j - r\Delta t\, q_j - \frac{\Delta t}{\Delta S} rS_j \Delta S + \frac{\epsilon C \Delta t}{q_j + \epsilon - q_j}$$

$$= q_j - r\Delta t\, q_j - r\Delta t\, S_j + C\Delta t$$

Recall that $q_j = E - S_j$ (compare with (75)), and, therefore,

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

provided that $C \geq rE$.

If (79) holds and $\{V_j^n\}$ satisfies (80), it follows directly from (78) and (77) 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, which leads to unacceptable computing times. This is particularly so in the case of multi-asset options (compare Nielsen, Skavhaug and Tveito, in prep.). In this section we derive an implicit and a semi-implicit scheme. For both schemes we assume that

$$C \geq rE \tag{82}$$

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

$$\Delta t \leq \frac{\epsilon}{rE} \tag{83}$$

is satisfied. Note that this condition is a significantly milder restriction than (79). To avoid having to solve non-linear algebraic systems, we find the use of a semi-implicit scheme an attractive alternative.

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

$$\begin{aligned} &\frac{V_j^{n+1} - V_j^n}{\Delta t} + \frac{1}{2}\sigma^2 S_j^2 \frac{V_{j-1}^n - 2V_j^n + V_{j+1}^n}{(\Delta S)^2} \\ &\quad + rS_j \frac{V_{j+1}^n - V_j^n}{\Delta S} - rV_j^n + \frac{\epsilon C}{V_j^{n+1/2} + \epsilon - q_j} = 0 \end{aligned} \tag{84}$$

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 (84) can be rearranged as

$$\left(1 + r\Delta t + 2\alpha_j + \beta_j\right) V_j^n = V_j^{n+1} + \left(\alpha_j + \beta_j\right) V_{j+1}^n + \alpha_j V_{j-1}^n + \frac{\epsilon \Delta t C}{V_j^{n+1/2} + \epsilon - q_j} \tag{85}$$

where

$$\alpha_j = \frac{1}{2}\frac{\Delta t}{(\Delta S)^2} S_j^2 \sigma^2, \qquad \beta_j = r\frac{\Delta t}{\Delta S} S_j \tag{86}$$

Our aim is to show that

$$V_j^n \geq \max(q_j, 0), \qquad \forall\, j, n \tag{87}$$

We do this in two steps; first, we show that

$$V_j^n \geq q_j, \qquad \forall\, j \tag{88}$$

and next that

$$V_j^n \geq 0 \tag{89}$$

In order to prove (88), we introduce

$$u_j^n = V_j^n - q_j \tag{90}$$

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

$$\left(1 + r\Delta t + 2\alpha_j + \beta_j\right) u_j^n = u_j^{n+1} + \left(\alpha_j + \beta_j\right) u_{j+1}^n + \alpha_j u_{j-1}^n + \frac{\epsilon \Delta t\, C}{u_j^{n+1/2} + \epsilon} - r\Delta t\, E \tag{91}$$

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

$$u^n = \min_j u_j^n \tag{92}$$

and let k be an index such that

$$u_k^n = u^n \tag{93}$$

For $j = k$, it follows from (91) that

$$(1 + r\Delta t + 2\alpha_k + \beta_k)\, u^n \geq u_k^{n+1} + (\alpha_k + \beta_k)\, u^n + \alpha_k u^n + \frac{\epsilon \Delta t\, C}{u_k^{n+1/2} + \epsilon} - r\Delta t\, E \tag{94}$$

or

$$(1 + r\Delta t)\, u^n \geq u_k^{n+1} \frac{\epsilon \Delta t\, C}{u_k^{n+1/2} + \epsilon} - r\Delta t\, E \tag{95}$$

Let us now consider the fully implicit case. Here (95) takes the form

$$(1 + r\Delta t)\, u^n - \frac{\epsilon \Delta t\, C}{u_k^n + \epsilon} + r\Delta t\, E \geq u_k^{n+1} \geq u^{n+1} \tag{96}$$

If we assume that

$$u^{n+1} \geq 0 \tag{97}$$

we have

$$F(u^n) \geq 0 \tag{98}$$

where

$$F(u) = (1 + r\Delta t)\,u - \frac{\epsilon \Delta t\, C}{u + \epsilon} + r\Delta t\, E \tag{99}$$

Since

$$F(0) = \Delta t(rE - C) \leq 0 \tag{100}$$

(compare (82)), and

$$F'(u) = 1 + r\Delta t + \frac{\epsilon \Delta t\, C}{(u + \epsilon)^2} > 0 \tag{101}$$

it follows from (98) that

$$u^n \geq 0 \tag{102}$$

Consequently, by induction on n, it follows from (90) that

$$V_j^n \geq q_j, \qquad \forall\, j \tag{103}$$

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

Next we consider the semi-implicit scheme and we assume that (83) holds. It follows from (95) that

$$(1 + r\Delta t)u^n \geq \frac{u_k^{n+1}\left(u_k^{n+1} + \epsilon\right) + \epsilon \Delta t\, C - r\Delta t\, E\left(u_k^{n+1} + \epsilon\right)}{u_k^{n+1} + \epsilon} \tag{104}$$

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

$$G(u) = u(u + \epsilon) + \epsilon \Delta t\, C - r\Delta t\, E(u + \epsilon) \tag{105}$$

Then

$$G(0) = \Delta t\, \epsilon (C - Er) \geq 0 \tag{106}$$

(compare (82)), and

$$G'(u) = 2u + \epsilon - r\Delta t\, E \tag{107}$$

so $G'(u) \geq 0$ for $u \geq 0$ provided that (83) holds. Hence, we have

$$u_j^n \geq 0 \tag{108}$$

and thus by (90)

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

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

Next we consider (89), ie, we want to show that

$$V_j^n \geq 0 \tag{109}$$

As above, we define

$$V^n = \min_j V_j^n \tag{110}$$

and let k be an index such that

$$V_k^n = V^n \tag{111}$$

It follows from (85) that

$$(1 + r\Delta t + 2\alpha_k + \beta_k) V^n \geq V^{n+1} + (\alpha_k + \beta_k) V^n + \alpha_k V^n + \frac{\epsilon \Delta t C}{V_k^{n+1/2} + \epsilon - q_k} \tag{112}$$

or

$$(1 + r\Delta t) V^n \geq V^{n+1} + \frac{\epsilon \Delta t C}{V_k^{n+1/2} + \epsilon - q_k} \tag{113}$$

Since we have just seen that

$$V_k^{n+1/2} \geq q_k \tag{114}$$

both in the fully implicit and in the semi-implicit case, it follows from (113) that

$$(1 + r\Delta t) V^n \geq V^{n+1} \tag{115}$$

and then it follows by induction on n that

$$V_j^n \geq 0, \qquad \forall\, j \tag{116}$$

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

THEOREM 3 *If (82) holds, the numerical solution computed by the fully implicit scheme (84) satisfies the bound*

$$V_j^n \geq \max(E - S_j, 0), \qquad \forall\, j \tag{117}$$

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

Similarly, if (82) and (83) hold, the numerical solution computed by the semi-implicit version of (84) satisfies the bound (117).

5.3 Numerical experiments

To provide further insight we will now 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, ie, that the discrete analogue to (8) holds.

In order to test the convergence properties of the penalty schemes, we will consider discrete L_1, L_2 and L_∞ norms. More precisely, for a discrete function, g, defined on the mesh

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

we define the norms

$$\| g \|_1 = \Delta x \left[\frac{|g_0| + |g_{M+1}|}{2} + \sum_{j=1}^{M} |g_j| \right] \tag{118}$$

$$\| g \|_2 = \left(\Delta x \left[\frac{(g_0)^2 + (g_{M+1})^2}{2} + \sum_{j=1}^{M} (g_j)^2 \right] \right)^{1/2} \tag{119}$$

$$\| g \|_\infty = \max_j |g_j| \tag{120}$$

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 (see, eg, Wilmott, Dewynne and Howison, 1993). Hence, the convergence properties of the discrete derivatives, implicitly defined by our methods, are also of interest. We will therefore 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 the L_1 sense:

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

$$| g(\cdot, t_n) |_1 = \sum_{j=1}^{M+1} \frac{|g_j^n - g_{j-1}^n|}{\Delta x} \Delta x = \sum_{j=1}^{M+1} |g_j^n - g_{j-1}^n|$$

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 $C = rE$ (compare with Theorems 2 and 3). For a decreasing sequence of ϵ values $\epsilon_0, \epsilon_1, \ldots, \epsilon_I$,

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

we compare the approximate option values generated by (78) (and the approximate option values generated by the implicit and semi-implicit versions of (84)) 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 (78) (and the error associated with the implicit and semi-implicit versions of (84)):

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

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

In the accompanying tables 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, ie, we compute

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

for each value of ϵ_i.

Case I

As already mentioned, 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 (Langtangen, 1999).

The results reported in Tables 2, 3 and 4 were computed as follows. In Table 2 we used the upwind explicit finite-difference scheme (76) with discretization parameters $\Delta S = 1.0 \times 10^{-3}$ and Δt computed according to (79). Table 3 presents the results generated by the fully implicit version of scheme (84). Here we applied the discretization parameters $\Delta S = \Delta t = 1.0 \times 10^{-3}$. Finally, Table 4 gives the results generated by the semi-implicit version of the finite-difference scheme (84). In this case we applied the discretization parameters $\Delta S = 1.0 \times 10^{-3}$ and $\Delta t = 5.0 \times 10^{-4}$. Hence, condition (83) 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 appears that the estimated option values provided by the penalty schemes converge towards the reference solution as $\epsilon \to 0$. This is the case not

TABLE 2 The penalty method applied to the American put problem (explicit time stepping, ϕ = 0.0 for all values of ϵ).

ϵ	L_1	L_2	L_∞	H_1	$\lvert\cdot\rvert_1$	CPU time (s)
10^{-1}	2.51×10^{-2}	2.63×10^{-2}	4.26×10^{-2}	1.07×10^{-1}	9.49×10^{-3}	129.5
10^{-2}	5.14×10^{-3}	6.43×10^{-3}	1.34×10^{-2}	4.09×10^{-2}	1.64×10^{-3}	129.5
10^{-3}	7.23×10^{-4}	1.06×10^{-3}	2.65×10^{-3}	1.14×10^{-2}	1.29×10^{-4}	129.6
10^{-4}	7.04×10^{-5}	1.17×10^{-4}	3.84×10^{-4}	2.53×10^{-3}	6.47×10^{-6}	130.2

TABLE 3 The penalty method applied to the American put problem (implicit time stepping, ϕ = 0.0 for all values of ϵ).

ϵ	L_1	L_2	L_∞	H_1	$\lvert\cdot\rvert_1$	CPU time (s)
10^{-1}	2.58×10^{-2}	2.37×10^{-2}	3.78×10^{-2}	2.19×10^{-1}	4.73×10^{-2}	7.8
10^{-2}	5.21×10^{-3}	5.75×10^{-3}	1.20×10^{-2}	5.01×10^{-2}	2.48×10^{-3}	7.8
10^{-3}	7.16×10^{-4}	9.42×10^{-4}	2.36×10^{-3}	1.11×10^{-2}	1.23×10^{-4}	7.8
10^{-4}	7.40×10^{-5}	1.03×10^{-4}	3.37×10^{-4}	2.37×10^{-3}	5.68×10^{-6}	8.4

TABLE 4 The penalty method applied to the American put problem (semi-implicit time stepping, ϕ = 0.0 for all values of ϵ).

ϵ	L_1	L_2	L_∞	H_1	$\lvert\cdot\rvert_1$	CPU time (s)
10^{-1}	2.51×10^{-2}	2.63×10^{-2}	4.26×10^{-2}	1.01×10^{-1}	9.49×10^{-3}	2.8
10^{-2}	5.14×10^{-3}	6.43×10^{-3}	1.34×10^{-2}	4.09×10^{-2}	1.64×10^{-3}	2.8
10^{-3}	7.23×10^{-4}	1.06×10^{-3}	2.64×10^{-3}	1.13×10^{-2}	1.29×10^{-4}	2.8
10^{-4}	7.01×10^{-5}	1.17×10^{-4}	3.83×10^{-4}	2.53×10^{-3}	6.46×10^{-6}	2.8

only for the L_1, L_2, L_∞ norms but also for the H^1 norm and the $|\cdot|_1$ semi-norm, which take the discrete derivatives into consideration. Furthermore, the constraint imposed by the possibility of early exercise holds, ie, $\phi = 0$ in all three tables. (Notice that, due to the final condition $V_{\epsilon_{i,j}}^{N+1} = q_j$, ϕ will always satisfy $\phi \le 0$. Consequently, if the conditions of Theorems 2 and 3 hold, ϕ must be zero.)

Note that the computational efficiency of the schemes differs significantly. Due to the severe restriction (79) 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 three to four Newton iterations on average.

Case II

For the upwind explicit finite-difference method, presented in Section 5.1, we showed that if (79) holds, the scheme satisfies a discrete analogue to the early exercise constraint (8) (compare with 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 $\phi = -3.93 \times 10^6$.

The restriction (83) on Δt for the semi-implicit scheme is milder. However, by choosing $\Delta t = 1.0 \times 10^{-3}$ and $\epsilon = 5.0 \times 10^{-5}$ – and, hence, violating (83) – we observed that $\phi = -2.72 \times 10^{-8}$. This leads to unacceptable results.

Recall that we proved Theorems 2 and 3 assuming that the constant C in the penalty term is larger than 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 resulted in approximate option values that do not satisfy the lower bound (117). The scheme became unstable and $\phi \approx -1.0 \times 10^{-6}$.

Finally, it should be mentioned that all computations reported in this paper were performed on a dual Dell Workstation 410 with two PIII 600 MHz microprocessors and 1 Gb RAM running the Linux operating system.

1. The case of continuous dividends clearly falls within the framework of our approach, whereas problems involving discrete dividends are not analysed in this paper.

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

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

REFERENCES

Amin, K., and Khanna, A. (1994). Convergence of American option values from discrete to continuous-time financial models. *Mathematical Finance* **4,** 289–304.

Barraquand, J., and Pudet, T. (1994). Pricing of American path-dependent contingent claims. Digital Research Laboratory, Paris.

Brennan, M., and Schwartz, E. (1977). The valuation of American put options. *Journal of Finance* **32,** 449–62.

Broadie, M., and Detemple, J. (1996). American option valuation: New bounds, approximations, and a comparison of existing methods. *Review of Financial Studies* **9**(4), 1211–50.

Crank, J. (1984). *Free and Moving Boundary Problems.* Oxford University Press.

Diffpack. http://www.nobjects.no/.

Duffie, D. (1996). *Dynamic Asset Pricing Theory.* Princeton University Press, New Jersey.

Hull, J. C. (1997). *Options, Futures and other Derivatives.* Prentice Hall, New Jersey.

Jaillet, P., Lamberton, D., and Lapeyre, B. (1990). Variational inequalities and the pricing of American options. *ACTA Applicandae Mathematicae* **21** (3), 263–89.

Kwok, Y. K. (1998). *Mathematical Models of Financial Derivatives.* Springer-Verlag, Heidelberg.

Langtangen, H. P. (1999). *Computational Partial Differential Equations. Numerical Methods and Diffpack Programming.* Springer-Verlag, Heidelberg.

Nielsen, B. F., Skavhaug, O., and Tveito, A. Penalty methods for the numerical solution of American multi-asset option problems. Technical report, Department of Informatics, University of Oslo, in preparation.

Ross, S. H. (1999). *An Introduction to Mathematical Finance.* Cambridge University Press.

Wilmott, P. (1998). *Derivatives, The Theory and Practice of Financial Engineering.* John Wiley & Sons, New York.

Wilmott, P., Dewynne, J., and Howison, S. (1993). *Option Pricing, Mathematical Models and Computation.* Oxford Financial Press.

Zhu, Y. I., and Abifaker, A. (1999). Applications of the singularity-separating method to double moving barrier options with rebates. Technical report, University of North Carolina at Charlotte.

Zhu, Y. I., Ren, H., and Xu, H. (1997). Improved effectiveness evaluating American options by the singularity-separating method. Technical report, University of North Carolina at Charlotte.

Zhu, Y. I., and Sun, Y. (1999). The singularity-separating method for two-factor convertible bonds. *Journal of Computational Finance* **3** (1), 91–110.

Zvan, R., Forsyth, P. A., and Vetzal, K. R. (1998). Penalty methods for American options with stochastic volatility. *Journal of Computational and Applied Mathematics* **91,** 199–218.