QUADRATIC CONVERGENCE OF A PENALTY METHOD FOR VALUING AMERICAN OPTIONS*

P.A. FORSYTH† AND K.R. VETZAL‡

Abstract. The convergence of a penalty method for solving the discrete regularized American option valuation problem is studied. Sufficient conditions are derived which both guarantee convergence of the nonlinear penalty iteration and ensure that the iterates converge monotonically to the solution. These conditions also ensure that the solution of the penalty problem is an approximate solution to the discrete linear complementarity problem. The efficiency and quality of solutions obtained using the implicit penalty method are compared with those produced with the commonly used technique of handling the American constraint explicitly. Convergence rates are studied as the timestep and mesh size tend to zero. It is observed that an implicit treatment of the American constraint does not converge quadratically (as the timestep is reduced) if constant timesteps are used. A timestep selector is suggested which restores quadratic convergence.

Key words. American option, penalty iteration, linear complementarity

AMS subject classifications. 65M12, 65M60, 91B28

1. Introduction. The valuation and hedging of financial option contracts is a subject of considerable practical significance. The holders of such contracts have the right to undertake certain actions so as to receive certain payoffs. The valuation problem consists of determining a fair price to charge for granting these rights. A related issue, perhaps of even more importance to practitioners, is how to hedge the risk exposures which arise from selling these contracts. An important feature of such contracts is the time when contract holders can exercise their rights. If this occurs only at the maturity date of the contract, the option is classified as “European”. If holders can exercise any time up to and including the maturity date, the option is said to be “American”. The value of a European option is given by the solution of the Black-Scholes PDE (see, e.g. [20]). An analytical solution can be obtained for cases with constant coefficients and simple payoffs. However, most options traded on exchanges are American. Such options must be priced numerically, even for constant coefficients and simple payoffs. Note also that the derivatives of the solution are of interest since they are used in hedging. More formally, the American option pricing problem can be posed as a differential linear complementarity problem (LCP).

In current practice, the most common method of handling the early exercise condition is simply to advance the discrete solution over a timestep ignoring the constraint, and then to apply the constraint explicitly. This has the disadvantage that the solution is in an inconsistent state at the beginning of each timestep (i.e. a discrete form of the LCP is not approximately satisfied). As well, this approach can obviously only be first order correct in time.

Another technique is to solve the discrete LCP using a relaxation method [20]. In terms of complexity, this method is particularly poor for pricing problems with one space-like dimension. A lower bound for the number of iterations required to solve the

*This work was supported by the Natural Sciences and Engineering Research Council of Canada, the Social Sciences and Humanities Research Council of Canada, the Royal Bank of Canada, and the Financial Industry Solutions Center (FISC), an SGI/Cornell University Joint Venture. In addition, Rational Software provided access to the Purify software development tool under the SEED program.

†Department of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1, paforsyt@elora.math.uwaterloo.ca

‡School of Accountancy, University of Waterloo, kvetzal@watarts.uwaterloo.ca

LCP to a given tolerance with a relaxation method would be the number of iterations required to solve the unconstrained problem using a preconditioned conjugate gradient method. Assuming that the mesh spacing in the asset price S direction is $O(\Delta S)$ and that the timestep size is $O(\Delta t)$, then the condition number of a discrete form of the parabolic option pricing PDE is $O\left[\Delta t/(\Delta S)^2\right]$. Let N be the number of timesteps. If we assume that $\Delta S = O(\Delta t) = 1/N$, then the number of iterations required per timestep would be $O(N^{1/2})$.

To overcome the problems with relaxation techniques, methods based on linear programming have been suggested [7, 11]. However, the computational complexity of this approach increases rapidly for problems having more than one space-like dimension.

A multigrid method has been suggested in [4] to accelerate convergence of the basic relaxation method. Although this is a promising technique, multigrid methods are usually strongly coupled to the type of discretization used, and hence are complex to implement in general purpose software.

It is well known that an LCP (or equivalently, a variational inequality) can be posed as a penalty method [8, 18, 15, 9, 6]. An advantage of this approach is that standard methods can be used to solve the resulting nonlinear algebraic equations. In this article, we will explore some aspects of using penalty methods for pricing American options. We will restrict attention to one dimensional problems, which are more amenable to analysis. However, we have successfully used penalty methods for two dimensional problems [22]. In this work, the nonlinear discrete penalized equations are solved using Newton iteration. This is closely related to the Newton methods in [5]. Note that relaxation methods are frequently used to solve the discrete penalized nonlinear equations [6].

The advantage of the penalty method is that a single technique can be used for one dimensional or multi-dimensional problems, and standard sparse matrix software can be used to solve the Jacobian matrix. This technique can be used for any type of discretization, in any dimension, and on unstructured meshes. In particular, there is no difficulty in handling cases where the early exercise region is multiply-connected, as in [22]. As well, a single method can be used to handle American options and other nonlinearities, such as uncertain volatility and transaction cost models [20, 1].

The objective of this article is to analyze the properties of penalty methods for solution of a discrete form of the LCP. We will also study the convergence of these methods as the timestep and mesh size are reduced to zero. We will determine sufficient conditions for smooth convergence of the penalty method in one dimension. The conditions require that certain properties of the discretization hold, and there are limitations on the timestep size. It may be possible to require weaker conditions perhaps using the methods in [15]. However, option pricing problems are typically degenerate parabolic, and in non-conservative form. This can be expected to complicate the methods in [15].

In practice, we observe that the penalty method works well for multi-factor options, with no timestep restrictions. In other words, although the conditions we derive are sufficient, they do not appear to be necessary. Consequently, it appears that the penalty method can be used for more general situations. In addition, we will compare the penalty method (where the LCP is approximately solved at each timestep) with an explicit technique for handling the American constraint.

If constant timesteps are used, we observe that second order convergence is not obtained as the timesteps and mesh size tend to zero. This phenomenon can be explained by examining the asymptotic behavior of the solution near the exercise boundary. A timestep selector is developed which restores second order convergence.

Asymptotically, the second order method is superior to the commonly used binomial lattice technique [12]. However, it is of practical interest to determine at what levels of accuracy a second order PDE method will be computationally more efficient than the lattice method. We present numerical comparisons to assist in this determination.

2. Formulation. Consider an asset with price S which follows the stochastic process

$$dS = \mu S dt + \sigma S dz \tag{2.1}$$

where μ is the drift rate, σ is volatility, and dz is the increment of a Wiener process. We wish to determine the value $V(S,t)$ of an American option where the holder can exercise at any time and receive the payoff $V^*(S,t)$. Denote the expiry time of the option by T, and let $\tau = T - t$. Then the American pricing problem can be formally stated as an LCP [20]

$$\begin{aligned} \mathcal{L}V &\geq 0 \\ (V - V^*) &\geq 0 \\ (\mathcal{L}V = 0) &\wedge (V - V^* = 0) \end{aligned} \tag{2.2}$$

where

$$\mathcal{L}V \equiv V_\tau - \left(\frac{\sigma^2}{2} S^2 V_{SS} + rSV_S - rV \right) \tag{2.3}$$

and r is the risk free rate of interest. A put option is a contract which gives the holder the right to sell the asset for K (known as the "strike"). A call option is similar except that the holder has the right to buy the asset for K. The payoff for a put is

$$V^*(S) = V(S, \tau = 0) = \max(K - S, 0). \tag{2.4}$$

The boundary conditions are

$$V(S,\tau) = 0 \quad ; \qquad S \to \infty, \tag{2.5}$$
$$\mathcal{L}V = V_\tau - rV \quad ; \qquad S \to 0. \tag{2.6}$$

Condition (2.5) follows from the payoff (2.4), while (2.6) is obvious given (2.3).

3. The Penalty Method. The basic idea of the penalty method is simple: we replace problem (2.2) by the nonlinear PDE

$$V_\tau = \frac{\sigma^2}{2} S^2 V_{SS} + rSV_S - rV + Q(V, V^*), \tag{3.1}$$

where the penalty term $Q(V, V^*)$ equals zero if $V \geq V^*$ and is positive if $V < V^*$. Intuitively, we can see how this would work. If $Q(V, V^*) = 0$, then we have

$$\begin{aligned} V &\geq V^*, \\ V_\tau &= \frac{\sigma^2}{2} S^2 V_{SS} + rSV_S - rV. \end{aligned} \tag{3.2}$$

If instead $Q(V, V^*) > 0$, then $Q(V, V^*)$ is defined so that

$$|V - V^*| < \epsilon \quad ; \qquad \epsilon \ll 1$$

$$V_\tau - \left(\frac{\sigma^2}{2} S^2 V_{SS} + rSV_S - rV \right) = Q(V, V^*) > 0. \tag{3.3}$$

4. Discretization. We will now discretize equation (3.1) and select a suitable form for the discrete penalty term. Let $V(S_i, \tau_n) = V_i^n$ be the discrete solution to equation (3.1) at asset value S_i, and time (going backwards) τ_n. Applying a standard finite volume method with variable timeweighting [22] then gives

$$\mathcal{F} V_i^{n+1} = q_i^{n+1}, \tag{4.1}$$

where

$$\begin{aligned} \mathcal{F} V_i^{n+1} \equiv & A_i \left(\frac{V_i^{n+1} - V_i^n}{\Delta \tau} \right) \\ & + (1-\theta) \left(\sum_{j \in \eta_i} \gamma_{ij} (V_j^{n+1} - V_i^{n+1}) + \sum_{j \in \eta_i} \vec{L_{ij}} \cdot \mathbf{U}_i V_{ij+1/2}^{n+1} - A_i r V_i^{n+1} \right) \\ & + \theta \left(\sum_{j \in \eta_i} \gamma_{ij} (V_j^n - V_i^n) + \sum_{j \in \eta_i} \vec{L_{ij}} \cdot \mathbf{U}_i V_{ij+1/2}^{n} - A_i r V_i^n \right). \end{aligned} \tag{4.2}$$

Fully implicit and Crank-Nicolson discretizations correspond to cases of $\theta = 0$ and $\theta = 1/2$ respectively, and

$$\begin{aligned} A_i &= (S_{i+1} - S_{i-1})/2 \\ \eta_i &= \{i-1, i+1\} \\ \Delta\tau &= \tau^{n+1} - \tau^n \\ \gamma_{ij} &= \frac{\sigma^2 S_i^2}{2|S_j - S_i|} \\ V_{ij+1/2}^{n+1} &= \text{ value of } V \text{ at the face between nodes } i \text{ and } j \\ \mathbf{U}_i &= (-rS_i)\hat{\mathbf{i}} \\ \vec{L_{ij}} &= \begin{cases} -\hat{\mathbf{i}} & \text{if } j = i+1 \\ +\hat{\mathbf{i}} & \text{if } j = i-1 \end{cases} \\ \hat{\mathbf{i}} &= \text{ unit vector in the positive } S \text{ direction.} \end{aligned} \tag{4.3}$$

The discrete penalty term q_i^{n+1} in equation (4.1) is given by

$$q_i^{n+1} = \begin{cases} (A_i/\Delta\tau)(V_i^* - V_i^{n+1}) Large & \text{if } V_i^{n+1} < V_i^* \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

where *Large* is the penalty factor (this will be related to the desired convergence tolerance below in §4.1). The face value $V_{ij+1/2}^{n+1}$ can be evaluated using either central weighting or, to ensure non-oscillatory solutions, a flux limiter [23]

$$V_{ij+1/2}^{n+1} = \begin{cases} (V_i^{n+1} + V_j^{n+1})/2 & \text{if } \gamma_{ij}^{n+1} + \vec{L_{ij}} \cdot \mathbf{U}_i/2 > 0 \\ \text{second order flux limiter [23]} & \text{otherwise.} \end{cases} \tag{4.5}$$

In general, for standard options with typical values for σ, r, central weighting can be used at most nodes (except perhaps as $S \to 0$). In order to determine sufficient conditions for the convergence of the nonlinear iteration for the penalized American equation, we require that the coefficients of the discrete equations have certain properties. We will ensure that these conditions are satisfied by using central or upstream weighting. (In practice, we have observed that even if these conditions are not met, convergence of the penalty method is still rapid). If we use central or upstream weighting in the following, then equation (4.1) becomes

$$\begin{aligned} V_i^{n+1} - V_i^n = {} & (1-\theta)\left(\Delta\tau \sum_{j\in\eta_i} \left(\bar{\gamma}_{ij} + \bar{\beta}_{ij}\right)(V_j^{n+1} - V_i^{n+1}) - r\Delta\tau V_i^{n+1}\right) \\ & + \theta\left(\Delta\tau \sum_{j\in\eta_i} \left(\bar{\gamma}_{ij} + \bar{\beta}_{ij}\right)(V_j^{n} - V_i^{n}) - r\Delta\tau V_i^{n}\right) \\ & + P_i^{n+1}(V_i^* - V_i^{n+1}), \end{aligned} \tag{4.6}$$

where

$$P_i^{n+1} = \begin{cases} Large & \text{if } V_i^{n+1} < V_i^* \\ 0 & \text{otherwise,} \end{cases} \tag{4.7}$$

and where

$$\begin{aligned} \bar{\gamma}_{ij} &= \frac{\sigma^2 S_i^2}{2A_i|S_j - S_i|} \\ \bar{\beta}_{ij} &= \begin{cases} \vec{L_{ij}} \cdot \mathbf{U}_i/2A_i & \text{if } \gamma_{ij} + \vec{L_{ij}} \cdot \mathbf{U}_i/2 \geq 0 \\ \max(\vec{L_{ij}} \cdot \mathbf{U}, 0)/A_i & \text{otherwise.} \end{cases} \end{aligned}$$

For future reference, we can write the discrete equations (4.6) in matrix form. Let $V^{n+1} = \left[V_0^{n+1}, V_1^{n+1}, \ldots, V_m^{n+1}\right]'$, $V^n = [V_0^n, V_1^n, \ldots, V_m^n]'$, $V^* = [V_0^*, V_1^*, \ldots, V_m^*]'$, and

$$[\hat{M}V^n]_i = -\left(\Delta\tau \sum_{j\in\eta_i} \left(\bar{\gamma}_{ij} + \bar{\beta}_{ij}\right)(V_j^n - V_i^n) - r\Delta\tau V_i^n\right). \tag{4.8}$$

Note that the first and last rows of $\hat{M}$ will have to be modified to take into account the boundary conditions. (An obvious method for applying conditions (2.5-2.6) results in the first and last rows of $\hat{M}$ being identically zero except for positive entries on the diagonal.) In the following, we will assume that upstream and central weighting are selected so that $\bar{\gamma}_{ij} + \bar{\beta}_{ij} \geq 0$. This implies that the matrix $\hat{M}$ is an M-matrix, i.e. a diagonally dominant matrix with positive diagonals and non-positive off-diagonals. Note that all of the elements of the inverse of an M-matrix are non-negative.

Let the diagonal matrix $\bar{P}$ be given by

$$\bar{P}(V^{n+1})_{ij} = \begin{cases} Large & \text{if } V_i^{n+1} < V_i^* \text{ and } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

We can then write the discrete equations (4.6) as

$$\left[I + (1-\theta)\hat{M} + \bar{P}(V^{n+1})\right] V^{n+1} = \left[I - \theta\hat{M}\right] V^n + \left[\bar{P}(V^{n+1})\right] V^*. \tag{4.10}$$

4.1. Solution of the Discrete LCP. The discrete form of the LCP (2.2) can be written as

$$\begin{aligned} \mathcal{F}V_i^{n+1} &\geq 0 \\ V_i^{n+1} - V_i^* &\geq 0 \\ (\mathcal{F}V_i^{n+1} = 0) \wedge (V_i^{n+1} &- V_i^* = 0), \end{aligned} \tag{4.11}$$

where $\mathcal{F}$ is given by equation (4.2). On the other hand, the discrete solution of the penalized problem (4.10) has the property that either

$$V_i^{n+1} - V_i^* \geq 0 \qquad (\Rightarrow q_i^{n+1} = 0 \text{ and } \mathcal{F}V_i^{n+1} = 0), \tag{4.12}$$

or

$$V_i^{n+1} - V_i^* \leq 0 \qquad (\Rightarrow q_i^{n+1} > 0 \text{ and } \mathcal{F}V_i^{n+1} > 0). \tag{4.13}$$

However, from equation (4.11) the exact solution of the discrete LCP has $V_i^{n+1} - V_i^* = 0$ at those nodes where $\mathcal{F}V_i^{n+1} > 0$. In order to obtain an approximate solution of (4.11) with an arbitrary level of precision, we need to show that the solution of (4.10) satisfies $V_i^{n+1} - V_i^* \to 0$ as *Large* $\to \infty$ for nodes where $\mathcal{F}V_i^{n+1} > 0$. This follows if we can show that the term

$$P_i^{n+1}(V_i^* - V_i^{n+1}) \tag{4.14}$$

in equation (4.6) is bounded independent of *Large*. It is also desirable that the bound be independent of the timestep and mesh spacing, so that *Large* can be chosen without regard to grid and timestep size. In Appendix A we determine sufficient conditions which allow us to bound (4.14). The results can be summarized as:

THEOREM 4.1 (Error in the penalty formulation of the discrete LCP). *Under the assumptions that the matrix $\hat{M}$ in equation* (4.10) *is an M-matrix and that timesteps are selected so that conditions* (*A*.9) *and* (*A*.10) *are satisfied, the penalty method* (4.10) *solves*

$$\mathcal{F}V_i^{n+1} \geq 0 \tag{4.15}$$

$$V_i^{n+1} - V_i^* \geq -\frac{C}{Large} \quad ; \qquad C > 0 \tag{4.16}$$

$$(\mathcal{F}V_i^{n+1} = 0) \wedge \left(|V_i^{n+1} - V_i^*| \leq \frac{C}{Large} \right) \tag{4.17}$$

where C is independent of Large, $\Delta\tau$, and ΔS.

In other words, we can obtain an approximate solution to the original discrete LCP (4.11) to any desired degree of precision. In practice, we can use the following heuristic argument to estimate the size of *Large* in terms of the relative accuracy required. In equation (4.10), suppose that $(1-\theta)\hat{M}V^{n+1}$ and $(I-\theta\hat{M})V^n$ are bounded independent of *Large*. Then, as *Large* $\to \infty$, equation (4.10) reduces to

$$V_i^{n+1} \simeq \left(\frac{Large}{1 + Large} \right) V_i^* \tag{4.18}$$

for nodes where $V_i^{n+1} < V_i^*$. If $V_i^* \neq 0$, then we have

$$\left| \frac{V_i^{n+1} - V_i^*}{V_i^*} \right| \simeq \frac{1}{Large}. \tag{4.19}$$

Therefore, if we require that the LCP be computed with a relative precision of *tol* for those nodes where $V_i^{n+1} < V_i^*$ then we should have $Large \simeq 1/tol$.

Note that in theory, if we are taking the limit as $\Delta S, \Delta\tau \to 0$, then we should have

$$Large = O\left(\frac{1}{\min\left[(\Delta S)^2, (\Delta\tau)^2\right]}\right). \tag{4.20}$$

This would mean that any error in the penalized formulation would tend to zero at the same rate as the discretization error. However, in practice it seems easier (to us at any rate) to specify the value of *Large* in terms of the required accuracy. In other words, we specify the maximum allowed error in the discrete penalized problem. We then reduce $\Delta S, \Delta\tau$ until the discretization error is reduced to this level of accuracy.

5. Penalty Iteration. Typically, a relaxation method is used to solve the discrete penalized equations [18, 6]. However, as noted earlier, such a technique suffers from poor complexity. We take a different approach, and simply apply full Newton iteration to the discrete nonlinear equations. Of course, due regard must be paid to the discontinuous derivative which appears in the penalty term.

Consistent with equation (4.9), we define

$$\frac{\partial P_i^{n+1}}{\partial V_i^{n+1}} = \begin{cases} Large & \text{if } V_i^{n+1} < V_i^* \\ 0 & \text{otherwise,} \end{cases} \tag{5.1}$$

so that Newton iteration applied to equation (4.10) yields the following algorithm. Let $(V^{n+1})^k$ be the k^{th} estimate for V^{n+1}. If $(V^{n+1})^0 = V^n$, then

Penalty American Constraint Iteration

For $k = 0, \ldots$ until convergence

$$\left[I + \hat{M} + \bar{P}\left((V^{n+1})^k\right)\right](V^{n+1})^{k+1} = \left[I - \hat{M}\right]V^n + \bar{P}\left((V^{n+1})^k\right)V^* \tag{5.2}$$

$$\text{if } \max_i \frac{|(V_i^{n+1})^{k+1} - (V_i^{n+1})^k|}{\max(1, |(V_i^{n+1})^{k+1}|)} < tol \quad \text{quit}$$

EndFor.

Note that from definition (4.7) we have effectively replaced $\partial P_i^{n+1}/\partial V_i^{n+1}$ at $V_i^{n+1} = V_i^*$ by the limit as $V_i^{n+1} \to (V_i^*)^+$.

It is worthwhile at this point to determine the complexity of the above iteration, compared to an explicit evaluation of the American constraint. Assuming that all of the coefficients are stored, that Crank-Nicolson timestepping is used with non-constant timesteps, and that there are I nodes in the S direction, each iteration of the penalty algorithm requires

(i) $6I$ multiplies to evaluate the right hand side of equation (5.2), where we have made the pessimistic assumption that $\left[\bar{P}(V^{n+1})\right]V^*$ requires I multiplies. This step also determines the entries in $\hat{M}$, assuming all possible quantities are precomputed and stored.

(ii) $2I$ multiply/divides to factor the matrix in equation (5.2).

(iii) $3I$ multiply/divides for the forward and back solve.

(iv) I divides for the convergence test. (This is also pessimistic, since we can skip the test on the first iteration, or if no constraint switches have occurred.)

This gives a total of $12I$ multiply/divides per penalty iteration. If constant timesteps are used, $4I$ multiplies are needed to evaluate the right hand side of (5.2), leading to a total of $10I$ multiply/divides per penalty iteration.

If an explicit method is used to evaluate the constraint, then there is only one matrix solve per timestep. To be precise here, an explicit method for handling the constraint is

Explicit American Constraint Timestep

$$\begin{aligned}\left[I + (1-\theta)\hat{M}\right]\hat{V}^{n+1} &= \left[I - \theta\hat{M}\right]V^n \\ V^{n+1} &= \max(\hat{V}^{n+1}, V^*).\end{aligned} \tag{5.3}$$

For constant timesteps (assuming that all coefficients are precomputed and stored),

(i) $3I$ multiply/divides are required to evaluate the right hand side of equation (5.3), assuming that $\bar{P} = 0$;

(ii) assuming that the matrix is factored once and the factors stored, $3I$ multiply/divides are required for the forward and back solve;

giving a total of $6I$ multiply/divides per timestep. For non-constant timesteps,

(i) $5I$ multiply/divides are required to evaluate the right hand side of equation (5.3), assuming that $\bar{P} = 0$;

(ii) $2I$ multiply/divides are required to factor the matrix;

(iii) $3I$ multiply/divides are required for the forward and back solve;

giving a total of $10I$ multiply/divides per timestep.

6. Convergence of the Penalty Iteration. For notational convenience, we will define $\bar{P}^k \equiv \bar{P}\left((V^{n+1})^k\right)$ and $\bar{V}^k \equiv (V^{n+1})^k$ so that the basic penalty algorithm (5.2) can be written as

$$\left[I + \hat{M} + \bar{P}^k\right]\bar{V}^{k+1} = \left[I - \hat{M}\right]V^n + \bar{P}^k V^*. \tag{6.1}$$

In Appendix B, we prove the following result:

THEOREM 6.1 (Convergence of the nonlinear iteration of the penalized equations). *Under the assumptions that the matrix $\hat{M}$ in equation (4.10) is an M-matrix and that timesteps are selected so that condition (A.10) is satisfied, the nonlinear iteration (5.2) converges to the unique solution to equation (4.10). Moreover, the iterates converge monotonically, i.e. $\bar{V}^{k+1} \geq \bar{V}^k$ for $k \geq 1$.*

In [22], it was demonstrated experimentally that using a smooth form of the penalty function (4.4) did not aid convergence of the solution of the nonlinear equations. Intuitively, this is somewhat surprising. It might be expected that the switch type penalty function (4.4), which has a discontinuous derivative, might cause oscillations during the iterations. However, the above result concerning monotonic convergence explains why the penalty iteration works so well, even with a non-smooth derivative. Since $\bar{V}^{k+1} \geq \bar{V}^k$ for $k \geq 1$, in the worst case we have $\bar{V}_i^0 \geq V_i^*$, $\bar{V}_i^1 < V_i^*$, $\bar{V}_i^p > V_i^*$ for some $p \geq 2$. No further constraint switches will occur. In other words, for any given node, the iteration will not oscillate between $\bar{V}_i^k > V_i^*$ and $\bar{V}_i^{k+1} < V_i^*$ $(k \geq 1)$.

7. Numerical Examples. In order to carry out a careful convergence study, we need to take into consideration the fact that the payoff function (2.4) has only piecewise smooth derivatives. This can cause problems if Crank-Nicolson timestepping is used. Specifically, oscillatory solutions can be generated [24]. For example, if we consider a simple European put option, then we know that the asymptotic solution near the expiry time $\tau = 0$ and close to the strike K is [20]

$$\text{put value } = O\left(\tau^{1/2}\right). \tag{7.1}$$

This would suggest that $V_{ttt} = O(\tau^{-5/2})$. The *local* finite difference truncation error for a Crank-Nicolson step (near $\tau = 0$) would then be $O\left[V_{ttt}(\Delta\tau)^3\right]$. If we set $\tau = \Delta\tau$ (the first step) then the local error would be $O[(\Delta\tau)^{1/2}]$, resulting in poor convergence. Fortunately, this analysis is a bit too simplistic. The behavior of the solution in equation (7.1) is due to the non-smooth payoff near K, which causes V_{SS} to behave (near $\tau = 0, S = K$) as $O\left(\tau^{-1/2}\right)$ [20]. This large value of V_{SS} causes a very rapid smoothing effect due to the parabolic nature of the PDE. Consequently, if an appropriate timestepping method is used, we can expect the initial errors to be damped very quickly. However, there is a problem with Crank-Nicolson timestepping. Crank-Nicolson is only *A-stable*, not *strongly A-stable*. This means that some errors are damped very slowly, resulting in oscillations in the numerical solution.

Since a finite volume discretization in one dimension can be viewed as a special type of finite element discretization, we can appeal to the finite element analysis in [16]. This analysis was specifically directed towards the case of parabolic PDEs with non-smooth initial conditions. Essentially, in [16] it is shown that if we take constant timesteps with a Crank-Nicolson method, then second order convergence (in time) can be guaranteed if (i) after each non-smooth initial state, we take *two* fully implicit timesteps, and then use Crank-Nicolson thereafter (payoffs with discontinuous derivatives qualify as non-smooth); and (ii) the initial conditions are l_2 projected onto the space of basis functions. In our case, this means that the initial condition should be approximated by continuous, piecewise linear basis functions. This corresponds to the folklore in the finance literature about *smoothing* initial conditions [10].

However, consider the case of a simple payoff such as that for a put option. Although this has a discontinuous derivative at K, no smoothing is required provided we have a node at K. This is because we have a piecewise linear representation of the initial condition, consistent with the implied basis functions used in the finite volume method. This also explains why binomial lattice methods (see §12) have non-smooth convergence behavior (there is no node at K if the number of timesteps is odd). In the case of a discontinuous initial condition, smoothing is necessary since this is not in the space of continuous piecewise linear basis functions. Finally, we remark that although second order convergence does not guarantee that the solution is non-oscillatory, in practice the above methods work well.

We can demonstrate the effectiveness of the simple idea of taking two fully implicit methods at the start and Crank-Nicolson thereafter (which we will henceforth refer to as *Rannacher smoothing* [16]) for a European put option with known solution. We will use the rather extreme value of $\sigma = .8$ for illustrative purposes. Results are provided in Table 7.1, which demonstrates that the solution with no smoothing converges erratically as the grid spacing and timestep size are reduced. In contrast, the smoothed solution shows quadratic convergence.

The reason for the poor convergence of the non-smoothed runs can be be explained by examining plots of the value V, delta (V_S), and gamma (V_{SS}), as shown in the left side of Figure 7.1. (Recall that it is of practical importance to determine delta and gamma for hedging purposes [12]). Note that although the value appears smooth, oscillations appear in delta (near the strike) and are magnified in gamma. The same problem was run using Rannacher smoothing, and the results are shown in

		No Smoothing			Rannacher Smoothing		
Nodes	Timesteps	Value	Change	Ratio	Value	Change	Ratio
68	25	14.50470			14.41872		
135	50	14.41032	.09438		14.44357	.02485	
269	100	14.43238	.02215	4.3	14.44982	.00625	4.0
539	200	14.44246	.01008	2.2	14.45138	.00156	4.0
1073	400	14.44726	.00480	2.1	14.45177	.00039	4.0

TABLE 7.1

Value of a European put, $\sigma = .8$, $T = .25$, $r = .10$, $K = 100$, $S = 100$. Exact solution (to seven figures): 14.45191. *Change is the difference in the solution from the coarser grid. Ratio is the ratio of the changes on successive grids.*

the right side of Figure 7.1. The oscillations in delta and gamma have disappeared. All subsequent runs will use Rannacher smoothing.

It might appear appropriate to use a timestepping method with better error damping properties, such as a second order BDF method [2]. However, our experience with this method for complex American style problems (see [21]) was poor. We conjecture that this is due to a lack of smoothness in the time direction, causing problematic behavior for multistep methods. This effect will be addressed in some detail below.

8. Implicit and Explicit Handling of the American Constraint. We will now compare an implicit treatment of the American constraint (using the penalty technique) with an explicit treatment (see pseudo-code (5.3)). In these examples we use constant timesteps, a convergence tolerance of $tol = 10^{-6}$ (see pseudo-code (5.2)), and consequently a value of $Large = 10^6$. As an additional accuracy check, for all runs we also monitored the quantity

$$\text{max American error } = \max_{n,i} \frac{\max[0, (V_i^* - V_i^n)]}{\max(1, V_i^*)}. \tag{8.1}$$

This is a measure of the maximum relative error in enforcing the American constraint using the penalty method. In all of the following examples, the observed value of equation (8.1) was $\simeq 10^{-9}$ if the penalty method was used with $tol = 10^{-6}$.

Two volatility values were used in these examples: $\sigma = .2, .8$. We truncate the computational domain at $S = S_{\max}$, where condition (2.5) is applied. The grid for $\sigma = .2$ used $S_{\max} = 200$, while the grid for $\sigma = .8$ used $S_{\max} = 1000$. Both grids were identical for $0 < S < 200$. The grid for $\sigma = .8$ added additional nodes for $200 < S < 1000$.

Table 8.1 compares results for implicit (penalty method) and explicit handling of the American constraint with constant timesteps. First, we note that for the penalty method the total number of nonlinear iterations is roughly the same across the two values of σ at each refinement level. This indicates that the volatility parameter has little effect on the number of iterations required. Now consider the results for the case where $\sigma = .2$. Taking into account the work per unit accuracy, the implicit method is slightly superior to the explicit technique. However, note that the implicit method does not appear to be converging quadratically (the ratio of changes is about 3 instead of 4, which we would expect for quadratic convergence). The explicit method appears to be converging at a first order rate (ratio of 2). Now consider the high volatility ($\sigma = .8$) results. Taking into account the total work, it would appear that in this case the explicit method is a little better than the implicit method. The latter seems to have an error ratio of about 2.9, while the explicit method has a somewhat lower convergence rate.

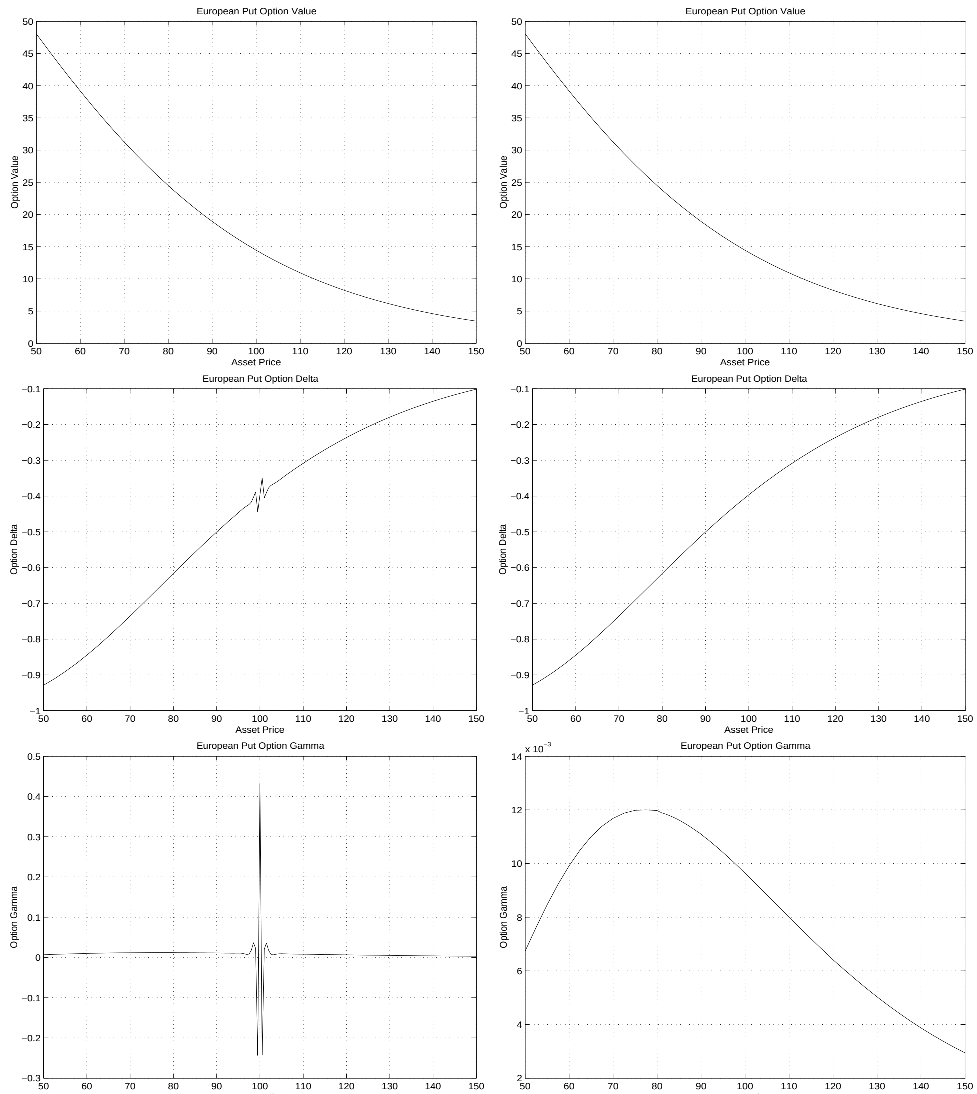

FIG. 7.1. *European put,* $\sigma = .8$*,* $T = .25$*,* $r = .10$*,* $K = 100$*. Crank-Nicolson timestepping, grid with 135 nodes. Left: no smoothing, right: Rannacher smoothing. Top: option value* (V)*, middle: delta* (V_S)*, bottom: gamma* (V_{SS})*.*

9. Analysis of Constant Timestep Examples. In terms of approximately solving the discrete LCP (4.11), the penalty method performs as the analysis predicts. The number of iterations per timestep is typically of the order $2.2 - 2.4$, independent of the volatility, for reasonable timesteps. Note that the algorithm (5.2) requires at least two iterations per timestep. The *a posteriori* check (8.1) (a maximum relative error of 10^{-9}, with $tol = 10^{-6}$ in terms of satisfaction of the discrete LCP constraint) indicates that the error introduced by the penalty method is quite small. This error is a function of tol (pseudo-code (5.2)), and hence can be adjusted to the desired level (which would of course affect the number of nonlinear iterations). Note that in these examples we have violated condition (A.10), which indicates that this condition

Nodes	Timesteps	Iters	Value	Change	Ratio	Work (flops)
explicit constraint, $\sigma = .2$						
55	25	25	3.04600			8.3×10^3
109	50	50	3.06049	.01449		3.3×10^4
217	100	100	3.06598	.00549	2.6	1.3×10^5
433	200	200	3.06822	.00224	2.5	5.2×10^5
865	400	400	3.06922	.00100	2.2	2.1×10^6
implicit constraint, $\sigma = .2$						
55	25	58	3.05607			3.2×10^4
109	50	117	3.06555	.00948		1.3×10^5
217	100	234	3.06854	.00299	3.2	5.1×10^5
433	200	471	3.06953	.00099	3.0	2.0×10^6
865	400	937	3.06988	.00035	2.8	8.1×10^6
explicit constraint, $\sigma = .8$						
68	25	25	14.61682			1.0×10^4
135	50	50	14.65685	.40020		4.1×10^4
269	100	100	14.67045	.01360	2.9	1.6×10^5
539	200	200	14.67542	.00497	2.7	6.5×10^5
1073	400	400	14.67738	.00196	2.5	2.6×10^6
implicit constraint, $\sigma = .8$						
68	25	56	14.62708			3.8×10^4
135	50	112	14.66219	.03511		1.5×10^5
269	100	226	14.67324	.01105	3.2	6.1×10^5
537	200	461	14.67686	.00362	3.1	2.5×10^6
1073	400	940	14.67813	.00127	2.9	1.0×10^7

TABLE 8.1

Value of an American put option, $T = .25$, $r = .10$, $K = 100$, $S = 100$. Iters is the number of nonlinear iterations. Change is the difference in the solution from the coarser grid. Ratio is the ratio of the changes on successive grids. Constant timesteps. Rannacher smoothing used. Work is measured in terms of number of multiply/divides.

is sufficient but not necessary for convergence of the penalty iteration. However, the results are disappointing in terms of the convergence of the discretization of the LCP. We do not observe quadratic convergence for the implicit handling of the American constraint.

An error ratio of about 2.8 would be consistent with global timestepping convergence at a rate of $O[(\Delta\tau)^{3/2}]$. Now, from [17, 19], we know that the value of an American call option (where the underlying asset pays a proportional dividend) behaves like $V = \text{const.} + O\left(\tau^{3/2}\right)$ near the exercise boundary and close to the expiration of the contract ($\tau \to 0$). This would give a value for $V_{\tau\tau\tau}$ in this region of

$$V_{\tau\tau\tau} = O[\tau^{-3/2}]. \tag{9.1}$$

It appears that the behavior of the American put near the exercise boundary and close to expiry is [14]

$$\begin{aligned} V &= \text{const.} + O[(\tau \log \tau)^{3/2}] \\ &\simeq \text{const.} + O[(\tau^{1-\epsilon})^{3/2})] \quad ; \qquad \epsilon > 0, \quad \epsilon \ll 1, \quad \tau \to 0. \end{aligned} \tag{9.2}$$

In the following we will ignore the ϵ in equation (9.2) and assume that the behavior of $V_{\tau\tau\tau}$ is given by equation (9.1).

From equation (9.1), the local time truncation error for Crank-Nicolson timestepping is (near the exercise boundary)

$$\text{local error } = O\left[\frac{(\Delta\tau)^3}{\tau^{3/2}}\right]. \tag{9.3}$$

Note that in this case there is no parabolic smoothing effect. This is due to the smooth-pasting condition [20] (i.e. V_S is continuous across the exercise boundary). This is because the diffusion term in the Black-Scholes equation approaches a finite limit at the exercise boundary. Assuming that the global error is of the order of the sum of the local errors, from equation (9.3) we obtain

$$\text{global error } = O\left[\sum_{i=1}^{i=1/\Delta\tau} \frac{(\Delta\tau)^3}{(i\Delta\tau)^{3/2}}\right] \simeq O[(\Delta\tau)^{3/2}], \tag{9.4}$$

which is consistent with the observed rate of convergence. Now, instead of taking constant timesteps, suppose we take timesteps which satisfy

$$\max_i(|V_i^{n+1} - V_i^n|) \simeq d, \tag{9.5}$$

where d is a specified constant. In order to take the limit to convergence, at each grid refinement we will halve both the grid spacing and d. It is reasonable to assume that the maximum change over a timestep (at least near $\tau = 0$) will occur near K. So, from equation (7.1),

$$\Delta V^{n+1} = \max_i(|V_i^{n+1} - V_i^n|) \simeq O\left[\frac{\Delta\tau^{n+1}}{\sqrt{\tau^n}}\right]. \tag{9.6}$$

Therefore, from equations (9.5) and (9.6), we have

$$\Delta\tau^{n+1} = O[d\sqrt{\tau^n}]. \tag{9.7}$$

Assuming a local error of the form (9.3), and using equations (9.3) and (9.7), this gives a local error with the variable timesteps satisfying equation (9.5) as

$$\text{local error } = O\left[\frac{\left((\Delta\tau)^{n+1}\right)^3}{(\tau^n)^{3/2}}\right] = O\left[\frac{d^3(\tau^n)^{3/2}}{(\tau^n)^{3/2}}\right] = O(d^3). \tag{9.8}$$

This implies a global error (with $O(1/d)$ timesteps) of

$$\text{global error } = O(d^2). \tag{9.9}$$

Therefore, suppose that we take variable timesteps consistent with (9.5). Then at each refinement stage, where we double the number of grid nodes, and double the number of timesteps (by halving d), we should see quadratic convergence. Note that we should reduce the initial timestep $\Delta\tau^0$ by four at each refinement. We make no claim that the above analysis of the time truncation error is in any way precise, but only suggestive of an appropriate timestepping strategy.

10. A Timestep Selector. The timestep selector used is based on a modified form of that suggested in [13]. Given an initial timestep $\Delta\tau^{n+1}$, then a new timestep is selected so that

$$\Delta\tau^{n+2} = \left(\min_i \left[\frac{\texttt{dnorm}}{\frac{|V(S_i,\tau^n+\Delta\tau^{n+1})-V(S_i,\tau^n)|}{\max(D,|V(S_i,\tau^n+\Delta\tau^{n+1})|,|V(S_i,\tau^n)|)}} \right] \right) \Delta\tau^{n+1}, \tag{10.1}$$

where `dnorm` is a target relative change (during the timestep) specified by the user. The scale D is selected so that the timestep selector does not take an excessive number of timesteps in regions where the value is small (for options valued in dollars, $D = 1$ is typically appropriate). In equation (10.1), we have normalized the factor used to estimate the new timestep. This is simply to avoid slow timestep growth for large values of the contract. This could be a problem with call options, for example, where the computational domain is truncated at a large value of S. If we did not examine the relative changes over a timestep, then it is possible that the timestep would be limited by large absolute changes in the solution (which would occur as $S \to \infty$), even though the relative changes were small.

Since $V(S_i = K, \tau \simeq 0) \simeq 0$, we expect that the denominator of equation (10.1) will take its largest value near $S = K$, since V increases rapidly there. Consequently, the timestep selector (10.1) will approximately enforce the condition that

$$\Delta V^{n+1} \simeq D \times \texttt{dnorm}. \tag{10.2}$$

Hence we will have

$$\Delta\tau^{n+1} = O(\texttt{dnorm}\sqrt{\tau^n}), \tag{10.3}$$

so that we should see a global error of $O[(\texttt{dnorm})^2]$, which follows from equation (9.9).

Note that timestep selector (10.1) estimates the change in the solution at the new timestep based on changes observed over the old timestep. Some adjustments can be made to this simple model if a more precise form for the time evolution of the solution is assumed, but we prefer (10.1) since it is simple and conservative.

In practice, we select a $(\Delta\tau)^0$ for the coarsest grid, and then $(\Delta\tau)^0$ is cut by four at each grid refinement. There is not much of a penalty for underestimating a suitable $(\Delta\tau)^0$ since the timestep will increase rapidly if the estimate is too conservative. In the following runs, we used values of $(\Delta\tau)^0 = 10^{-3}$ and $\texttt{dnorm} = .2$ on the coarsest grid. The value of `dnorm` was reduced by two at each grid refinement.

11. Variable Timestep Examples. Table 11.1 presents results for the cases considered in Table 8.1, but this time using the timestep selector (10.1). The timestep selector requires one divide per node, so we assume that the work required for each iteration of the implicit method is 13 multiply/divides per node, while the explicit method requires 11 multiply/divides per node per timestep. In this case, the implicit method appears to be a clear winner in terms of flops per unit accuracy. Use of variable timesteps actually seems to degrade the convergence of the explicit method. This can be explained by looking at the timestep history. The timestep selector uses small timesteps at the start, and then takes large steps at the end. Note that the average timestep size (total time divided by number of timesteps) is larger for the variable timestep run compared to the constant timestep run (Table 8.1). This clearly negatively impacts the explicit method, which seems to show a first order rate

Nodes	Timesteps	Iters	Value	Change	Ratio	Work (flops)
explicit constraint, $\sigma = .2$						
55	18	18	3.04499			1.1×10^4
109	33	33	3.05825	.01326		4.0×10^4
217	63	63	3.06425	.00600	2.2	1.5×10^5
433	122	122	3.06717	.00292	2.1	5.8×10^5
865	239	239	3.06863	.00146	2.0	2.3×10^6
implicit constraint, $\sigma = .2$						
55	18	45	3.06403			3.2×10^4
109	33	85	3.06867	.00464		1.2×10^5
217	63	164	3.06975	.00108	4.3	4.6×10^5
433	122	322	3.07002	.00027	4.0	1.8×10^6
865	239	636	3.07008	.00006	4.5	7.2×10^6
explicit constraint, $\sigma = .8$						
68	31	31	14.64828			2.3×10^4
135	66	66	14.66856	.02022		9.8×10^4
269	136	136	14.67472	.00616	3.3	4.0×10^5
537	276	276	14.67703	.00231	2.7	1.6×10^6
1073	554	554	14.67800	.00097	2.4	6.5×10^6
implicit constraint, $\sigma = .8$						
68	31	76	14.65863			6.7×10^4
135	66	161	14.67417	.01554		2.8×10^5
269	136	325	14.67778	.00361	4.3	1.1×10^6
537	276	655	14.67862	.00084	4.3	4.6×10^6
1073	554	1290	14.67882	.00020	4.2	1.8×10^7

TABLE 11.1

Value of an American put option, $T = .25$, $r = .10$, $K = 100$, $S = 100$. Iters is the number of nonlinear iterations. Change is the difference in the solution from the coarser grid. Ratio is the ratio of the changes on successive grids. Variable timesteps. Rannacher smoothing used. Work is measured in terms of number of multiply/divides.

of convergence. On the other hand, the implicit method appears to exhibit close to quadratic convergence.

Figure 11.1 shows value, delta, and gamma for the $\sigma = .2$ case, using both explicit and implicit treatments of the American constraint. Although the value and delta appear similar for both cases, there are clearly large oscillations in the gamma near the early exercise boundary for the explicit method. The implicit method does show some small oscillations near the exercise boundary. However, this is due to the use of Crank-Nicolson timestepping, as noted in [5]. These oscillations disappear if fully implicit timestepping is used, as shown in Figure 11.2.

12. Comparison With Binomial Lattice Methods. It is interesting to compare the results here with those obtained using the binomial lattice method, which is commonly used in finance [20]. In Appendix C, we show that this technique is simply an explicit finite difference method on a log-transformed grid. Consequently, the truncation error is $O(\Delta\tau)$, where the total number of steps is $N = O[1/(\Delta\tau)]$.

The binomial lattice method requires about $3/2N^2$ flops (counting only multiplies, and assuming all necessary factors are precomputed). Note that we obtain the value of the option at $t = 0$ only at the single point S_0^0, in contrast to the PDE methods which obtain values for all $S \in [0, S_{\max}]$. As a result, the methods are not directly comparable. Nevertheless, assuming that we are only interested in obtaining the solution at a single point, it is interesting and useful to compare these two techniques.

Given $N = O[1/(\Delta\tau)]$, the complexity of the binomial method is $O(N^2)$. Since

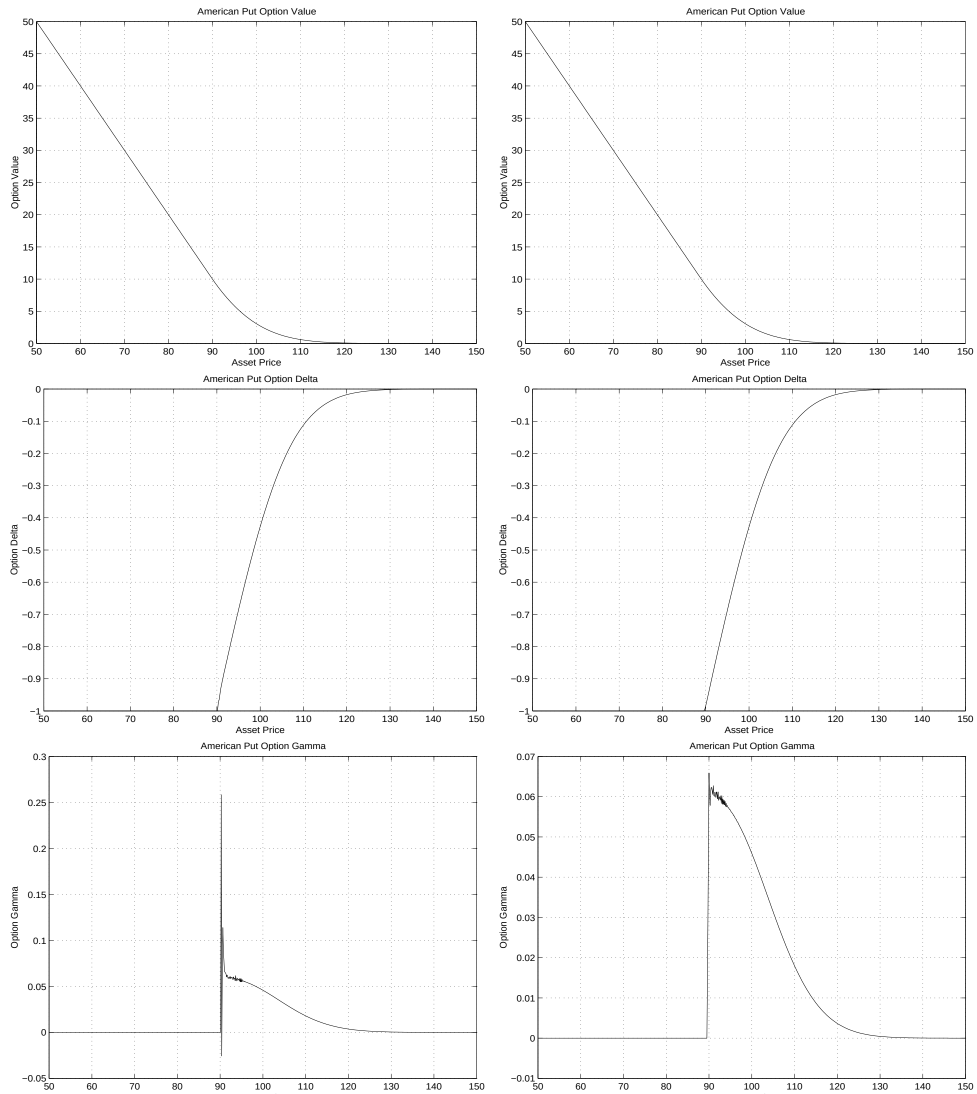

FIG. 11.1. *American put, $\sigma = .2$, $T = .25$, $r = .10$, $K = 100$. Crank-Nicolson timestepping, Rannacher smoothing, variable timesteps, grid with 433 nodes. Left: explicit constraint, right: implicit constraint. Top: option value (V), middle: delta (V_S), bottom: gamma (V_{SS}).*

the error in the lattice method is $O(\Delta\tau) = O(1/N)$, we have

$$\text{error binomial lattice } = O\left[(\text{complexity})^{-1/2}\right] . \tag{12.1}$$

Suppose instead that we use an implicit finite volume method with Crank-Nicolson timestepping, and that the penalty method is employed for handling the American constraint. The complexity of this approach is $O(N^2)$, where we have assumed that $N = O[1/(\Delta S)]$ (note that this is the case if we use the timestep selector (10.1) and $\texttt{dnorm} = O(\Delta S)$). It is also assumed that the the number of nonlinear iterations per timestep is constant as $\Delta S \rightarrow 0$, which is observed as long as $\texttt{dnorm} = O(\Delta S)$. When

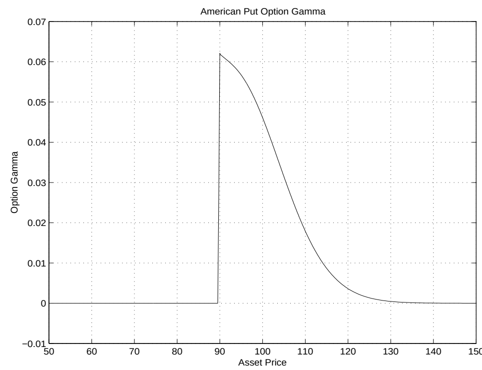

FIG. 11.2. *Gamma (V_{SS}) of an American put, $\sigma = .2$, $T = .25$, $r = .10$, $K = 100$. Fully implicit timestepping, Rannacher smoothing, variable timesteps. Grid with 433 nodes used. Constraint imposed implicitly.*

Timesteps	Value	Change	Ratio	Work (flops)
$\sigma = .2$				
50	3.06186			3.8×10^3
100	3.06611	0.00425		1.5×10^4
200	3.06810	0.00199	2.1	6.0×10^4
400	3.06913	0.00103	1.9	2.4×10^5
800	3.06962	0.00049	2.1	9.6×10^5
1600	3.06987	0.00025	2.0	3.8×10^6
3200	3.06999	0.00012	2.1	1.5×10^7
$\sigma = .8$				
50	14.62649			3.8×10^3
100	14.65269	0.02620		1.5×10^4
200	14.66582	0.01313	2.0	6.0×10^4
400	14.67238	0.00656	2.0	2.4×10^5
800	14.67563	0.00325	2.0	9.6×10^5
1600	14.67726	0.00163	2.0	3.8×10^6
3200	14.67807	0.00081	2.0	1.5×10^7

TABLE 12.1

Binomial lattice method. Value of an American put, $T = .25$, $r = .10$, $K = 100$, $S = 100$. Change is the difference in the solution from the coarser grid. Ratio is the ratio of the changes on successive grids. Work is measured as the number of multiplies.

timesteps are selected using (10.1), we have observed quadratic convergence. This implies

$$\text{error implicit finite volume } = O\left(N^{-2}\right) = O\left[(\text{complexity})^{-1}\right]. \qquad (12.2)$$

Therefore the implicit finite volume method is asymptotically superior to the binomial lattice method, even if the solution is desired at only one point.

It is interesting to determine at what levels of accuracy we can expect the implicit PDE method to become more efficient than the binomial method. Table 12.1 gives the results for a binomial lattice solution (algorithm (C.2)) for the problems solved earlier using an implicit PDE approach. This table should be compared to Table 11.1.

For further points of comparison, we also computed solutions to the problem used in [10]. We used two versions of the problem in [10], one with an expiry time of $T = 1$ and the other with $T = 5$. Figure 12.1 summarizes the convergence of both the binomial lattice and PDE methods for all four problems. The absolute

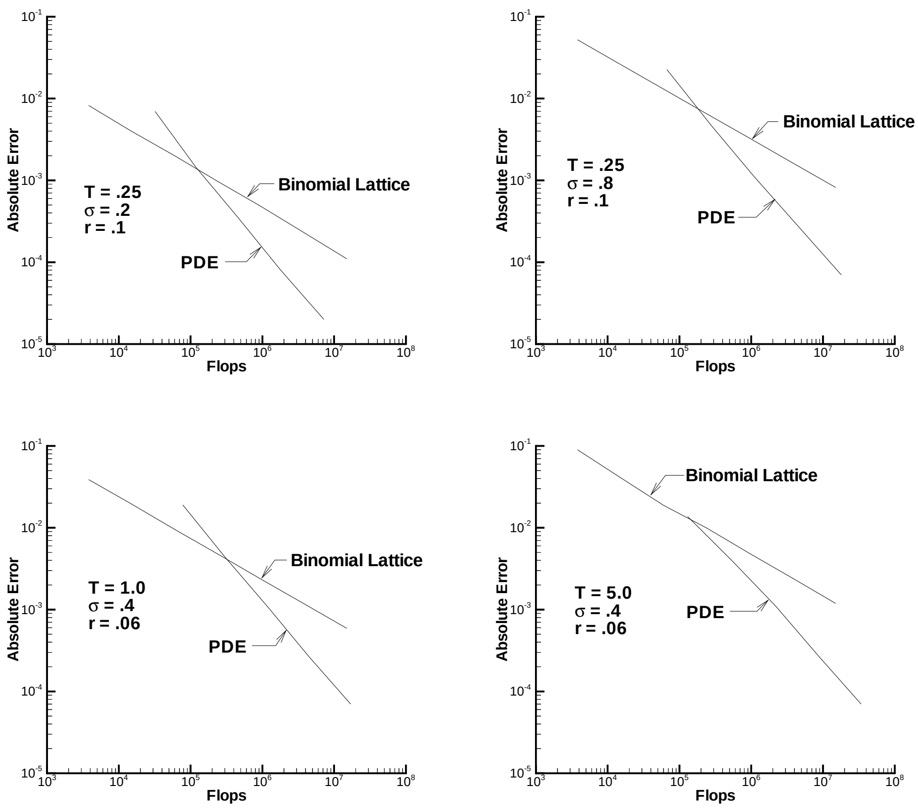

FIG. 12.1. *American Put,* $K = 100$. *Absolute error as a function of number of floating point operations (flops), measured as the number of multiplies/divides for all the test problems, at* $S = 100$, $t = 0$.

error is computed by taking the *exact* solution as obtained by extrapolating the PDE solution down to zero grid and timestep size, assuming quadratic behavior. The PDE method becomes more efficient than the binomial lattice method at tolerances between $.01 - .001$ depending on the problem parameters. It appears that for short term or low volatility options, the crossover point is closer to .001, while for long term or high volatility options, the crossover is closer to .01. These crossover points occur at tolerances which would be used in practice. Note that in these comparisons, we are putting the best possible light on the binomial lattice method, since we ignore the fact that we obtain much more information with the implicit PDE technique.

13. Application of Penalty Methods to More General Problems. As derived in the Appendices, sufficient conditions for monotone convergence of the penalty iteration are that (i) the discretized differential operator is an M-matrix; and (ii) condition (A.10) on the timestep is satisfied. In practice, we have found that these conditions are not necessary for rapid convergence of the penalty iteration. For example in [22], we have applied the penalty method to American options with stochastic volatility, with good results. In this case, the discretized differential operator was not an M-matrix, and if a flux limiter was used, the discretized differential operator was nonlinear. We also routinely violated the timestep condition (A.10). As long as the Rannacher smoothing technique was used, the solution was sufficiently smooth that no ill effects were observed with the penalty iteration.

Note that the discrete M-matrix condition was not required in the proof of convergence of the penalty method for an elliptic obstacle problem with the Laplacian as the differential operator [15]. However, timestep restrictions were required in the proofs of convergence of the penalty method for parabolic problems in [18]. In view of our computational experience, it appears to us that these conditions are artificial. We conjecture that the penalty iteration converges monotonically under much weaker conditions than those outlined in the Appendices.

14. Conclusion. We have derived sufficient conditions so that the solution of the discrete penalized equations solves an approximate version of the discrete LCP formulation of the American option pricing problem. The error in the approximation can be made arbitrarily small by increasing the penalty factor. We have also given sufficient conditions so that a Newton iteration method for solving the discrete nonlinear penalized equations converges monotonically to the unique solution of the nonlinear algebraic equations. This explains the observed rapid convergence of this technique.

If constant timesteps are used, the computed solution appears to converge at less than a second order rate in the limit as the grid spacing and timestep are reduced to zero. An heuristic analysis of the behavior of the solution near the exercise boundary indicates that convergence (with constant timesteps) occurs only at the rate $(\Delta t)^{3/2}$. However, a timestep selection method was suggested which, based on our analysis, should be expected to restore quadratic convergence. Numerical experiments confirmed that this convergence rate was indeed obtained using this timestep selector.

In general, the use of an implicit penalty method combined with the timestep selector can be recommended. As well as being more efficient in terms of the number of flops per unit accuracy, the solution obtained using an implicit method is qualitatively superior to the solution obtained using an explicit method for handling the American constraint. The explicit solution exhibited large oscillations in gamma near the exercise boundary.

The implicit PDE method is asymptotically superior to the standard (in finance) binomial lattice method, which has only linear convergence. However, if low accuracy solutions are required at only a single point, then a binomial method can be more efficient than the PDE approach. For typical parameters, the crossover point where the PDE method is to be preferred occurs at an absolute error tolerance of between $.01-.001$. However, if information at more than a single point is desired, then the PDE method is always preferable. As well, the binomial lattice method is highly optimized for simple cases. For example, the addition of discretely observed barriers [3, 25] causes difficulties for binomial methods. However, this case presents no particular difficulty for a PDE finite volume method.

The penalty method described here has been applied to multi-dimensional problems as shown in [22]. This method has the advantage that standard sparse matrix software can be used to solve the Jacobian matrix. This is especially important for multi-factor problems. In fact, the penalty method in [22] was applied to problems which did not satisfy the sufficient conditions derived in this work, with no apparent ill-effects. It is a topic of further research to extend the convergence results in this paper to the more general problems described in [22].

Appendix A. Error in the Penalty Formulation. In this Appendix, we determine sufficient conditions which allow us to bound (4.14). Suppose that node k is the node where the penalty term $P_k^{n+1}(V_k^* - V_k^{n+1})$ attains its maximum. Consider the term

$$[\hat{M}(V^* - V^{n+1})]_k = \Delta\tau \sum_{j \in \eta_k} \left(\bar{\gamma}_{kj} + \bar{\beta}_{kj}\right) [(V_k^* - V_k^{n+1}) - (V_j^* - V_j^{n+1})] + r\Delta\tau[V_k^* - V_k^{n+1}]. \tag{A.1}$$

Since the penalty term attains its maximum value at node k, we have

$$\begin{aligned} [(V_k^* - V_k^{n+1}) - (V_j^* - V_j^{n+1})] &\geq 0 \\ V_k^* - V_k^{n+1} &\geq 0. \end{aligned} \tag{A.2}$$

Since $\bar{\gamma}_{kj} + \bar{\beta}_{kj} \geq 0$, it follows from equations (A.1-A.2) that $[\hat{M}(V^* - V^{n+1})]_k \geq 0$ at node k. Alternatively,

$$[\hat{M}(V^{n+1})]_k \leq [\hat{M}(V^*)]_k, \tag{A.3}$$

implying

$$[I + (1-\theta)\hat{M}(V^{n+1})]_k \leq [I + (1-\theta)\hat{M}(V^*)]_k. \tag{A.4}$$

Writing equation (4.10) at node k, we have

$$\left(\left[I + (1-\theta)\hat{M}\right] V^{n+1}\right)_k = \left(\left[I - \theta\hat{M}\right] V^n\right)_k + \left(\left[\bar{P}(V^{n+1})\right](V^* - V^{n+1})\right)_k. \tag{A.5}$$

Noting that $P_k^{n+1}(V_k^* - V_k^{n+1}) \geq 0$, it follows from equations (A.5) and (A.3) that

$$\begin{aligned} \left|[P_k^{n+1}(V_k^* - V_k^{n+1})]\right| &= \left\|P^{n+1}(V^* - V^{n+1})\right\|_\infty \\ &\leq \left|\left(\left[I + (1-\theta)\hat{M}\right] V^{n+1}\right)_k\right| + \left|\left(\left[I - \theta\hat{M}\right] V^n\right)_k\right| \\ &\leq \left|\left(\left[I + (1-\theta)\hat{M}\right] V^*\right)_k\right| + \left|\left(\left[I - \theta\hat{M}\right] V^n\right)_k\right| \\ &\leq \left\|\left[I + (1-\theta)\hat{M}\right] V^*\right\|_\infty + \left\|\left[I - \theta\hat{M}\right] V^n\right\|_\infty \\ &\leq \|V^*\|_\infty + (1-\theta)\left\|\hat{M}V^*\right\|_\infty + \left\|\left[I - \theta\hat{M}\right] V^n\right\|_\infty. \end{aligned} \tag{A.6}$$

We now proceed to bound each of the terms on the right hand side of equation (A.6). Given a typical payoff of the form

$$V^* = V^0 = \max(K - S, 0) \tag{A.7}$$

where K is the strike, we have $\|V^*\|_\infty = K$. In bounding $||\hat{M}V^*||_\infty$, we note that the worst case occurs at $S_i = K$, so that

$$\begin{aligned} \|\hat{M}V^*\|_\infty &\leq \text{const}\,.\,|\hat{M}V^*|_i \quad ; \qquad S_i = K \\ &= O\left(\frac{\Delta\tau}{\Delta S}\right), \end{aligned} \tag{A.8}$$

where $\Delta S = \min_i(S_i - S_{i-1})$. We assume that the timestep and mesh size are reduced to zero in such a way that

$$\frac{\Delta\tau}{\Delta S} = \text{const}\,., \tag{A.9}$$

where this constant is independent of $\Delta\tau, \Delta S$. (It does not make any sense to drive the S discretization to zero if the timestep truncation error is also not reduced as well.) Consequently, we can assume that $\|\hat{M}V^*\|_\infty$ is bounded independent of *Large* and $\Delta\tau, \Delta S$.

If we also assume that the timestep is selected so that

$$1 - \theta\left(\Delta\tau \sum_{j\in\eta_i}\left(\bar{\gamma}_{ij} + \bar{\beta}_{ij}\right) + r\Delta\tau\right) \geq 0, \tag{A.10}$$

then we have (recalling that $\hat{M}$ is an M-matrix with row sum $r\Delta\tau$)

$$\left\|\left[I - \theta\hat{M}\right]V^n\right\|_\infty \leq (1 - r\Delta\tau)\,\|V^n\|_\infty \leq \|V^n\|_\infty\,. \tag{A.11}$$

Assuming condition (A.10) is satisfied, it follows from equations (4.10) and (A.7) that

$$\|V^n\|_\infty \leq \max(\left\|V^{n-1}\right\|_\infty, \|V^*\|_\infty) = \|V^*\|_\infty = K. \tag{A.12}$$

Note that if a fully implicit discretization is used ($\theta = 0$), then condition (A.10) is trivially satisfied. For Crank-Nicolson timestepping, condition (A.10) implies that $\Delta\tau/(\Delta S)^2 \leq \text{const}\,.$ as $\Delta S, \Delta\tau \to 0$.

Consequently, we have shown that

$$\left\|P^{n+1}(V^* - V^{n+1})\right\|_\infty \leq 2K + O\left(\frac{\Delta\tau}{\Delta S}\right). \tag{A.13}$$

In other words, at any node where $V_i^{n+1} < V_i^*$, we have $|Large(V_i^* - V_i^{n+1})| \leq C$, where C is independent of *Large*. Therefore, by choosing *Large* sufficiently large, the error in the solution of the LCP can be made arbitrarily small, and Theorem 4.1 follows.

Appendix B. Monotone Convergence. We will first prove that iteration (6.1) has a *monotone* property. Writing (6.1) for iteration k gives

$$\left[I + \hat{M} + \bar{P}^{k-1}\right]\bar{V}^k = \left[I - \hat{M}\right]V^n + \bar{P}^{k-1}V^*. \tag{B.1}$$

This can be written as

$$\left[I + \hat{M} + \bar{P}^k\right]\bar{V}^k + \left[\bar{P}^{k-1} - \bar{P}^k\right]\bar{V}^k = \left[I - \hat{M}\right]V^n + \bar{P}^{k-1}V^*. \tag{B.2}$$

Subtracting equation (B.2) from equation (6.1) gives

$$\left[I + \hat{M} + \bar{P}^k\right](\bar{V}^{k+1} - \bar{V}^k) = \left[\bar{P}^k - \bar{P}^{k-1}\right](V^* - \bar{V}^k) \quad ; \qquad k \geq 1. \tag{B.3}$$

Now examine each of the components of the right hand side of equation (B.3). There are two possible cases:

$$\begin{aligned}
\text{Case 1:}\qquad \bar{V}_i^k < V_i^* &\Rightarrow \bar{P}_{ii}^k = Large\\
&\Rightarrow (Large - \bar{P}_{ii}^{k-1})(V^* - \bar{V}^k)_i \geq 0\\
&\Rightarrow \left[\bar{P}^k - \bar{P}^{k-1}\right]_i (V^* - \bar{V}^k)_i \geq 0,\\
\text{Case 2:}\qquad \bar{V}_i^k \geq V_i^* &\Rightarrow \bar{P}_{ii}^k = 0\\
&\Rightarrow (-\bar{P}_{ii}^{k-1})(V^* - \bar{V}^k)_i \geq 0\\
&\Rightarrow \left[\bar{P}^k - \bar{P}^{k-1}\right]_i (V^* - \bar{V}^k)_i \geq 0.
\end{aligned}$$

Thus we always have

$$\left[\bar{P}^k - \bar{P}^{k-1}\right](V^* - \bar{V}^k) \geq 0 \quad ; \qquad k \geq 1. \tag{B.4}$$

Since $\left[I + \hat{M} + \bar{P}^k\right]$ is an M-matrix, it follows from equations (B.3-B.4) that $(\bar{V}^{k+1} - \bar{V}^k) \geq 0$ for $k \geq 1$, or, in component form, $(\bar{V}^{k+1} - \bar{V}^k)_i \geq 0 \; \forall i$ for $k \geq 1$.

If $\bar{V}_i^k \geq V_i^*$, then $\bar{V}_i^{k+1}$ is bounded by the strike K provided condition (A.10) is satisfied. This follows from the payoff condition (2.4) and equation (4.10) since $\hat{M}$ is an M-matrix with row sum $r\Delta\tau$, and $(I - \theta\hat{M})$ is a matrix with non-negative entries and row sum $1 - \theta r \Delta\tau$. Since the iterates form a non-decreasing bounded sequence, it follows that the penalty iteration (6.1) converges.

We now demonstrate that the solution obtained by the penalty iteration is unique. Suppose there are two solutions $\bar{V}_1$ and $\bar{V}_2$ to the penalized problem. Let $\bar{P}^1 \equiv P(\bar{V}_1)$ and $\bar{P}^2 \equiv P(\bar{V}_2)$. Then

$$\left[I + \hat{M} + \bar{P}^1\right]\bar{V}_1 = \left[I - \hat{M}\right]V^n + \bar{P}^1 V^* \tag{B.5}$$

$$\left[I + \hat{M} + \bar{P}^2\right]\bar{V}_2 = \left[I - \hat{M}\right]V^n + \bar{P}^2 V^*. \tag{B.6}$$

We can write equation (B.5) as

$$\left[I + \hat{M} + \bar{P}^2\right]\bar{V}_1 + \left[\bar{P}^1 - \bar{P}^2\right]\bar{V}_1 = \left[I - \hat{M}\right]V^n + \bar{P}^1 V^*. \tag{B.7}$$

Subtracting equation (B.6) from equation (B.7) gives

$$\left[I + \hat{M} + \bar{P}^2\right](\bar{V}_1 - \bar{V}_2) = \left[\bar{P}^1 - \bar{P}^2\right](V^* - \bar{V}_1). \tag{B.8}$$

Using a similar argument as we used in proving monotone iteration, we have that

$$\left[\bar{P}^1 - \bar{P}^2\right](V^* - \bar{V}_1) \geq 0. \tag{B.9}$$

Since $\left[I + \hat{M} + \bar{P}^2\right]$ is an M-matrix, it follows from equations (B.8-B.9) that $(\bar{V}_1 - \bar{V}_2) \geq 0$. Interchanging subscripts, we have $(\bar{V}_2 - \bar{V}_1) \geq 0$, and hence $\bar{V}_2 = \bar{V}_1$. Consequently, Theorem 6.1 follows.

Appendix C. The Binomial Lattice Method. Let $S_m^n = u^{2m-n} S_0^0$ for $m = 0, \ldots, n$ denote the value of the asset price at time $t_n = n\Delta t$ and lattice point m, where $u = e^{\sigma\sqrt{\Delta t}}$ and $\Delta t = T/N$. Note that T is the expiry time of the option and N is the number of timesteps. Also note that we are considering time t going forward in this case, in contrast to $\tau = T - t$ (time going backwards) as in the previous sections. This results in a solution algorithm which proceeds backwards from $t = T$ to $t = 0$ (i.e. from $t = t_N$ to $t = 0$).

Let V_m^n be the value of the option associated with asset price S_m^n, at time $t = n\Delta t$. Of course, we have $V_m^N = \max(K - S_m^N, 0)$ for $m = 0, \ldots, N$. Define

$$p = \frac{e^{r\Delta t} - e^{-\sigma\sqrt{\Delta t}}}{e^{\sigma\sqrt{\Delta t}} - e^{-\sigma\sqrt{\Delta t}}}. \tag{C.1}$$

Then the value of the American put option V_0^0 (at the single point $S = S_0^0$) is obtained from the following algorithm:

Binomial Lattice Algorithm

For $n = N-1, \ldots, 0$

For $m = 0, \ldots, n$

$$\begin{aligned}\bar{V} &= e^{-r\Delta t}\left(pV_{m+1}^{n+1} + (1-p)V_m^{n+1}\right) \\ V_m^n &= \max(K - S_m^n, \bar{V})\end{aligned} \tag{C.2}$$

EndFor

EndFor.

The above method is usually derived in the financial literature based on probabilistic arguments. In fact, we can see that this is equivalent to an explicit finite difference method with a particular choice for the timestep. Consider the Black-Scholes equation for a European option:

$$V_t + \frac{\sigma^2}{2}S^2 V_{SS} + rSV_S - rV = 0. \tag{C.3}$$

Define a new variable $X = \log S$, so that equation (C.3) becomes

$$V_t + \frac{\sigma^2}{2}V_{XX} + (r - \frac{\sigma^2}{2})V_X - rV = 0. \tag{C.4}$$

Letting $V = e^{rt}W$, equation (C.4) becomes

$$W_t + \frac{\sigma^2}{2}W_{XX} + (r - \frac{\sigma^2}{2})W_X = 0. \tag{C.5}$$

Now let $W_m^n = W(\log S_0^0 + (2m-n)\sigma\sqrt{\Delta\tau}, n\Delta\tau)$ for $m = 0, \ldots, n$. Discretizing equation (C.5) using central differencing in the X direction and an explicit timestepping method, we obtain

$$W_m^n = \left[p^*\left(W_{m+1}^{n+1}\right) + (1-p^*)\left(W_m^{n+1}\right)\right] + O[(\Delta t)^2] \tag{C.6}$$

where $p^* = 1/2\left[1 + \sqrt{\Delta t}\,(r/\sigma - \sigma/2)\right]$. Writing (C.6) in terms of V_m^n gives

$$V_m^n = e^{-r\Delta t}\left[p^*\left(V_{m+1}^{n+1}\right) + (1-p^*)\left(V_m^{n+1}\right)\right] + O[(\Delta t)^2]. \tag{C.7}$$

Expanding p in equation (C.1) in a Taylor series, noting the definition of p^*, and assuming that $V_{m+1}^{n+1} - V_m^{n+1} = O(\sqrt{\Delta\tau})$, we obtain

$$V_m^n = e^{-r\Delta t}\left[p\left(V_{m+1}^{n+1}\right) + (1-p)\left(V_m^{n+1}\right)\right] + O[(\Delta t)^2]. \tag{C.8}$$

Comparing equation (C.8) with algorithm (C.2), we can see that the binomial lattice method is simply an explicit finite difference discretization of the discrete LCP (2.2), with the American constraint applied explicitly.

REFERENCES

[1] M. Avellaneda, A. Levy, and A Parás. Pricing and hedging derivative securities in markets with uncertain volatilities. *Appl. Math. Fin.*, 2:73–88, 1995.

[2] J. Becker. A second order backward difference method with variable timesteps for a parabolic problem. *BIT*, 38:644–662, 1998.

[3] P.P. Boyle and S.H. Lau. Bumping up against the barrier. *J. Derivatives*, 1:6–14, Summer 1994.
[4] N. Clarke and K. Parrot. The multigrid solution of two factor American put options. *Appl. Math. Fin.*, 6:177–195, 1999.
[5] T.F. Coleman, Y.Li, and A. Verma. A Newton method for American option pricing. Cornell Theory Center Technical Report CTC9907, 1999.
[6] K. Deckelnick and K. Siebert. $W^{1,\infty}$ convergence of the discrete free boundary for obstacle problems. *IMA J. Num. Anal.*, 20:481–489, 2000.
[7] M.A.H. Dempster and J.P. Hutton. Fast numerical valuation of American, exotic and complex options. *Appl. Math. Fin.*, 4:1–20, 1997.
[8] E.M. Elliot and J.R. Ockendon. *Weak and Variational Methods for Moving Boundary Problems*. Pitman, 1982.
[9] A. Friedman. *Variational Principles and Free-Boundary Problems*. Robert E. Krieger, 1988.
[10] S. Heston and G. Zhou. On the rate of convergence of discrete-time contingent claims. *Math. Fin.*, 10:53–75, 2000.
[11] J. Huang and J. Pang. Option pricing and linear complimentarity. *J. Comp. Fin.*, 2:31–60, Fall 1998.
[12] J. Hull. *Options, Futures and Other Derivatives*. Prentice Hall, 4th edition, 2000.
[13] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 1987.
[14] R. Kuske and J. Keller. Optimal exercise boundary for an American put option. *Appl. Math. Fin.*, 5:107–116, 1998.
[15] R. Nochetto. Sharp l^∞ error estimates for semilinear problems with free boundaries. *Num. Math.*, 54:243–255, 1988.
[16] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Num. Math.*, 43:309–327, 1984.
[17] I. Rupf, J. Dewynne, S. Howison, and P. Wilmott. Some mathematical results in the pricing of American options. *Euro. J. Appl. Math.*, 4:381–398, 1993.
[18] R. Scholz. Numerical solution of the obstacle problem by a penalty method: Part ii, time dependent problems. *Num. Math.*, 49:255–268, 1986.
[19] P. Wilmott. *Derivatives*. Wiley, 1998.
[20] P. Wilmott, J. Dewynne, and S. Howison. *Option Pricing*. Oxford Financial Press, 1993.
[21] H. Windcliff, P.A. Forsyth, and K.R. Vetzal. Shout options: A framework for pricing contracts which can be modified by the investor. *J. Comp. Appl. Math.*, forthcoming, 1999.
[22] R. Zvan, P. A. Forsyth, and K. R. Vetzal. Penalty methods for American options with stochastic volatility. *J. Comp. Appl. Math.*, 91:119–218, 1998.
[23] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Robust numerical methods for PDE models of Asian options. *J. Comp. Fin.*, 1:39–78, Winter 1998.
[24] R. Zvan, K.R. Vetzal, and P.A. Forsyth. Swing low, swing high. *RISK*, 11:71–75, March 1998.
[25] R. Zvan, K.R. Vetzal, and P.A. Forsyth. PDE methods for pricing barrier options. *J. Econ. Dyn. Con.*, 24:1563–1590, 2000.