SIAM J. SCI. COMPUT.
Vol. 33, No. 5, pp. 2144–2168

METHODS FOR PRICING AMERICAN OPTIONS UNDER REGIME SWITCHING*

Y. HUANG†, P. A. FORSYTH‡, AND G. LABAHN‡

Abstract. We analyze a number of techniques for pricing American options under a regime switching stochastic process. The techniques analyzed include both explicit and implicit discretizations with the focus being on methods which are unconditionally stable. In the case of implicit methods we also compare a number of iterative procedures for solving the associated nonlinear algebraic equations. Numerical tests indicate that a fixed point policy iteration, coupled with a direct control formulation, is a reliable general purpose method. Finally, we remark that we formulate the American problem as an abstract optimal control problem; hence our results are applicable to more general problems as well.

Key words. regime switching, American options, iterative methods

AMS subject classifications. 65N06, 93C20

DOI. 10.1137/110820920

1. Introduction.

The standard approach to valuation of contingent claims (also known as derivatives) is to specify a stochastic process for the underlying asset and then construct a dynamic, self-financing hedging portfolio to minimize risk. The initial cost of constructing the portfolio is then considered to be the fair value of the contingent claim. This has been used with great success in the case of stochastic processes having constant volatility in the case of both European and (the more difficult) American options.

However, it is well known that a financial model which follows a stochastic process having constant volatility is not consistent with market prices. Recent research has shown that models based on stochastic volatility, jump diffusion, and regime switching processes produce better fits to market data. A nonexhaustive list of regime switching applications includes insurance [22], electricity markets [21, 40], natural gas [12, 2], optimal forestry management [11], trading strategies [15], valuation of stock loans [44], convertible bond pricing [3], and interest rate dynamics [27]. Regime switching models are intuitively appealing, and computationally inexpensive compared to a stochastic volatility jump diffusion model.

In this paper we study numerical techniques for the solution of American option contracts under regime switching. While our examples focus on problems with constant properties in each regime, the numerical methods developed can easily be applied to cases where the properties in each regime are more complex. An example would be the use of price-dependent regime switching (i.e., default hazard rates) in convertible bond pricing [4]. A number of different methods has been proposed for handling American options under regime switching models. Semianalytic approaches have been

*Submitted to the journal's Methods and Algorithms for Scientific Computing section January 13, 2011; accepted for publication (in revised form) June 1, 2011; published electronically September 1, 2011. This work was supported by Credit Suisse, New York and the Natural Sciences and Engineering Research Council of Canada.

http://www.siam.org/journals/sisc/33-5/82092.html

†Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1 (yqhuang@ecemail.uwaterloo.ca).

‡Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1 (paforsyt@uwaterloo.ca, glabahn@uwaterloo.ca).

suggested in, for example, [26, 9]. Numerical methods include lattice methods [25], penalty methods using explicit forms for the penalty term [29], and a Crank–Nicolson method suggested in [42]. However, in each case such methods have fundamental limitations. For example, while the semianalytic methods can be very efficient for certain classes of models, they are difficult to apply, in general, to problems with time- and asset-dependent coefficients, as would be typical of commodity applications [12, 2]. Lattice methods are popular with practitioners because they are easy to understand and to implement. However, they are essentially explicit finite difference techniques and as such have timestep limitations due to stability considerations. The penalty method of [29] uses an explicit coupling of the penalty term and the regime coupling terms. This avoids expensive iteration at each timestep, but at the cost of also incurring timestep limitations due to stability considerations.

We will focus exclusively on methods which are unconditionally stable and which can be easily generalized to handle a variety of stochastic price models. We model our American option under regime switching as a set of coupled partial differential equations (PDEs) variational inequalities (VIs). As a base case, we discretize these PDE-VIs and use an explicit method for the regime coupling terms and the American constraint. In order to develop more efficient methods, we formulate the discretized PDE-VIs using both a penalty method [20] and a direct control approach [8]. In these cases we use implicit methods for the regime coupling and the American constraint. This requires a solution of a system of nonlinear algebraic equations.

While implicit coupling methods are more expensive per step than explicit coupling methods, one also needs to consider the rate of convergence in order to compare various methods. In addition, there are a number of iterative methods available for solving the nonlinear algebraic equations. We carry out a convergence analysis of the iterative method used to solve the nonlinear discretized algebraic equations. It is convenient to consider these equations as a special case of the general form of discretized Hamilton–Jacobi–Bellman (HJB) equations, as discussed in [19, 23]. The previously mentioned numeric approaches (for regime switching) are all simply special cases of this general form. This allows us to use a single framework to analyze the convergence of various iterative methods. These include full policy iteration [30], fixed point policy iteration [23], and a method whereby the regime coupling terms are lagged at each iteration, but the American option problem is solved to high accuracy within each regime [37]. In addition, using the same framework, we also analyze a global-in-time iteration procedure suggested in [31] (see also [5, 6]) whereby a sequence of optimal stopping problems is solved. We include numerical tests that compare unconditionally stable methods which do not require the solution of discretized equations at each timestep with the approaches described above.

One significant advantage of our general approach is that our convergence results can be immediately applied to any type of optimal control problem (not just an American constraint) based on regime switching or Markov modulated jump diffusions [18]. We should also mention that these methods can also be applied to switching problems [34], which arise, for example, in optimal operation of power plants. Our methods also make no assumptions about the form of the American constraint. The numerical experiments indicate that use of Crank–Nicolson timestepping and direct control formulation, coupled with a fixed point policy iteration, is a very effective and general purpose method. At the other end of the spectrum we show that the theoretical upper bound on the rate of convergence of the global-in-time method coupled with its significant storage requirements makes this uncompetitive with the other methods.

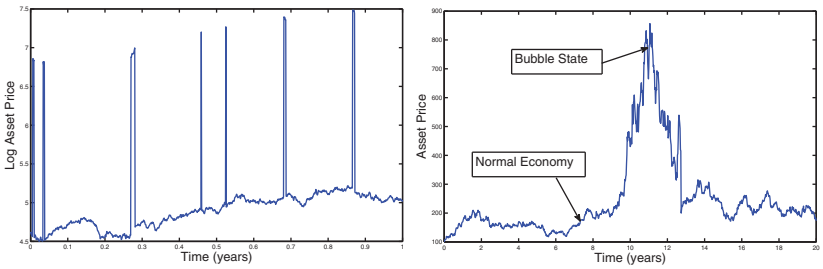

FIG. 2.1. *Examples of two state regime switching models. Left: parameters selected to simulate price spikes, typical of electricity prices. Right: parameters selected to simulate a bubble in asset prices.*

The remainder of the paper is organized as follows. The regime switching model is formulated in the next section with the no-arbitrage price of an American option given as a system of HJB equations. Section 3 details the three types of discretizations (explicit, implicit-direct control, and the implicit-penalty method) used for approximating the resulting optimal control equations. Section 4 describes the general form of the algebraic system of equations which occur for the two implicit discretizations. Section 5 considers the four distinct iterative methods used for solving the algebraic system of equations. The following section gives a numerical comparison of the various methods.

2. Regime switching: Formulation. Let $\sigma^j, j = 1, \ldots, K$ be a finite set of discrete volatilities for our model. Shifts between these states are controlled by a continuous Markov chain. Under the real world measure, the stochastic process for the underlying asset S is

$$dS = \mu_j^{\mathbb{P}}\ S\ dt + \sigma^j\ S\ dZ + \sum_{k=1}^{K}(\xi_{jk} - 1)\ S\ dX_{jk}, \quad j = 1, \ldots, K, \tag{2.1}$$

where dZ is the increment of a Wiener process, and $\mu_j^{\mathbb{P}}$ is the drift in regime j. In addition,

$$\begin{aligned} dX_{jk} &= \begin{cases} 1 \text{ with probability,} & \lambda_{jk}^{\mathbb{P}}\ dt + \delta_{jk}, \\ 0 \text{ with probability,} & 1 - \lambda_{jk}^{\mathbb{P}}\ dt - \delta_{jk}, \end{cases} \\ &\lambda_{jk}^{\mathbb{P}} \geq 0, \quad j \neq k, \\ &\lambda_{jj}^{\mathbb{P}} = -\sum_{\substack{k=1 \\ k \neq j}}^{K} \lambda_{jk}^{\mathbb{P}}. \end{aligned} \tag{2.2}$$

It is understood that there can be only one transition over any infinitesimal time interval, and that $\lambda_{jk}^{\mathbb{P}} \geq 0, j \neq k$. When a transition from $j \to k$ occurs, then the asset price jumps $S \to \xi_{jk} S$. For notational completeness, $\xi_{jj} = 1$. The superscript $\mathbb{P}$ refers to the objective probability measure. We assume that ξ_{jk} are deterministic functions of (S, t).

Regime switching processes are simple yet rich models of realistic stochastic phenomena observed in the economy. It is well known, for example, that a two state regime switching model with constant parameters can reproduce a volatility smile [43]. Figure 2.1 shows a single stochastic path for a two regime model, using different

parameters. The left plot shows *spike* effects that would be typical of electricity prices [21]. The right plot shows a stochastic path typical of an asset price bubble [38].

Let $\mathcal{V}_j(S,\tau)$ be the no-arbitrage value of our contingent claim in regime j where as usual we have $\tau = T - t$, so we are working backwards in time, with T being the expiry time of the contingent claim. Suppose we construct a hedging portfolio P such that

$$P = -\mathcal{V}_j + e\ S + \sum_{k=1}^{K-1} w_k F_k, \tag{2.3}$$

where e is the number of units of the underlying asset with price S, and w_k is the number of units of the additional hedging instruments with price F_k. Assuming that the set of assets with prices $\{S, F_1, \ldots, F_{K-1}\}$ forms a nonredundant set [28], it is possible to set up a perfect hedge. The existence of the perfect hedge allows us to define *risk neutral* transition probabilities λ_{jk} and the quantities

$$\lambda_{jj} = -\sum_{\substack{k=1\\k\neq j}}^{K} \lambda_{jk}, \quad \rho_j = \sum_{\substack{k=1\\k\neq j}}^{K} \lambda_{jk}(\xi_{jk} - 1), \quad \lambda_j = \sum_{\substack{k=1\\k\neq j}}^{K} \lambda_{jk}. \tag{2.4}$$

In practical applications, the quantities λ_{ij}, ξ_{ij} are determined by calibration to market prices [3].

Define the differential operators

$$\begin{aligned}\mathcal{L}_j\mathcal{V}_j &= \frac{\sigma_j^2 S^2}{2}\mathcal{V}_{j,SS} + (r-\rho_j)S\mathcal{V}_{j,S} - (r+\lambda_j)\mathcal{V}_j \\ &= \frac{\sigma_j^2 S^2}{2} D_{SS}\mathcal{V}_j + (r-\rho_j)SD_S\mathcal{V}_j - (r+\lambda_j)\mathcal{V}_j \\ \mathcal{J}_j\mathcal{V} &= \sum_{\substack{k=1\\k\neq j}}^{K} \frac{\lambda_{jk}}{\lambda_j}\mathcal{V}_k(\xi_{jk}S,\tau),\end{aligned} \tag{2.5}$$

with D_S and D_{SS} denoting the usual partial derivative operators and r the risk-free rate. The no-arbitrage price of an American option is then given by [28]

$$\min\left[\ \mathcal{V}_{j,\tau} - \mathcal{L}_j\mathcal{V}_j - \lambda_j\mathcal{J}_j\mathcal{V}, \quad \mathcal{V}_j - \mathcal{V}^*\ \right] = 0, \quad j = 1, \ldots, K, \tag{2.6}$$

where $\mathcal{V}^*$ is the payoff.

For computational purposes, (2.6) will be posed on the localized domain

$$(S,\tau) \in [0, S_{\max}] \times [0, T]. \tag{2.7}$$

No boundary condition is required at $S = 0$, while at $S = S_{\max}$ a Dirichlet condition is imposed (in this paper we use the payoff). The payoff condition when $\tau = 0$ is given by

$$\mathcal{V}(S,0) = \mathcal{V}^*(S). \tag{2.8}$$

We truncate any jumps which would require data outside the computational domain. The error in this approximation is small in regions of interest if $S_{\max}$ is sufficiently large [28]. More precisely, the term $\mathcal{V}_k(\xi_{jk}S,\tau)$ in (2.5) is replaced by $\mathcal{V}_k(\min(S_{\max}, \xi_{jk}S),\tau)$.

3. Discretization. In this section we describe three different discretizations of (2.6). The first method is a partially explicit method which makes use of (2.6) directly, while the other two are implicit methods. The implicit methods work with optimal control formulations, one being a direct control and the other using a penalty method.

Define a set of nodes $\{S_1, S_2, \ldots, S_{i_{\max}}\}$, and denote the nth timestep by $\tau^n = n\Delta\tau$. Let $V^n_{i,j}$ be the approximate solution of (2.6) at (S_i, τ^n), regime j, and define vectors V^n

$$V^n = [V^n_{1,1}, \ldots, V^n_{i_{\max},1}, \ldots, V^n_{1,K}, \ldots, V^n_{i_{\max},K}]' \tag{3.1}$$

of size $N = K \times i_{\max}$. It will sometimes be convenient to use a single or double subscript when referring to an entry in V^n,

$$V^n_\ell = V^n_{i,j}, \quad \ell = (j-1)i_{\max} + i, \tag{3.2}$$

which will be clear from the context. In addition, we use the notation

$$V^n_{*,j} = [V^n_{1,j}, V^n_{2,j}, \ldots, V^n_{i_{\max},j}]' \tag{3.3}$$

to denote an approximate solution for a given regime j. Let $\mathcal{L}^h_j, \mathcal{J}^h_j$ be the discrete form of the operators $\mathcal{L}_j, \mathcal{J}_j$. Our discretization can be represented as

$$(\mathcal{L}^h_j V^n)_{ij} = \alpha_{i,j} V^n_{i-1,j} + \beta_{i,j} V^n_{i+1,j} - (\alpha_{i,j} + \beta_{i,j} + r + \lambda_j) V^n_{i,j}, \tag{3.4}$$

with three point finite difference operators. A weighted average of central, forward, and backward differencing is used as described in Appendix A.

Remark 3.1 (positive coefficient discretization). Algorithm A.1 in Appendix A guarantees that the positive coefficient condition

$$\alpha_{i,j} \geq 0, \quad \beta_{i,j} \geq 0 \tag{3.5}$$

holds, with central weighting used as much as possible.

In the case of $\mathcal{J}^h_j$, we use linear interpolation for the discretization,

$$[\mathcal{J}^h_j V^n]_{i,j} = \sum_{\substack{k=1 \\ k \neq j}}^{K} \frac{\lambda_{jk}}{\lambda_j} I^h_{i,j,k} V^n, \tag{3.6}$$

where

$$\begin{aligned} I^h_{i,j,k} V^n &= w V^n_{m,k} + (1-w) V^n_{m+1,k}, \quad w \in [0,1], \\ &\simeq \mathcal{V}_k(\min(S_{\max}, \xi_{jk} S_i), \tau^n). \end{aligned} \tag{3.7}$$

Let $(\Delta S)_{\max} = \max_i(S_{i+1} - S_i)$, $(\Delta\tau)_{\max} = \max(\tau^{n+1} - \tau^n)$. The mesh and timesteps are parameterized by a discretization parameter h such that

$$(\Delta S)_{\max} = C_1 h, \quad (\Delta\tau)_{\max} = C_2 h, \tag{3.8}$$

with C_1, C_2 being positive constants. We will carry out tests letting $h \to 0$.

Observe that the discretization method is at least first order correct, and taking into account (3.6) and (3.7), note the following results. Let $\mathbf{e}$ be the $i_{\max}$ length vector $[1, 1, \ldots, 1]'$. Then, we have

$$\begin{aligned} \left[\mathcal{L}^h_j \mathbf{e}\right]_i &= -(r + \lambda_j), \quad i < i_{\max}, \\ \left[\mathcal{J}^h_j \mathbf{e}\right]_i &= 1, \quad i < i_{\max}. \end{aligned} \tag{3.9}$$

Based on these discrete operators, we consider the following three approaches.

3.1. Explicit American constraint and regime coupling. A first order in time method can be constructed using the discretization

$$\begin{aligned}\left(\frac{1}{\Delta\tau}-\mathcal{L}_j^h\right)\hat{V}_{i,j}^{n+1}&=\frac{V_{i,j}^n}{\Delta\tau}+\lambda_j[\mathcal{J}_j^hV^n]_{i,j},\quad i<i_{\max},\\ \hat{V}_{i,j}^{n+1}&=\mathcal{V}_i^*,\quad i=i_{\max},\\ V_{i,j}^{n+1}&=\max(\hat{V}_{i,j}^{n+1},\mathcal{V}_i^*).\end{aligned}\tag{3.10}$$

PROPOSITION 3.1. *If a positive coefficient method is used to form $\mathcal{L}_j^h$, and linear interpolation is used in $\mathcal{J}_j^h$, then scheme* (3.10) *is unconditionally stable.*

Proof. Writing out (3.10) for $i<i_{\max}$, noting (3.4) gives

$$\left(\frac{1}{\Delta\tau}+\alpha_{i,j}+\beta_{i,j}+r+\lambda_j\right)\hat{V}_{i,j}^{n+1}=\alpha_{i,j}\hat{V}_{i-1,j}^{n+1}+\beta_{i,j}\hat{V}_{i+1,j}^{n+1}+\frac{V_{i,j}^n}{\Delta\tau}+\lambda_j[\mathcal{J}_j^hV^n]_{i,j}.$$

Noting (3.5), (3.6), and (3.9), this then implies

$$\left(\frac{1}{\Delta\tau}+\alpha_{i,j}+\beta_{i,j}+r+\lambda_j\right)|\hat{V}_{i,j}^{n+1}|\le(\alpha_{i,j}+\beta_{i,j})\|\hat{V}_{i,j}^{n+1}\|_\infty+\left(\frac{1}{\Delta\tau}+\lambda_j\right)\|V^n\|_\infty.\tag{3.11}$$

From $\|V^0\|_\infty=\|\mathcal{V}^*\|_\infty$, a straightforward maximum analysis gives

$$\|V^{n+1}\|_\infty\le\|V^n\|_\infty.\qquad\square\tag{3.12}$$

Remark 3.2. Note that the regime coupling terms $\mathcal{J}_j^h$ in scheme (3.10) are handled explicitly; hence method (3.10) requires only solution of K decoupled tridiagonal systems in each timestep, and consequently is very inexpensive. However, we can expect that convergence as $h\to0$ will be at most at a first order rate.

3.2. Direct control discretization. Rewrite (2.6) in control form [8]

$$\max_{\varphi\in\{0,1\}}\left[\Omega\,\varphi(\mathcal{V}^*-\mathcal{V}_j)-(1-\varphi)(\mathcal{V}_{j,\tau}-\mathcal{L}_j\mathcal{V}_j-\lambda_j\mathcal{J}_j\mathcal{V})\right]=0,\tag{3.13}$$

where we have introduced a scaling factor $\Omega>0$ into (3.13). Mathematically, of course, the scaling factor does not affect the solution of (3.13). However, any iterative method will require comparing the two (in general) nonzero terms in the $\max(\cdot)$ expression. We can see that a scaling factor is required since the two terms in the $\max(\cdot)$ expression have different units.

Discretizing (3.13) gives

$$\begin{aligned}(1-\varphi_{i,j}^{n+1})&\left(\frac{V_{i,j}^{n+1}}{\Delta\tau}-\theta\mathcal{L}_j^hV_{i,j}^{n+1}\right)+\Omega\,\varphi_{i,j}^{n+1}V_{i,j}^{n+1}\\ &=(1-\varphi_{i,j}^{n+1})\frac{V_{i,j}^n}{\Delta\tau}+\Omega\,\varphi_{i,j}^{n+1}\mathcal{V}_i^*+(1-\varphi_{i,j}^{n+1})\lambda_j\theta[\mathcal{J}_j^hV^{n+1}]_{i,j}\\ &\quad+(1-\varphi_{i,j}^{n+1})(1-\theta)\left[\mathcal{L}_j^hV_{i,j}^n+\lambda_j[\mathcal{J}_j^hV^n]_{i,j}\right],\quad i<i_{\max},\\ V_{i,j}^{n+1}&=\mathcal{V}_i^*,\quad i=i_{\max},\end{aligned}\tag{3.14}$$

where

$$\{\varphi_{i,j}^{n+1}\} \in \arg\max_{\varphi\in\{0,1\}}\Big\{\Omega\ \varphi(\mathcal{V}_i^* - V_{i,j}^{n+1}) - (1-\varphi)\Big(\frac{V_{i,j}^{n+1}-V_{i,j}^{n}}{\Delta\tau} - \theta\big(\mathcal{L}_j^h V_{i,j}^{n+1} + \lambda_j[\mathcal{J}_j^h V^{n+1}]_{i,j}\big) - (1-\theta)\big(\mathcal{L}_j^h V_{i,j}^{n} + \lambda_j[\mathcal{J}_j^h V^{n}]_{i,j}\big)\Big)\Big\},$$

(3.15)

and our timestepping method is fully implicit ($\theta = 1$) or Crank–Nicolson ($\theta = 1/2$).

3.3. Penalty method. The penalized form of (2.6) [20] is

$$\mathcal{V}_{j,\tau}^{\varepsilon} = \mathcal{L}_j\mathcal{V}_j^{\varepsilon} + \lambda_j\mathcal{J}_j\mathcal{V}^{\varepsilon} + \max_{\varphi\in\{0,1\}}\left[\varphi\frac{(\mathcal{V}^* - \mathcal{V}_j^{\varepsilon})}{\varepsilon}\right]. \tag{3.16}$$

We remind the reader that the basic idea of the penalty method is to discretize (3.16) and let $\varepsilon \to 0$ as the mesh tends to zero.

Using fully implicit ($\theta = 1$) or Crank–Nicolson ($\theta = 1/2$) timestepping, the discrete form of (3.16) is then

$$\begin{aligned}\frac{V_{i,j}^{n+1}}{\Delta\tau} - \theta\mathcal{L}_j^h V_{i,j}^{n+1} + \frac{\varphi_{i,j}^{n+1}}{\varepsilon}V_{i,j}^{n+1} &= \frac{V_{i,j}^{n}}{\Delta\tau} + \frac{\varphi_{i,j}^{n+1}}{\varepsilon}\mathcal{V}_i^* + \lambda_j\theta[\mathcal{J}_j^h V^{n+1}]_{i,j} \\ &\quad + (1-\theta)\big[\mathcal{L}_j^h V_{i,j}^{n} + \lambda_j[\mathcal{J}_j^h V^{n}]_{i,j}\big], \quad i < i_{\max}, \\ V_{i,j}^{n+1} &= \mathcal{V}_i^*, \quad i = i_{\max},\end{aligned} \tag{3.17}$$

where

$$\varphi_{i,j}^{n+1} \in \arg\max_{\varphi\in\{0,1\}}\left\{\frac{\varphi}{\varepsilon}(V_{i,j}^{n+1} - \mathcal{V}_i^*)\right\}. \tag{3.18}$$

In order to ensure that this discretization is consistent, we choose

$$\varepsilon = C_3\ \Delta\tau. \tag{3.19}$$

Remark 3.3. Equation (2.6) is a special case of the more general systems of VIs considered in [13], where it is shown that VIs such as (2.6) have unique, continuous viscosity solutions. Note that the definition of a viscosity solution must be generalized for systems of PDEs [13]. It is straightforward to show, using the methods in [19], that schemes (3.10), (3.14), and (3.17) are unconditionally l_∞ stable ($\theta = 1$), monotone, and consistent, and hence converge to the viscosity solution. Of course, if $\mathcal{V}^*$ has certain smoothness properties, then smooth solutions can be expected in some cases. However, this is not the main focus of this work. We are primarily interested in efficiently solving the discretized equations.

4. Form of the discretized equations. For the explicit American method (3.10), each timestep requires only the solution of a set of linear tridiagonal systems, with no nonlinear iteration being required. However, in the case of both the direct control method (section 3.2) and the penalty method (section 3.3) we require the solution of nonlinear equations at each timestep. In these cases the nonlinear algebraic equations are of the form

$$\begin{aligned}&\mathcal{A}^*(Q)\ U = \mathcal{C}(Q) \\ &\text{with} \quad Q_\ell = \arg\max_{Q\in Z}\Big[-\mathcal{A}^*(Q)U + \mathcal{C}(Q)\Big]_\ell,\end{aligned} \tag{4.1}$$

where $\mathcal{A}^*$ is of size $N \times N$, and U, $\mathcal{C}$ are vectors of size N. Z is the set of admissible controls. Here $\mathcal{A}^*$ and $\mathcal{C}$ denote the coefficients of the associated linear systems, while Q_ℓ is the control for the ℓth node. For many of the methods that we use, it is convenient to separate $\mathcal{A}^*(Q)$ as

$$\mathcal{A}^*(Q) = \mathcal{A}(Q) - \mathcal{B}(Q), \tag{4.2}$$

with $\mathcal{A}(Q)$ providing the terms which couple only nodes within the same regime, and $\mathcal{B}(Q)$ containing all the terms which couple different regimes. The explicit formulae for $\mathcal{A}(Q), \mathcal{B}(Q), \mathcal{C}(Q)$ (and hence for $\mathcal{A}^*(Q)$) are defined in Appendix B.

Remark 4.1. It is important to note that $[\mathcal{A}]_{\ell,m}, [\mathcal{B}]_{\ell,m}, [\mathcal{C}]_\ell$ (and hence also $[\mathcal{A}^*]_{\ell,m}$) depend only on Q_ℓ.

When we separate the regime coupling terms, the following properties of $\mathcal{A}, \mathcal{B}$ become important.

PROPOSITION 4.1. *Assuming that Algorithm* A.1 *is used, discretizations* (3.14) *and* (3.17) *result in matrices* $\mathcal{A}, \mathcal{B}$ *having the following properties:*

(a) $\mathcal{B}(Q) \geq 0$.

(b) *Suppose row* ℓ *corresponds to grid node* (i, j). *Then the* ℓ*th row sums for* $\mathcal{A}(Q^k)$ *and* $\mathcal{B}(Q^k)$ *are as follows:*

Direct control:

$$\begin{aligned} \textit{Row\_Sum}_{\ell}\ (\ \mathcal{A}(Q)\ ) &= \begin{cases} (1-\varphi_\ell)\big(\frac{1}{\Delta\tau} + \theta(r+\lambda_j)\big) + \varphi_\ell \Omega, & i < i_{\max}, \\ 1, & i = i_{\max}, \end{cases} \\ \textit{Row\_Sum}_{\ell}\ (\ \mathcal{B}(Q)\ ) &= \begin{cases} (1-\varphi_\ell)\lambda_j \theta, & i < i_{\max}, \\ 0, & i = i_{\max}. \end{cases} \end{aligned} \tag{4.3}$$

Penalty method:

$$\begin{aligned} \textit{Row\_Sum}_{\ell}\ (\ \mathcal{A}(Q)\ ) &= \begin{cases} \frac{1}{\Delta\tau} + \theta(r+\lambda_j) + \frac{\varphi_\ell}{\varepsilon}, & i < i_{\max}, \\ 1, & i = i_{\max}, \end{cases} \\ \textit{Row\_Sum}_{\ell}\ (\ \mathcal{B}(Q)\ ) &= \begin{cases} \theta\lambda_j, & i < i_{\max}, \\ 0, & i = i_{\max}. \end{cases} \end{aligned} \tag{4.4}$$

(c) *The matrices* $\mathcal{A}^*(Q)$ *and* $\mathcal{A}(Q)$ *in* (B.6) *are strictly diagonally dominant M matrices* [39].

Proof. (a) follows from (3.6)–(3.7) and the definition of $\mathcal{B}(Q)$ in Appendix B. (b) follows from properties (3.9), equations (3.14) and (3.17), and Appendix B. Since a positive coefficient discretization is used, $\mathcal{A}^*(Q)$ and $\mathcal{A}(Q)$ have nonpositive offdiagonals and strictly positive rowsums, and hence they are M matrices [39]. ∎

The following proposition will be useful [23].

PROPOSITION 4.2. *Suppose* A, B *are* $N \times N$ *matrices, with* A *being a strictly diagonally dominant M matrix, and* $B \geq 0$. *Then*

$$\|A^{-1}B\|_\infty \leq \max_\ell \left\{ \frac{\sum_u B_{\ell,u}}{\sum_u A_{\ell,u}} \right\}. \tag{4.5}$$

5. Solution of the discretized equations. We consider several techniques for solution of (4.1) (often split as in (4.2)) at each timestep.

5.1. Policy iteration. Policy iteration is a standard procedure used in dynamic programming applications, and is given in Algorithm 5.1. Various options are available for solving the system $\left[\mathcal{A}^*(Q^k)\right]U^{k+1} = \mathcal{C}(Q^k)$ on line 4. For example, a direct sparse matrix method can be used (based on, e.g., minimum degree ordering), or we can use a preconditioned GMRES technique [36], or even a simple iteration based on the obvious splitting $\mathcal{A}^*(Q) = \mathcal{A}(Q) - \mathcal{B}(Q)$. If $(U^{k+1})^m$ is the mth estimate for U^{k+1}, then simple iteration is

$$\mathcal{A}(Q^k)(U^{k+1})^{m+1} = \mathcal{B}(Q^k)(U^{k+1})^m + \mathcal{C}(Q^k). \tag{5.1}$$

Algorithm 5.1 Policy iteration.

1: $U^0 =$ Initial solution vector of size N
2: **for** $k = 0, 1, 2, \ldots$ until converge **do**
3: $\quad Q_\ell^k = \arg\max_{Q_\ell \in Z} \left\{ -\mathcal{A}^*(Q)U^k + \mathcal{C}(Q) \right\}_\ell$
4: $\quad$ Solve $\mathcal{A}^*(Q^k)U^{k+1} = \mathcal{C}(Q^k)$
5: $\quad$ **if** $k > 0$ and $\max_\ell \frac{|U_\ell^{k+1} - U_\ell^k|}{\max\left[scale, |U_\ell^{k+1}|\right]} < tolerance$ **then**
6: $\quad\quad$ break from the iteration
7: $\quad$ **end if**
8: **end for**

With respect to convergence of Algorithm 5.1, it is straightforward to prove the following [19].

THEOREM 5.1 (convergence of policy iteration). *If*

(a) *the matrix $\mathcal{A}^*(Q^k)$ is an M matrix, and*
(b) *the vector $\mathcal{C}(Q)$ and the matrices $\mathcal{A}^*(Q^k)$ and $\mathcal{A}^*(Q^k)^{-1}$ are bounded independent of Q^k,*

then Algorithm 5.1 *converges to the unique solution of* (4.1).

Proof. For the convenience of the reader, we give a brief outline here, and refer the reader to [19] for details. Rearrange Algorithm 5.1 in the form

$$\mathcal{A}^*(Q^k)(U^{k+1} - U^k) = \left[-\mathcal{A}^*(Q^k)U^k + \mathcal{C}(Q^k)\right] - \left[-\mathcal{A}^*(Q^{k-1})U^k + \mathcal{C}(Q^{k-1})\right] \tag{5.2}$$

and note that since Q^k maximizes $-\mathcal{A}^*(Q)U^k + \mathcal{C}(Q)$, the right-hand side of (5.2) is nonnegative, and since $\mathcal{A}^*(Q^k)$ is an M matrix, $(U^{k+1} - U^k) \geq 0$. U^k is bounded independent of k, and hence the iterates form a bounded nondecreasing sequence. ∎

COROLLARY 5.2. *Policy iteration converges unconditionally for both the direct control* (3.14) *and penalty* (3.17) *discretizations.*

Proof. $\mathcal{A}^*(Q^k)$ is an M matrix from Proposition 4.1. The vector $\mathcal{C}(Q)$ and matrix $\mathcal{A}^*(Q)$ are easily bounded independent of Q (for fixed grid and timesteps); see the definitions of these quantities in Appendix B. From Proposition 4.2, we have that

$$\|A^*(Q)^{-1}\|_\infty \leq \max_\ell \left\{ \frac{1}{\sum_u A^*_{\ell,u}} \right\} = \max_\ell \frac{1}{Row\_Sum\ _\ell\ (\mathcal{A}(Q) - \mathcal{B}(Q))}. \tag{5.3}$$

Hence, from Proposition 4.1, we have that

$$\max_{\ell} \frac{1}{Row\_Sum\ _{\ell}\ (\mathcal{A}^*(Q))} \leq \begin{cases} \max(1, \Delta\tau, 1/\Omega) & \text{direct control,} \\ \max(1, \Delta\tau) & \text{penalty method,} \end{cases} \tag{5.4}$$

and hence $\|A^*(Q)^{-1}\|_\infty$ is bounded for fixed $\Delta\tau$ and fixed Ω. ∎

5.2. Fixed point policy iteration. In an effort to minimize the work required to solve the linear system at each iteration, a fixed point policy iteration was suggested in [23]. The approach makes use of the splitting (4.2) and is given in Algorithm 5.2.

Algorithm 5.2 Fixed point policy iteration.

$U^0 =$ Initial solution vector of size N
for $k = 0, 1, 2, \ldots$ until converge **do**
  $Q_\ell^k = \arg\max_{Q_\ell \in Z} \left\{ -\left[\mathcal{A}(Q) - \mathcal{B}(Q)\right]U^k + \mathcal{C}(Q)\right\}_\ell$
  Solve $\mathcal{A}(Q^k)U^{k+1} = \mathcal{B}(Q^k)U^k + \mathcal{C}(Q^k)$
  if $k > 0$ and $\max_\ell \frac{|U_\ell^{k+1} - U_\ell^k|}{\max\left[scale, |U_\ell^{k+1}|\right]} < tolerance$ **then**
    break from the iteration
  end if
end for

THEOREM 5.3 (convergence of fixed point policy iteration). *If the conditions required for Theorem* 5.1 *are satisfied, and, in addition,*

(a) *the matrices* $\mathcal{A}(Q)$ *and* $\|[\mathcal{A}(Q)]^{-1}\|_\infty$ *are bounded, and*
(b) *there is a constant* $C_4 < 1$ *such that*

$$\|\mathcal{A}(Q^k)^{-1}\mathcal{B}(Q^{k-1})\|_\infty \leq C_4 \quad \textit{and} \quad \|\mathcal{A}(Q^k)^{-1}\mathcal{B}(Q^k)\|_\infty \leq C_4, \tag{5.5}$$

then the fixed point policy iteration in Algorithm 5.2 *converges.*

Proof. See [23]. ∎

COROLLARY 5.4. *The fixed point policy iteration converges unconditionally for the penalty discretization* (3.17) *and converges for the direct control discretization* (3.14) *if*

$$\Omega > \theta \cdot \hat{\lambda}, \quad \textit{where} \quad \hat{\lambda} = \max_j \lambda_j. \tag{5.6}$$

Proof. Our discretization satisfies the conditions for Theorem 5.1 from Corollary 5.2. (a) can be shown using the same steps as used to bound $\mathcal{A}^*(Q)$ and $\|[\mathcal{A}^*(Q)]^{-1}\|_\infty$ in the proof of Corollary 5.2. To prove (b), consider first the direct control method. From Propositions 4.1 and 4.2,

$$\begin{aligned} \|A^{-1}(Q^k)B(Q^p)\|_\infty &\leq \max_\ell \frac{\sum_u B(Q^p)_{\ell,u}}{\sum_u A(Q^k)_{\ell,u}} \\ &= \max_{i,j} \frac{(1 - \varphi_{i,j}^p)\lambda_j\theta}{(1 - \varphi_{i,j}^k)\left(\frac{1}{\Delta\tau} + \theta(r + \lambda_j)\right) + \varphi_{i,j}^k\Omega}. \end{aligned} \tag{5.7}$$

Consequently,

$$\begin{aligned} \max_{p\in\{k,k-1\}} \|A^{-1}(Q^k)B(Q^p)\|_\infty &\leq \max_{\substack{i,j \\ p\in\{k,k-1\}}} \frac{(1 - \varphi_{i,j}^p)\lambda_j\theta}{(1 - \varphi_{i,j}^k)\left(\frac{1}{\Delta\tau} + \theta(r + \lambda_j)\right) + \varphi_{i,j}^k\Omega} \\ &\leq \max\left[\max_j \frac{\lambda_j\theta\Delta\tau}{1 + \theta(r + \lambda_j)\Delta\tau}, \ \max_j \frac{\lambda_j\theta}{\Omega}\right]. \end{aligned} \tag{5.8}$$

Consider the penalty method case. Again, recalling Propositions 4.1 and 4.2, we have

$$\begin{aligned}\|A^{-1}(Q^k)B(Q^p)\|_\infty &\le \max_\ell \frac{\sum_u B(Q^p)_{\ell,u}}{\sum_u A(Q^k)_{\ell,u}} \\ &= \max_{i,j} \frac{\theta\lambda_j}{\frac{1}{\Delta\tau} + \theta(r+\lambda_j) + \frac{\varphi^k_{i,j}}{\varepsilon}}. \end{aligned} \tag{5.9}$$

As a result (penalty case),

$$\begin{aligned}\max_{p\in\{k,k-1\}} \|A^{-1}(Q^k)B(Q^p)\|_\infty &\le \max_{\substack{i,j \\ p\in\{k,k-1\}}} \frac{\theta\lambda_j}{\frac{1}{\Delta\tau} + \theta(r+\lambda_j) + \frac{\varphi^k_{i,j}}{\varepsilon}} \\ &\le \max_j \frac{\theta\lambda_j\Delta\tau}{1+\theta(r+\lambda_j)\Delta\tau} < 1. \end{aligned} \tag{5.10}$$

∎

5.3. Local policy iteration. In [37], Salmi and Toivanen solve a single regime American pricing problem with jump diffusion by lagging the jump terms and then solving the American linear complementarity problem (LCP) (with the frozen jump terms) at each iteration. In [42], a block LCP method was suggested, with lagged regime coupling terms.

Based on the above idea, we can formulate a type of *local policy iteration*, as given in Algorithm 5.3. Note that line 3 of this algorithm requires the solution of the nonlinear local control problem with the regime coupling terms (that is, $\mathcal{B}U^k$) lagged one iteration.

Algorithm 5.3 Local policy iteration.

1: $U^0 =$ Initial solution vector of size N
2: **for** $k = 0, 1, 2, \ldots$ until converge **do**
3: Solve: $\max_{Q\in Z} \left\{ -\mathcal{A}(Q)U^{k+1} + \mathcal{B}(Q)U^k + \mathcal{C}(Q) \right\} = 0$
4: **if** converged **then**
5: break from the iteration
6: **end if**
7: **end for**

Convergence of this method was proved in [37], in the context of jump diffusions, based on special properties of the LCP form of the American pricing problem. Here we can give a more general proof of this result, one which can be applied to any control problem of the form (4.1).

THEOREM 5.5. *If $\mathcal{A}(Q)$ is an M matrix, $\mathcal{B}(Q) \ge 0$, and*

$$\max_{Q\in Z} \|\mathcal{A}(Q)^{-1}\mathcal{B}(Q)\|_\infty \le C_5 < 1, \tag{5.11}$$

then the local policy iteration (5.3) *converges. Furthermore, if U^* is the solution to* (4.1)*, and $E^k = U^k - U^*$, then*

$$\|E^{k+1}\|_\infty \le C_5 \|E^k\|_\infty. \tag{5.12}$$

Proof. If U^* is a solution to (4.1), then

$$\max_{Q'\in Z} \left\{ -\mathcal{A}(Q')U^* + \mathcal{B}(Q')U^* + \mathcal{C}(Q') \right\} = 0, \tag{5.13}$$

while from Algorithm 5.3, we have

$$\max_{Q\in Z}\Big\{-\mathcal{A}(Q)U^{k+1}+\mathcal{B}(Q)U^k+\mathcal{C}(Q)\Big\}=0. \tag{5.14}$$

Subtracting (5.13) from (5.14), we obtain

$$\begin{aligned}0&=\max_{Q\in Z}\Big\{-\mathcal{A}(Q)U^{k+1}+\mathcal{B}(Q)U^k+\mathcal{C}(Q)\Big\}\\&\quad-\max_{Q'\in Z}\Big\{-\mathcal{A}(Q')U^*+\mathcal{B}(Q')U^*+\mathcal{C}(Q')\Big\}\\&\le\max_{Q\in Z}\Big\{-\mathcal{A}(Q)(U^{k+1}-U^*)+\mathcal{B}(Q)(U^k-U^*)\Big\}.\end{aligned} \tag{5.15}$$

If $\hat{Q}$ satisfies

$$\hat{Q}\in\arg\max_{Q\in Z}\Big\{-\mathcal{A}(Q)(U^{k+1}-U^*)+\mathcal{B}(Q)(U^k-U^*)\Big\}, \tag{5.16}$$

then, from (5.15), we have

$$\mathcal{A}(\hat{Q})E^{k+1}\le\mathcal{B}(\hat{Q})E^k, \tag{5.17}$$

or, since $\mathcal{A}(Q)$ is an M matrix,

$$E^{k+1}\le\mathcal{A}(\hat{Q})^{-1}\mathcal{B}(\hat{Q})E^k\ \le C_5\|E^k\|_\infty\mathbf{e}, \tag{5.18}$$

where $\mathbf{e}=[1,1,\ldots,1]'$. Similarly,

$$\begin{aligned}0&=\max_{Q\in Z}\Big\{-\mathcal{A}(Q)U^{k+1}+\mathcal{B}(Q)U^k+\mathcal{C}(Q)\Big\}\\&\quad-\max_{Q'\in Z}\Big\{-\mathcal{A}(Q')U^*+\mathcal{B}(Q')U^*+\mathcal{C}(Q')\Big\}\\&\ge\min_{Q\in Z}\Big\{-\mathcal{A}(Q)(U^{k+1}-U^*)+\mathcal{B}(Q)(U^k-U^*)\Big\}.\end{aligned} \tag{5.19}$$

Hence if

$$\bar{Q}\in\arg\min_{Q\in Z}\Big\{-\mathcal{A}(Q)(U^{k+1}-U^*)+\mathcal{B}(Q)(U^k-U^*)\Big\}, \tag{5.20}$$

then

$$E^{k+1}\ge\mathcal{A}(\bar{Q})^{-1}\mathcal{B}(\bar{Q})E^k\ge-C_5\|E^k\|_\infty\mathbf{e}. \tag{5.21}$$

Equations (5.18) and (5.21) then give result (5.12). ∎

COROLLARY 5.6. *Local policy iteration for* (3.14) *and* (3.17) *converges at the rate*

$$\frac{\|E^{k+1}\|_\infty}{\|E^k\|_\infty}\le\frac{\theta\hat{\lambda}\Delta\tau}{1+\theta(r+\hat{\lambda})\Delta\tau},\quad \textit{where } \hat{\lambda}=\max_j\lambda_j. \tag{5.22}$$

Proof. From Proposition 4.1, discretizations (3.14) and (3.17) ensure that $\mathcal{A}(Q)$ is an M matrix and that $\mathcal{B}(Q)\ge 0$.

For the direct control method, setting $p = k$ in (5.7),

$$\|A^{-1}(Q^k)B(Q^k)\|_\infty \le \max_j \frac{\theta\lambda_j\Delta\tau}{1+\theta(r+\lambda_j)\Delta\tau}. \tag{5.23}$$

For the penalty method, from (5.9),

$$\|A^{-1}(Q^k)B(Q^k)\|_\infty \le \max_j \frac{\theta\lambda_j\Delta\tau}{1+\theta(r+\lambda_j)\Delta\tau}. \tag{5.24}$$ ∎

Remark 5.1. Note that a sufficient condition for the convergence of simple iteration (5.1) is $\|\mathcal{A}(Q^k)^{-1}\mathcal{B}(Q^k)\|_\infty < 1$. Consequently, since $\|\mathcal{A}(Q^k)^{-1}\mathcal{B}(Q^k)\|_\infty < 1$ unconditionally for both the penalty and direct control methods, then simple iteration always converges.

5.4. Global-in-time iteration. In [31] a method was suggested whereby the regime coupling terms are frozen and the entire solution is obtained (over all timesteps) with these frozen terms. The regime coupling terms are then updated, and the entire solution (for all timesteps) is generated again. This is repeated until convergence is obtained. A similar idea was suggested for American options with jump diffusion [5] and for Asian options under jump diffusion [6]. Effectively, a sequence of optimal stopping problems is solved. This approach is also popular for impulse control problems [33].

Let $(V^n)^k$ be the kth iterate for the solution at timestep n. The global-in-time iteration can then be described as in Algorithm 5.4.

Algorithm 5.4 Global-in-time iteration.

```
(V^n)^0 = payoff;  n = 0,...,L;  j = 1,...,K;  i = 1,...,i_max;  Δτ = T/L
for k = 0,1,2,... until converge do
   for n = 1,2,...,L do
      Solve: max_{Q∈Z} [ − A(Q)(V^n)^{k+1} + B(Q)(V^n)^k + C(Q) ] = 0
   end for
   if converged then
      break from the iteration
   end if
end for
```

We can rewrite Algorithm 5.4 into a form which resembles Algorithm 5.3 (local policy iteration) as follows. Define the $N(L+1)$ length vectors

$$\mathbb{V} = [(V^0)', \ldots, (V^L)']', \tag{5.25}$$

so that $\mathbb{V}$ contains the solution at each node and regime for all timesteps. Similarly, $\mathbb{Q}$ contains the controls at each node and regime for all timesteps. It will be convenient to refer to the entries in $\mathbb{V}$ using a single or triple index, depending on the context:

$$\begin{aligned} \mathbb{V}_\ell &= \mathbb{V}_{i,j,n}, \quad \ell = n\ i_{\max}K + (j-1)i_{\max} + i, \\ &= V^n_{i,j}. \end{aligned} \tag{5.26}$$

We will also use the notation

$$\mathbb{V}_{*,*,n} = V^n \tag{5.27}$$

to refer to the N length subvector of $\mathbb{V}$ which refers to nodes and regimes associated with a single time τ^n.

Define $N(L+1) \times N(L+1)$ matrices $\mathbb{A}, \mathbb{B}$ and $N(L+1)$ length vector $\mathbb{C}$ as in Appendix C. Let $\mathbb{V}^k$ be the kth iterate for $\mathbb{V}$. Algorithm 5.4 can now be rewritten as in Algorithm 5.5.

Algorithm 5.5 Global-in-time iteration: rewritten.

```
1: V^0 = payoff;  n = 0, ..., L;  j = 1, ..., K;  i = 1, ..., i_max;  Δτ = T/L
2: for k = 0, 1, 2, ... until converge do
3:    Solve: max_{Q∈Z} [ − A(Q)V^{k+1} + B(Q)V^k + C(Q) ] = 0
4:    if converged then
5:       break from the iteration
6:    end if
7: end for
```

Algorithm 5.5 is now identical to Algorithm 5.3, and hence we can apply Theorem 5.5 to obtain

$$\frac{\|\mathbb{V}^{k+1} - \mathbb{V}^{\infty}\|_{\infty}}{\|\mathbb{V}^{k} - \mathbb{V}^{\infty}\|_{\infty}} \leq \|\mathbb{A}^{-1}\mathbb{B}\|_{\infty}. \tag{5.28}$$

In Appendix C we obtain the bound for the direct control formulation ($\theta = 1$, fully implicit case)

$$\|\mathbb{A}^{-1}\mathbb{B}\|_{\infty} \leq \left[1 - \frac{1}{[1 + \Delta\tau(\hat{\lambda} + r)]^L}\right]\left(\frac{\hat{\lambda}}{\hat{\lambda} + r}\right), \tag{5.29}$$

with $\hat{\lambda} = \max\ \lambda_j$. Note that since $L\Delta\tau = T$, we have

$$\begin{aligned} \frac{\hat{\lambda}\Delta\tau}{1 + (r + \hat{\lambda})\Delta\tau} &\leq \frac{\hat{\lambda} L \Delta\tau}{1 + (r + \hat{\lambda}) L \Delta\tau} \\ &= \frac{\hat{\lambda} T}{1 + (r + \hat{\lambda}) T} \\ &\leq \left[1 - \frac{1}{[1 + \Delta\tau(\hat{\lambda} + r)]^L}\right]\left(\frac{\hat{\lambda}}{\hat{\lambda} + r}\right). \end{aligned} \tag{5.30}$$

Consequently, in terms of provable bounds, the convergence rate of the global-in-time iteration is considerably worse than that for local policy iteration (5.22).

If $k_{\max}$ iterations of the global-in-time algorithm is required to meet the convergence tolerance, then this is equivalent to the same work as required to determine a complete solution with local policy iteration, where on average $k_{\max}$ local policy iterations are required in each step. The larger bound on the convergence rate for global-in-time iteration suggests that we can expect the total cost of the global-in-time iteration to be larger than local policy iteration. Indeed, note that the number of iterations per timestep for the local policy iteration must tend to unity as $\Delta\tau \to 0$, independent of the mesh size (from (5.23)–(5.24)). It is, of course, not possible to do better than this.

Assuming that $L = O(N)$, the global-in-time method requires $O(N^2)$ storage, since we have to store the entire solution for all timesteps, at each iteration, compared to $O(N)$ storage for local policy iteration.

As a result of the convergence bounds and storage inefficiencies, we will not study this method further.

6. Numerical results. For our example in this section we use a three regime model, shown in (6.1). The numerical tests use the transition probability array $\boldsymbol{\lambda}$, jump amplitudes $\boldsymbol{\xi}$, and volatilities $\boldsymbol{\sigma}$ given in (6.1). Other data are given in Table 6.1.

$$\boldsymbol{\lambda} = \begin{bmatrix} -3.2 & 0.2 & 3.0 \\ 1.0 & -1.08 & .08 \\ 3.0 & 0.2 & -3.2 \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{bmatrix} 1.0 & 0.90 & 1.1 \\ 1.2 & 1.0 & 1.3 \\ 0.95 & 0.8 & 1.0 \end{bmatrix}, \quad \boldsymbol{\sigma} = \begin{bmatrix} .2 \\ .15 \\ .30 \end{bmatrix}. \tag{6.1}$$

We consider two payoffs: a put option with payoff $V^* = \max(K - S, 0)$, and an American butterfly with payoff

$$V^* = \max(S - K_1, 0) - 2\max(S - (K_1 + K_2)/2, 0) + \max(S - K_2, 0). \tag{6.2}$$

We assume the existence of an American contract with payoff (6.2), which can only be early exercised as a unit. This contract has been used as a severe test case by several authors [1, 41, 32].

The variable timestep selector described in [16] is used. This problem is solved on a sequence of (unequally spaced) grids. At each grid refinement, a new fine grid node is inserted between each two coarse grid nodes, and the timestep control parameter is halved. Table 6.2 shows the number of nodes, variables, and timesteps, for various levels of grid refinement, for both American put and American butterfly examples.

6.1. Explicit American constraint and regime coupling. We first compare the explicit coupling scheme (3.10) with a fully implicit method ($\theta = 1$) and a Crank–Nicolson method ($\theta = 1/2$), using the fixed point policy iteration of section 5.2. The

TABLE 6.1
Data for the regime switching, American problem.

Expiry time	.50
Exercise	American
Strike (put) K	100
Butterfly parameters K_1, K_2	90, 110
Risk-free rate r	.02
Penalty parameter ε	$10^{-6}\Delta\tau$
Scale factor Ω	$1/\varepsilon$
$S_{\max}$	5000
Convergence tolerance (e.g., (5.1))	10^{-8}

TABLE 6.2
Grid/timestep data for convergence study, regime switching example. On each grid refinement, new fine grids are inserted between each two coarse grid nodes, and the timestep control parameter is halved.

Refine	S nodes	Timesteps (put)	Timesteps (butterfly)	Unknowns
0	51	34	34	153
1	101	66	67	303
2	201	130	132	603
3	401	256	261	1203
4	801	507	519	2403
5	1601	1010	1033	4803
6	3201	2015	2062	9603
7	6401	4023	4118	19203

TABLE 6.3

Comparison of various timestepping methods. Value at $t = 0$, $S = 100$, regime 1. Data in Table 6.1 and in (6.1). Grid data in Table 6.2. Explicit coupling refers to scheme (3.10). Ratio is the ratio of successive changes as the grid is refined. Put payoff.

	Explicit coupling		Fully implicit		Crank–Nicolson	
Refine	Value	Ratio	Value	Ratio	Value	Ratio
0	7.255090541	N/A	7.554014817	N/A	7.618940039	N/A
1	7.431866668	N/A	7.586363330	N/A	7.618460248	N/A
2	7.524864056	2.0	7.602663717	1.98	7.618359301	4.7
3	7.571017345	1.97	7.610447983	2.1	7.618341970	5.28
4	7.594451442	1.95	7.614390087	2.0	7.618334755	2.4
5	7.606402700	2.0	7.616354573	1.95	7.618333108	4.4
6	7.612363761	2.0	7.617344461	2.02	7.618332684	3.9
7	7.615345680	2.0	7.617838205	2.0	7.618332568	3.7

TABLE 6.4

Comparison of various timestepping methods. Value at $t = 0$, $S = 93$, regime 2. Data in Table 6.1 and in (6.1). Grid data in Table 6.2. Explicit coupling refers to scheme (3.10). Ratio is the ratio of successive changes as the grid is refined. Butterfly payoff.

	Explicit coupling		Fully implicit		Crank–Nicolson	
Refine	Value	Ratio	Value	Ratio	Value	Ratio
0	3.916837172	N/A	4.408997074	N/A	4.444203298	N/A
1	4.159434148	N/A	4.435239516	N/A	4.452662566	N/A
2	4.281954975	1.98	4.435239516	1.8	4.458280993	1.5
3	4.351246652	1.8	4.455493332	2.4	4.459866276	3.5
4	4.391669077	1.7	4.458045879	2.3	4.460228635	4.4
5	4.415803618	1.7	4.459228339	2.2	4.460321583	3.9
6	4.430834006	1.6	4.459799179	2.1	4.460345221	3.9
7	4.440487206	1.55	4.460078178	2.04	4.460351242	3.9

Crank–Nicolson method uses the standard Rannacher timestepping [35] modification; that is, two fully implicit steps are used at the beginning, and Crank–Nicolson is used thereafter.

The put payoff results are shown in Table 6.3, and the butterfly results are given in Table 6.4. Figure 6.1 shows the value of the American butterfly at $t = 0$. The Crank–Nicolson method requires $3 - 5$ iterations per timestep (more details are given in later sections), and hence a Crank–Nicolson solution is about $3 - 5$ times more expensive than the explicit coupling method (3.10) at the same grid refinement level. Taking into account accuracy requirements, one can see that the implicit coupling methods are much more efficient than scheme (3.10), unless the requirements are very low. In these tables, ratio refers to the ratio of successive changes in the solution as the grid/timesteps are refined. A ratio of four would indicate quadratic convergence, and a ratio of two would indicate linear convergence. Note that Table 6.4 indicates sublinear convergence for the explicit coupling method (butterfly payoff).

6.2. Full policy iteration. In the case of regime switching, an obvious candidate method is policy iteration as in Algorithm 5.1. Each iteration requires solution of the sparse matrix $(\mathcal{A}-\mathcal{B})$. This matrix has a block tridiagonal structure with extra nonzero entries due to the regime coupling terms in $\mathcal{B}$. Note that the incidence matrix is no longer symmetric. However, one might imagine that modern sparse matrix solvers would be able to efficiently solve this linear system.

Table 6.5 shows the number of nonzeros in the factors of $(\mathcal{A} - \mathcal{B})$ at level five grid refinement. Since the structure of $\mathcal{A} - \mathcal{B}$ is nonsymmetric, we use a minimum degree ordering based on the structure of $(\mathcal{A} - \mathcal{B}) + (\mathcal{A} - \mathcal{B})'$. The actual symbolic

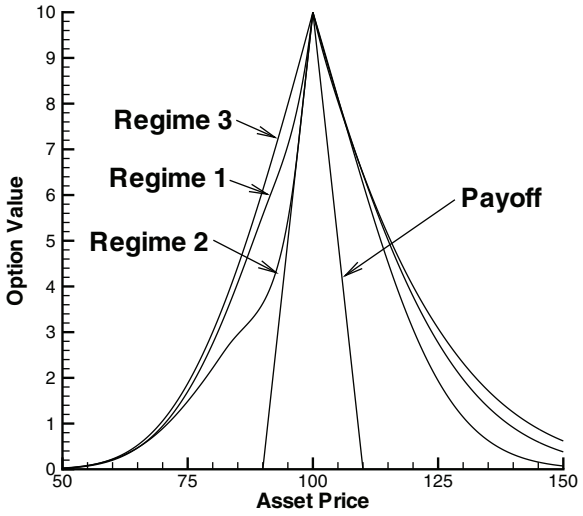

FIG. 6.1. *American butterfly, regime switching. Data in Table* 6.1 *and in* (6.1).

TABLE 6.5
Nonzeros in factors, direct solve of $(\mathcal{A}-\mathcal{B})$*, level five. Minimum degree ordering used.*

$\boldsymbol{\xi}$ in (6.1)	$\xi_{i,j} = 1.0$ $\forall i,j$	Number of unknowns
282860	33609	4803

TABLE 6.6
Comparison of full policy iteration, Algorithm 5.1*, using a direct solve, full policy iteration with an iterative solution (GMRES), full policy iteration with simple iteration* (5.1)*, and fixed point policy iteration, Algorithm* 5.2*, grid refinement level five. Regime switching, American option, penalty formulation, put payoff. All methods used the same number of timesteps. Crank–Nicolson timestepping used. Data in Table* 6.1 *and in* (6.1).

Linear solution method	Outer iterations per step	Inner iterations per outer iteration	CPU time (normalized)
	Full policy iteration, Algorithm 5.1		
Direct (min degree)	2.4	N/A	48.50
GMRES (ILU(0))[36]	2.4	1.91	4.85
Simple iteration (5.1)	2.4	2.06	1.53
	Fixed point policy iteration, Algorithm 5.2		
Direct (tridiagonal)	3.22	N/A	1.0

factorization is carried out using the structure of $(\mathcal{A}-\mathcal{B})$. The number of nonzeros in the factors is highly sensitive to the data in the jump size matrix $\boldsymbol{\xi}$. For comparison, we also computed the number of nonzeros in $(\mathcal{A}-\mathcal{B})$ for the case when we set all the jump sizes to one (see Table 6.5).

Table 6.6 gives the normalized CPU time for a complete solution (grid refinement level five) using various methods for solution of the sparse matrix (full policy iteration) compared with a fixed point policy iteration solution. In the case that an iterative method was used to solve the full policy matrix, the inner convergence tolerance was as given in Table 6.1. It is clear that use of full policy iteration for this problem is not efficient, primarily due to the cost of the matrix solve.

TABLE 6.7

Number of fixed point policy iterations per timestep. Penalty refers to (3.17). *Direct control refers to* (3.14). *All methods used the same total number of timesteps. Crank–Nicolson timestepping used. Data in Table* 6.1 *and in* (6.1). *Grid data in Table* 6.2. *American put. Fixed point policy iteration as in Algorithm* 5.2.

Refinement	Direct control (section 3.2)			Penalty
	$\Omega = 100$	$\Omega = 10^4$	$\Omega = 10^6/(\Delta\tau)$	(section 3.3)
0	5.40	5.40	5.40	5.40
1	4.75	4.75	4.75	4.75
2	4.25	4.25	4.25	4.25
3	3.99	3.75	3.75	3.75
4	3.97	3.70	3.50	3.55
5	4.12	3.75	3.17	3.22
6	4.65	4.26	3.00	3.04
7	6.48	5.19	3.00	3.03

6.3. Fixed point policy iteration. In this section, we will examine some of the issues arising in the use of fixed point policy iteration, as described in Algorithm 5.2. Both the direct control method (section 3.2) and the penalty formulation (section 3.3) will be considered.

Table 6.7 shows a comparison of the penalty formulation (3.17) with the direct control formulation (3.14) for various choices of the scaling factor Ω. The penalty parameter was $\varepsilon = 10^{-6}\Delta\tau$. All methods used the same timestep sequence, and the solutions agreed to eight digits. All choices of the scaling factor satisfied the convergence condition (5.6).

Examination of the iterates for the direct control method showed that at small grid sizes, the iteration appeared to have difficulty determining where the exercise boundary was located. This was due to the fact that there were many nodes which had values very close to the payoff value. Consequently, it appeared to be desirable to increase the size of Ω as the grid is refined.

As a result, a natural choice for Ω is the same form as used for $1/\varepsilon$, that is,

$$\Omega = \frac{1}{\varepsilon} = \frac{C}{\Delta\tau}. \tag{6.3}$$

In this case C is a dimensionless constant. With this form for Ω, both terms in the $\max(\cdot)$ expression in (3.13) have the same units.

Table 6.7 shows that form (6.3) is a good choice for the direct control formulation for very fine grids. Note that the form (6.3) for the penalty method was suggested in [20]. This form of the penalty term guarantees that the discretization of the penalty formulation (3.16) is consistent as $h \to 0$. Note that consistency holds for any $C > 0$ [7].

Table 6.8 shows the effect of the choice of the dimensionless constant C in (6.3). An American put was used, with grid refinement level five. For large values of the scaling factor Ω (or equivalently, $1/\varepsilon$), one might suspect that the iteration may no longer converge, due to floating point precision problems. This will be a result of subtracting two nearly equal floating point values in both algorithms.

Using the methods in [24], we can estimate the largest values of C which can be used before round off prevents convergence. For both penalty and direct control formulations, the estimate for this maximum value of C (designated by $C_{\max}$) which can be used in finite precision arithmetic is

$$C_{\max} \simeq \frac{tolerance}{2\delta}, \tag{6.4}$$

TABLE 6.8

Value of the American put, $t = 0$, $S = 100$. Penalty refers to (3.17). *Direct control refers to* (3.14). *All methods used the same total number of timesteps. Crank–Nicolson timestepping used. Data in Table* 6.1 *and in* (6.1). *Grid data in Table* 6.2. *Fixed point policy iteration as in Algorithm* 5.2. $***$ *indicates algorithm failed to converge after* 300 *iterations in any timestep. Level five grid refinement.*

Ω or $1/\varepsilon$	Direct control	Penalty
$10^9/(\Delta\tau)$	***	***
$10^8/(\Delta\tau)$	7.618333108	7.618333108
$10^7/(\Delta\tau)$	7.618333108	7.618333108
$10^6/(\Delta\tau)$	7.618333108	7.618333107
$10^5/(\Delta\tau)$	7.618333108	7.618333106
$10^4/(\Delta\tau)$	7.618333108	7.618333088
$10^3/(\Delta\tau)$	7.618333108	7.618332912
$10^2/(\Delta\tau)$	7.618333108	7.618331174
$10^1/(\Delta\tau)$	7.618333108	7.618314664
$1/(\Delta\tau)$	7.618333108	7.618144290
...	...	...
$10^{-6}/(\Delta\tau)$	7.618333108	
$10^{-7}/(\Delta\tau)$	***	

where *tolerance* is the convergence tolerance in Algorithms 5.1 and 5.2, and δ is the unit roundoff. In our case, we have *tolerance* $= 10^{-8}$ and in double precision $\delta \simeq 10^{-16}$, hence $C_{\max} \simeq 10^8$. This is a conservative estimate, as can be seen in Table 6.8.

On the other hand, from (5.6), we can see that if C is too small, then the direct control fixed point policy iteration is not guaranteed to converge. Equation (5.6) suggests that (for this problem) $C_{\min} \simeq 10^{-3}$. Again, this would appear to be a conservative estimate (see Table 6.8).

We remind the reader that the penalty method will converge for any $C > 0$ as $h \to 0$. However, if C is too small, then this will affect the number of correct digits for any finite h. From Table 6.3, we can see that about six digits are correct for level five grid refinement. This indicates that for the penalty method, the usable range of $C = [10^2, 10^8]$ (see Table 6.8). Based on many years of experience with penalty methods [45], we have found that it is safe to use a penalty constant C two orders of magnitude less than $C_{\max}$ estimated from (6.4). This value also minimizes errors at any finite value of h. Defining a *practical range* of C to be values which give accuracy at about the level of the discretization error, and which are two orders of magnitude less than $C_{\max}$, means that for this problem, we have a practical range of $C = [10^2, 10^6]$, which is much smaller than the practical range for C for the direct control formulation.

6.4. Local policy iteration. Table 6.9 compares the statistics for a solution of the American butterfly and American put, obtained using both the fixed point policy iteration (Algorithm 5.2) and local policy iteration (Algorithm 5.3). The local American problem was formulated with the penalty approach.

Note that the average number of outer iterations per timestep for the local policy iteration is almost the same as the average number of fixed point policy iterations per timestep. This suggests that there is not much point in solving the local policy iteration to convergence at each outer iteration. The main source of error in the iteration appears to be the regime coupling, which requires about three iterations to resolve, which is roughly what one would expect in this case from estimate (5.22).

In general, even for a tridiagonal LCP, an iterative method is required [14] for the local American problem (line 3 in Algorithm 5.3). In the special case of a sim-

TABLE 6.9

Comparison of local policy iteration (Algorithm 5.3*) and fixed point-policy iteration (Algorithm* 5.2*). Inner iterations refers to the average number of iterations required to solve the local American problems. Outer iteration refers to the number of iterations required to resolve the regime coupling. All methods used the same total number of timesteps. Crank–Nicolson timestepping used. Data in Table* 6.1 *and in* (6.1)*. Grid data in Table* 6.2*. Refinement level five.*

Method	Outer iterations per timestep	Inner iterations per outer iteration	Normalized CPU time
American butterfly			
Fixed point policy	3.23	N/A	1.0
Local policy	3.20	1.75	1.44
American put			
Fixed point policy	3.17	N/A	1.0
Local policy	3.16	1.73	1.41

ple put or call, only a single iteration is necessary [10], since the exercise region is simply connected to the boundary. Consequently, for a simple put or call, it would always be more efficient to use the direct Brennan–Schwartz method [10] to solve the local American problem, as in [37], for the local policy iteration. However, the standard Brennan–Schwartz algorithm [10] cannot be directly applied to the American butterfly.

Since the number of outer iterations for the local policy iteration is almost identical to the number of fixed point-policy iterations, use of local policy iteration coupled with the Brennan–Schwartz [10] (for the put case) method would not result in significant savings compared to the fixed point policy iteration. Since the fixed point policy iteration makes no assumptions about the form of the payoff, this would appear to indicate that the fixed point policy iteration is a good general purpose method.

Note that one might expect that the ratio of CPU times in Table 6.9 would be roughly the same as the average number of inner iterations per timestep (1.75–1.73). The actual CPU time ratio (1.44–1.41) is somewhat less. This is simply because each inner penalty iteration is extremely efficient (a tridiagonal solve, followed by a simple comparison test). The outer iteration requires more complex data structure manipulation. Consequently, the actual ratios of CPU times will be highly implementation specific.

7. Conclusions. We have analyzed several methods for pricing American options under a regime switching stochastic process. By formulating this problem as an abstract optimal control problem, we can obtain some previously known results, for some special cases, very simply. However, using our general framework, it is trivial to extend these results to other control problems and stochastic processes. For example, all the analysis presented here can be applied to Markov modulated jump diffusions [18], provided that the integral terms are discretized in the usual fashion [17]. In addition, these techniques can also be applied to switching problems [34].

Our analysis and numerical tests indicate that Crank–Nicolson timestepping, combined with a fixed point policy iteration, using a direct control formulation, is an effective and robust method for solution of American option problems in a regime switching context.

Appendix A. Discrete equation coefficients. The discrete coefficients in equation (3.4) are given in the following. We use standard three point operators for the first and second derivatives.

Central differencing:

$$\alpha_{i,j}^{cent} = \left[\frac{(\sigma^j)^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r - \rho_j) S_i}{S_{i+1} - S_{i-1}} \right],$$

$$\beta_{i,j}^{cent} = \left[\frac{(\sigma^j)^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r - \rho_j) S_i}{S_{i+1} - S_{i-1}} \right]. \tag{A.1}$$

Forward/backward differencing (upstream):

$$\alpha_{i,j}^{ups} = \left[\frac{(\sigma^j)^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} + \max\left(0, \frac{-(r - \rho_j) S_i}{S_i - S_{i-1}}\right) \right],$$

$$\beta_{i,j}^{ups} = \left[\frac{(\sigma^j)^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \max\left(0, \frac{(r - \rho_j) S_i}{S_{i+1} - S_i}\right) \right]. \tag{A.2}$$

A weighted average of central and upstream differencing is used (see Algorithm A.1). The weighting is determined on a node by node basis. This guarantees that central differencing is used as much as possible, while guaranteeing that the condition (3.5) holds.

Algorithm A.1 Differencing method.

for $i = 1, 2, \ldots$ **do**
  $\omega = 1$
  if $\alpha_{i,j}^{cent} < 0$ **then**
    $\omega = \frac{\alpha_{i,j}^{ups}}{\alpha_{i,j}^{ups} - \alpha_{i,j}^{cent}}$
  else
    if $\beta_{i,j}^{cent} < 0$ **then**
      $\omega = \frac{\beta_{i,j}^{ups}}{\beta_{i,j}^{ups} - \beta_{i,j}^{cent}}$
    end if
  end if
  $\alpha_{i,j} = \omega \cdot \alpha_{i,j}^{cent} + (1 - \omega) \cdot \alpha_{i,j}^{ups}, \qquad \beta_{i,j} = \omega \cdot \beta_{i,j}^{cent} + (1 - \omega) \cdot \beta_{i,j}^{ups}$
end for

In order to get some idea of when upstream differencing would be used, it is instructive to consider a simple case. Suppose that constant grid spacing is used with $S_{i+1} - S_i = S_i - S_{i-1} = \Delta S$, with $S_i = i \Delta S$. Then, the condition

$$(\sigma^j)^2 \geq \frac{|r - \rho_j|}{i} \tag{A.3}$$

is required to ensure that $\alpha_{i,j}^{cent} \geq 0$, $\beta_{i,j}^{cent} \geq 0$. For the examples used in this paper, $|r - \rho_j| < 0.3$ and $\sigma^2 > 0.0225$. This suggests that upstream weighting would be used for nodes $i = 1, \ldots, i^*$, with $i^* < 15$. Note that this means that the upstream nodes are confined to a region near $S = 0$, which shrinks as the mesh is refined (since condition (A.3) is independent of the mesh spacing). In addition, since Algorithm A.1 uses a weighted average of central and upstream differencing, the first order error term will generally have a coefficient less than unity. Since upstream weighting is used for points remote from the areas of the mesh normally of interest, we can expect close to second order convergence at the interesting nodes, which is what we see in practice.

Appendix B. Matrix form: Discrete equations. In this section we define matrices $\mathcal{A}, \mathcal{B}$ and vector $\mathcal{C}$ to represent the discrete equations in sections 3.2 and 3.3.

Let U and Q be the vectors

$$\begin{aligned} U &= [U_{1,1}, \ldots, U_{i_{\max},1}, \ldots, U_{1,j_{\max}}, \ldots, U_{i_{\max},j_{\max}}]', \\ Q &= [\varphi_{1,1}, \ldots, \varphi_{i_{\max},1}, \ldots, \varphi_{1,j_{\max}}, \ldots, \varphi_{i_{\max},j_{\max}}]', \end{aligned} \tag{B.1}$$

and let ℓ be a row index corresponding to grid node (i, j), i.e., $\ell = (j-1) * i_{\max} + i$. Then we can write discrete equations (3.14) as follows.

B.1. Direct control. The discretized equations (3.14) can then be written in terms of matrices $\mathcal{A}, \mathcal{B}$ and vector $\mathcal{C}$ defined as $(i < i_{\max})$

$$\begin{aligned} [\mathcal{A}(Q)U]_\ell &= [\mathcal{A}U]_\ell = (1-\varphi_\ell)\left(\frac{U_\ell}{\Delta\tau} - \theta\mathcal{L}_j^h U_\ell\right) + \varphi_\ell \Omega\, U_\ell, \\ [\mathcal{B}(Q)U]_\ell &= [\mathcal{B}U]_\ell = (1-\varphi_\ell)\lambda_j\theta[\mathcal{J}_j^h U]_\ell, \\ [\mathcal{C}(Q)]_\ell &= C_\ell = (1-\varphi_\ell)\frac{V_\ell^n}{\Delta\tau} + \varphi_\ell \Omega\, \mathcal{V}_i^* \\ &\qquad + (1-\varphi_\ell)(1-\theta)\big[\mathcal{L}_j^h V_\ell^n + \lambda_j[\mathcal{J}_j^h V^n]_\ell\big]. \end{aligned} \tag{B.2}$$

B.2. Penalty method. Equation (3.17) can be written in terms of $\mathcal{A}, \mathcal{B}$ and vector $\mathcal{C}$ defined as $(i < i_{\max})$

$$\begin{aligned} [\mathcal{A}(Q)U]_\ell &= [\mathcal{A}U]_\ell = \frac{U_\ell}{\Delta\tau} - \theta\mathcal{L}_j^h U_\ell + \frac{\varphi_\ell}{\varepsilon} U_\ell, \\ [\mathcal{B}(Q)U]_\ell &= [\mathcal{B}U]_\ell = \lambda_j\theta[\mathcal{J}_j^h U]_\ell, \\ [\mathcal{C}(Q)]_\ell &= \mathcal{C}_\ell = \frac{V_\ell^n}{\Delta\tau} + \frac{\varphi_\ell}{\varepsilon}\mathcal{V}_i^* \\ &\qquad + (1-\theta)\big[\mathcal{L}_j^h V_\ell^n + \lambda_j[\mathcal{J}_j^h V^n]_\ell\big]. \end{aligned} \tag{B.3}$$

B.3. Dirichlet condition. At $i = i_{\max}$, we define (for both discretizations)

$$[\mathcal{A}U]_\ell = U_{i_{\max},j}, \quad [\mathcal{B}U]_\ell = 0, \quad \mathcal{C}_\ell = \mathcal{V}_{i_{\max}}^*, \quad \ell = (j-1)i_{\max} + i_{\max}. \tag{B.4}$$

B.4. General form. Define a vector of controls Q as in (B.1), with $q_\ell = \varphi_\ell$, with admissible controls Z,

$$Z = \big\{\varphi \;\mid\; \varphi \in \{0, 1\}\big\}. \tag{B.5}$$

The final discretized equations (3.14) and (3.17) can then be written as

$$\sup_{Q \in Z}\Big\{-\mathcal{A}(Q)V^{n+1} + \mathcal{B}(Q)V^{n+1} + \mathcal{C}(Q)\Big\} = 0. \tag{B.6}$$

Appendix C. Matrix form of global-in-time equations (in Algorithm 5.5). We restrict our attention to the case of fully implicit timestepping $\theta = 1$ and a direct control formulation. Bearing in mind the subscripting conventions in (5.26), the discretized equations in Algorithm 5.5 can be written in terms of $\mathcal{A}, \mathcal{B}$ of the direct

control discretization (B.2),

$$[\mathbb{A}(\mathbb{Q})\mathbb{V}]_\ell = [\mathbb{A}\mathbb{V}]_\ell = \begin{cases} [\mathcal{A}(\mathbb{Q}_{*,*,n})\mathbb{V}_{*,*,n}]_\ell - (1-\varphi_{i,j,n})\frac{\mathbb{V}_{i,j,n-1}}{\Delta\tau}, & i < i_{\max}, \quad n > 0, \\ \mathbb{V}_{i,j,n}, & i = i_{\max} \text{ or } n = 0, \end{cases}$$

$$[\mathbb{B}(\mathbb{Q})\mathbb{V}]_\ell = [\mathbb{B}\mathbb{V}]_\ell = \begin{cases} [\mathcal{B}(\mathbb{Q}_{*,*,n})\mathbb{V}_{*,*,n}]_\ell, & i < i_{\max}, \quad n > 0, \\ 0, & i = i_{\max} \text{ or } n = 0, \end{cases}$$

$$[\mathbb{C}(\mathbb{Q})]_\ell = \mathbb{C}_\ell = \begin{cases} \varphi_{i,j,n}\Omega\ V_i^*, & i < i_{\max}, \quad n > 0, \\ \mathcal{V}_i^*, & i = i_{\max} \text{ or } n = 0. \end{cases} \tag{C.1}$$

It follows from Propositions 4.1 and 4.2 that $\mathbb{B} \geq 0$, $\mathbb{A}$ is an M matrix, and that

$$\|\mathbb{A}^{-1}\mathbb{B}\|_\infty \leq \frac{\hat{\lambda}}{\hat{\lambda}+r}. \tag{C.2}$$

However, we can obtain a sharper bound. Let Y, Z be $N(L+1)$ length vectors, with Z arbitrary, and $\mathbb{A}Y = \mathbb{B}Z$. Then

$$\frac{\|Y\|_\infty}{\|Z\|_\infty} = \|\mathbb{A}^{-1}\mathbb{B}\|_\infty. \tag{C.3}$$

From (C.1), noting Proposition 4.1 and using the facts that $\mathbb{B} \geq 0$ and $\mathbb{A}$ is an M matrix give

$$\begin{aligned}[(1-\varphi_{i,j,n})(1+(r+\lambda_j)\Delta\tau) + \Omega\varphi_{i,j,n}]\,\|Y_{*,j,n}\|_\infty \leq \|Y_{*,*,n-1}\|_\infty(1-\varphi_{i,j,n}) \\ + (1-\varphi_{i,j,n})\lambda_j\Delta\tau\|Z\|_\infty.\end{aligned} \tag{C.4}$$

Noting that when $\varphi_{i,j,n} = 1$, $|Y_{i,j,n}| = 0$, we need only consider the case $\varphi_{i,j,n} = 0$, so that (C.4) becomes

$$(1+(r+\lambda_j)\Delta\tau)\|Y_{*,j,n}\|_\infty \leq \|Y_{*,*,n-1}\|_\infty + \lambda_j\Delta\tau\|Z\|_\infty. \tag{C.5}$$

Suppose $\|Y_{*,*,n}\|_\infty = \|Y_{*,\hat{j},n}\|_\infty$, and let $\hat{\lambda}^{(n)} = \lambda_{\hat{j}}$. Then (C.5) becomes

$$(1+\Delta\tau(r+\hat{\lambda}^{(n)}))\|Y_{*,*,n}\|_\infty \leq \|Y_{*,*,n-1}\|_\infty + \hat{\lambda}^{(n)}\Delta\tau\|Z\|_\infty. \tag{C.6}$$

Since $\mathbb{A}Y = \mathbb{B}Z$, note from (C.1) that $\mathbb{Y}_{*,*,0} = 0$. Consequently,

$$\begin{aligned}\|Y_{*,*,1}\|_\infty &\leq \frac{\hat{\lambda}^{(1)}\Delta\tau\|Z\|_\infty}{[1+\Delta\tau(r+\hat{\lambda}^{(1)})]} \\ &\leq \frac{\hat{\lambda}\Delta\tau\|Z\|_\infty}{[1+\Delta\tau(r+\hat{\lambda})]}, \\ \|Y_{*,*,2}\|_\infty &\leq \frac{\hat{\lambda}^{(2)}\Delta\tau\|Z\|_\infty}{[1+\Delta\tau(r+\hat{\lambda}^{(2)})]} + \frac{1}{[1+\Delta\tau(\hat{\lambda}^{(2)}+r)]}\left[\frac{\hat{\lambda}\Delta\tau\|Z\|_\infty}{[1+\Delta\tau(r+\hat{\lambda})]}\right] \\ &\leq \frac{\hat{\lambda}\Delta\tau\|Z\|_\infty}{[1+\Delta\tau(r+\hat{\lambda})]} + \frac{1}{[1+\Delta\tau(\hat{\lambda}+r)]}\left[\frac{\hat{\lambda}\Delta\tau\|Z\|_\infty}{[1+\Delta\tau(r+\hat{\lambda})]}\right].\end{aligned} \tag{C.7}$$

Continuing in this way, we obtain

$$\|Y\|_\infty \le \|Z\|_\infty \frac{\hat{\lambda}\Delta\tau}{[1+\Delta\tau(\hat{\lambda}+r)]}\left[1+\frac{1}{[1+\Delta\tau(\hat{\lambda}+r)]}+\cdots+\frac{1}{[1+\Delta\tau(\hat{\lambda}+r)]^{L-1}}\right]$$

$$= \|Z\|_\infty \left[1-\frac{1}{[1+\Delta\tau(\hat{\lambda}+r)]^L}\right]\left(\frac{\hat{\lambda}}{\hat{\lambda}+r}\right), \tag{C.8}$$

which gives

$$\|\mathbb{A}^{-1}\mathbb{B}\|_\infty \le \left[1-\frac{1}{[1+\Delta\tau(\hat{\lambda}+r)]^L}\right]\left(\frac{\hat{\lambda}}{\hat{\lambda}+r}\right). \tag{C.9}$$

Remark C.1 (Crank–Nicolson timestepping). Note that the above bound is obtained only for fully implicit timestepping. If Crank–Nicolson timestepping is used, then in order to ensure that $\mathbb{A}$ is an M matrix, we would require the usual severe timestep condition (i.e., twice the maximum explicit timestep size).

Remark C.2 (previous work). A bound similar to that in (C.9) was obtained in [5] in the context of a global-in-time method for American options under jump diffusion. The bound (C.2) was obtained for a global-in-time method for American options under regime switching in [31]. The bound in [31] was obtained based on a functional iteration approach, and does not appear to be as sharp as bound (C.9), in the context of a numerical algorithm.

REFERENCES

[1] A. Almendral and C.W. Oosterlee, *Accurate evaluation of European and American options under the CGMY process*, SIAM J. Sci. Comput., 29 (2007), pp. 93–117.

[2] L. Andersen, *Markov models for commodity futures: Theory and practice*, Quantitative Finance, 10 (2010), pp. 831–854.

[3] E. Ayache, *Equity-credit problem*, in Encyclopedia of Quantitative Finance, R. Cont, ed., Wiley, New York, 2010, pp. 571–575.

[4] E. Ayache, P.A. Forsyth, and K.R. Vetzal, *The valuation of convertible bonds with credit risk*, J. Derivatives, 11 (2003), pp. 25–40.

[5] E. Bayraktar and H. Xing, *Pricing American options for jump diffusions by iterating optimal stopping problems for diffusions*, Math. Methods Oper. Res., 70 (2009), pp. 505–525.

[6] E. Bayraktar and H. Xing, *Pricing Asian options for jump diffusions*, Math. Finance, 21 (2011), pp. 117–143.

[7] A. Belanger, P.A. Forsyth, and G. Labahn, *Valuing the guaranteed minimum death benefit clause with partial withdrawals*, Appl. Math. Finance, 16 (2009), pp. 451–496.

[8] O. Bokanowski, S. Maroso, and H. Zidani, *Some convergence results for Howard's algorithm*, SIAM J. Numer. Anal., 47 (2009), pp. 3001–3026.

[9] S. Boyarchenko and S. Levendorskiĭ, *American options in regime switching models*, SIAM J. Control Optim., 48 (2010), pp. 1353–1376.

[10] J. Brennan and E.S. Schwartz, *The valuation of American put options*, J. Finance, 32 (1977), pp. 449–462.

[11] S. Chen and M. Insley, *Regime switching in stochastic models of commodity prices: An application to an optimal tree harvesting problem*, J. Econom. Dynam. Control, to appear.

[12] Z. Chen and P.A. Forsyth, *Implications of a regime switching model on natural gas storage valuation and optimal operation*, Quantitative Finance, 10 (2009), pp. 159–176.

[13] S. Crepey, *About pricing equations in finance*, in Paris-Princeton Lectures on Mathematical Finance, Lecture Notes in Math. A.R. Carmona, ed., Springer, Berlin, 2010, 2003, pp. 63–203.

[14] C.W. Cryer, *The efficient solution of linear complemetarity problems for tridiagonal Minkowski matrices*, ACM Trans. Math. Software, 9 (1983), pp. 199–214.

[15] M. Dai, Q. Zhang, and Q.J. Zhu, *Trend following trading under a regime switching model*, SIAM J. Financial Math., 1 (2010), pp. 780–810.

[16] Y. d'Halluin, P.A. Forsyth, and G. Labahn, *A penalty method for American options with jump diffusion processes*, Numer. Math., 97 (2004), pp. 321–352.
[17] Y. d'Halluin, P.A. Forsyth, and K.R. Vetzal, *Robust numerical methods for contingent claims under jump diffusion processes*, IMA J. Numer. Anal., 25 (2005), pp. 87–112.
[18] R.J. Elliot, L. Chang, and T.K. Siu, *Pricing options under a generalized Markov modulated jump diffusion model*, Stoch. Anal. Appl., 25 (2007), pp. 821–843.
[19] P.A. Forsyth and G. Labahn, *Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance*, J. Compuat. Finance, 11 (2008), pp. 1–44.
[20] P.A. Forsyth and K.R. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM J. Sci. Comput., 23 (2002), pp. 2095–2122.
[21] N. Haldrup and M.O. Nielsen, *A regime switching long memory model for electricity prices*, J. Econom., 135 (2006), pp. 349–376.
[22] M. Hardy, *A regime switching model of long term stock returns*, North American Actuarial Journal, 5 (2001), pp. 41–53.
[23] Y. Huang, P.A. Forsyth, and G. Labahn, *Combined fixed point and policy iteration for HJB equations in finance*, SIAM J. Numer. Anal., submitted.
[24] Y. Huang, P.A. Forsyth, and G. Labahn, *Iterative Methods for the Solution of the Singular Control Formulation of a GMWB Pricing Problem*, Numer. Math., submitted.
[25] Y. Huen and H. Yang, *Option pricing with regime switching by trinomial tree*, J. Comput. Appl. Math., 233 (2010), pp. 1821–1833.
[26] K. R. Jackson and V. Surkov, *Fourier space timestepping for option pricing with Levy models*, J. Comput. Finance, 12 (2008), pp. 1–29.
[27] A. Kanas, *On real interest rate dynamics and regime switching*, J. Banking and Finance, 32 (2008), pp. 2089–2098.
[28] J.S. Kennedy, *Hedging Contingent Claims in Markets with Jumps*, Ph.D. thesis, Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, 2007.
[29] A.Q.M. Khaliq and R.H. Liu, *New numerical scheme for pricing American options with regime switching*, Internat. J. Theoret. Appl. Finance, 12 (2009), pp. 319–340.
[30] H.J. Kushner and P.G Dupuis, *Numerical Methods for Stochastic Control Problems in Continupus Time*, Springer-Verlag, New York, 1991.
[31] H. Le and C. Wang, *A finite time horizon optimal stopping problem with regime switching*, SIAM J. Control Optim., 48 (2010), pp. 5193–5213.
[32] R.H. Nochetto, T. van Petersdorff, and C.S. Zhang, *A posteriori error analysis for a class of integral equations with variational inequalities*, Numer. Math., 116 (2010), pp. 519–552.
[33] B. Ocksendal and A. Sulem, *Applied Stochastic Control of Jump Diffusions*, Springer, Heidelberg, 2005.
[34] H. Pham, V.L. Vath, and X.Y. Zhou, *Optimal switching over multiple regimes*, SIAM J. Control Optim., 48 (2009), pp. 2217–2253.
[35] R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numer. Math., 43 (1984), pp. 309–327.
[36] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[37] S. Salmi and J. Toivanen, *An iterative method for pricing American options under jump-diffusion models*, Appl. Numer. Math., 61 (2011), pp. 821–831.
[38] J. Tu, *Is regime switching in stock returns important in portfolio decisions?*, Management Sci., 56 (2010), pp. 1198–1215.
[39] R. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1961.
[40] M.I.M. Wahib, Z. Lin, and C.P. Edirisingh, *Pricing swing options in the electricity market under regime switching uncertainty*, Quantitative Finance, 10 (2010), pp. 975–994.
[41] J.H. Witte and C. Reisinger, *On the Penalization Error for American Options in a Jump Model*, Working paper, University of Oxford, Oxford, UK, 2010.
[42] H. Yang, *A numerical analysis of American options with regime switching*, J. Sci. Comput., 44 (2010), pp. 69–91.
[43] D. Yao, Q. Zhang, and X.Y. Zhou, *A regime-switching model for European options*, in Stochastic Processes, Optimization, and Control Theory: Applications in Financial Engineering, Queueing Networks, and Manufacturing Systems, Internat. Ser. Oper. Res. Management Sci. 94, Springer, New York, 2006, pp. 281–300.
[44] Q. Zhang and X.Y. Zhou, *Valuation of stock loans with regime switching*, SIAM J. Control Optim., 48 (2009), pp. 1229–1250.
[45] R. Zvan, P.A. Forsyth, and K.R. Vetzal, *Penalty methods for American options with stochastic volatility*, J. Comput. Appl. Math., 91 (1998), pp. 199–218.