A Comparison of Iterated Optimal Stopping and Local Policy Iteration for American Options Under Regime Switching *

J. Babbin[†] P. A. Forsyth[‡] G. Labahn[§]

May 9, 2013

Abstract

A theoretical analysis tool, iterated optimal stopping, has been used as the basis of a numerical algorithm for American options under regime switching [25]. Similar methods have also been proposed for American options under jump diffusion [4] and Asian options under jump diffusion [5]. An alternative method, local policy iteration, has been suggested in [27, 19]. Worst case upper bounds on the convergence rates of these two methods suggest that local policy iteration should be preferred over iterated optimal stopping [19]. In this article, numerical tests are presented which indicate that the observed performance of these two methods is consistent with the worst case upper bounds. In addition, while these two methods seem quite different, we show that either one can be converted into the other by a simple rearrangement of two loops.

Keywords: Iterated optimal stopping, local policy iteration, regime switching
AMS: 60G40, 65N06, 65K15
Version: May 9, 2013

1 Introduction

In this work, our focus is on comparing iterative methods for solving discretized Hamilton Jacobi Bellman equations. As an example, we consider the case of an American option, assuming that the underlying asset follows a regime switching model. This gives rise to a coupled system of variational inequalities. After discretizing these equations, using a fully implicit or Crank-Nicolson finite difference scheme, the main challenge is efficiently solving the resulting nonlinear algebraic equations.

We center our comparison on two methods, *iterated optimal stopping* and *local policy iteration*. Iterated optimal stopping was developed in a recent article [25], where the authors analyse the properties of the solution of a finite time optimal stopping (American) option pricing problem under regime switching. That particular work is based on similar analysis carried out in [4] for American options under jump diffusion, and in [5] for Asian options under jump diffusion. The method of analysis in [25] constructs a sequence of functions which converge monotonically to the exact solution of the optimal stopping problem. In [25], a numerical method is developed based in this concept. The iterated optimal stopping technique was also used to develop numerical algorithms for American and Asian options under jump diffusion [4, 5].

In [27], a local policy iteration method was developed for solving the discretized variational inequality for pricing American options under a jump diffusion process. This method was generalized for the case of American options under regime switching in [19]. In addition, a form of local policy iteration was suggested in [30], however no analysis of the convergence of this iteration was given.

*This work was supported by Tata Consulting Services and the Natural Sciences and Engineering Research Council of Canada.

[†] Computational Mathematics, University of Waterloo, Waterloo ON, Canada N2L 3G1 jbabbin@uwaterloo.ca
[‡] Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1 paforsyt@uwaterloo.ca
[§] Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1 glabahn@uwaterloo.ca

Superficially, iterated optimal stopping and local policy iteration appear to be very different algorithms. However, when viewed in more general terms, these two techniques are simply different iterative methods for solving the same set of discretized nonlinear algebraic equations. In [19], worst case upper bounds on the convergence rate for both methods were developed. The bound for local policy iteration is favorable compared to the bound for iterated optimal stopping. However, it is not clear that the upper bounds in [19] are tight. In addition, no numerical tests using iterated optimal stopping are given in [19].

In this paper we provide an extensive set of numerical tests to study the performance of iterated optimal stopping and local policy iteration. Our tests show that the performance of both algorithms is consistent in practice with the worst case upper bounds. However it is also the case that these methods appear to approach their worst case bounds only for long term contracts. We also observe that implementation is not an issue. Indeed, we show that an existing implementation of iterated optimal stopping can be converted to local policy iteration by effectively interchanging two loops. Finally, iterated optimal stopping requires significantly more storage than local policy iteration. As a result we recommend the use of local policy iteration.

The remainder of the paper proceeds as follows. Section 2 gives the basics of the regime switching model, including the nonlinear partial differential equation for pricing American options. The continuous version of the iterated optimal stopping method is presented in Section 3 while the discretization of the continuous American option problem is presented in Section 4. Section 5 provides the matrix representation of the nonlinear discrete equations with iterative methods for solving these equations given in the following section, including algorithm formulations of local policy iteration and iterated optimal stopping. Numerical tests appear in Section 7 followed by a concluding section.

2 Regime Switching Model

Financial practitioners are intimately aware of the shortcomings of the now ubiquitous Black-Scholes framework. Asset prices driven by a geometric Brownian motion process with constant volatility and drift cannot replicate the distribution of returns seen in historical stock prices [28]. A regime switching process is a computationally parsimonious technique for introducing stochastic volatility into the underlying stochastic model. Regime switching processes have been applied to problems in electricity markets [6], long term insurance guarantees [18], forestry valuation [12], and gas storage [13].

Define a finite set of K regimes and a volatility σ_j and drift rate $\mu_j^{\mathbb{P}}$ associated with each state $j = 1 \ldots K$. The $\mathbb{P}$ denotes that drift rates are observed in the real-world measure. A continuous Markov chain process is used to transition between any two states. The system of stochastic equations governing the regime switching process is

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

where S is the asset price, dZ is the increment of a Wiener process and ξ_{jk} is the jump amplitude when a transition from state $j \to k$ occurs. The Poisson process dX_{jk} is defined by

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

where λ_{jk} is the probability of transitioning from regime $j \to k$. Observe that the asset price jumps from $S \to \xi_{jk} S$ when such a transition occurs. To prevent a jump in the absence of a regime switch, set $\xi_{jj} = 1$.

To determine the fair-value of an option under regime switching, the hedging portfolio in regime j, P_j is constructed

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

where $\mathcal{V}_j$ is the no-arbitrage value of the option in regime j, e is the amount held of the underlying asset with price S, and w_k is the amount held of the additional hedging instruments with price F_k. Provided that the underlying asset and the additional hedging instruments form a non-redundant set, all risk can be eliminated through a dynamic hedging strategy [24]. This permits the use of a no-arbitrage argument leading to the American option pricing equations

$$\min\left[\ \frac{\partial \mathcal{V}_j}{\partial \tau} - \mathcal{L}_j \mathcal{V}_j - \lambda_j \mathcal{J}_j \mathcal{V},\ \ \mathcal{V}_j - \mathcal{V}^*\ \right] = 0\ ;\ j = 1, \dots, K\ , \tag{2.4}$$

where $\tau = T - t$ represents time running backwards from expiry T, and $\mathcal{V}^*(S) = \mathcal{V}(S, \tau = 0)$ is the payoff condition. The differential operators $\mathcal{L}_j$ and $\mathcal{J}_j$ are defined as

$$\mathcal{L}_j \mathcal{V}_j = \left(\frac{\sigma_j^2 S^2}{2}\right) \frac{\partial^2 \mathcal{V}_j}{\partial S^2} + (r - \rho_j) S \frac{\partial \mathcal{V}_j}{\partial S} - (r + \lambda_j)\mathcal{V}_j \tag{2.5a}$$

$$\mathcal{J}_j \mathcal{V} = \sum_{\substack{k=1 \\ k \neq j}}^{K} \frac{\lambda_{jk}}{\lambda_j} \mathcal{V}_k(\xi_{jk} S, \tau)\ , \tag{2.5b}$$

where $r > 0$ is the risk-free rate (i.e. the rate of return of a security which has no possibility of default) and

$$\rho_j = \sum_{\substack{k=1 \\ k \neq j}}^{K} \lambda_{jk}(\xi_{jk} - 1)\ \ ;\ \ \lambda_j = \sum_{\substack{k=1 \\ k \neq j}}^{K} \lambda_{jk}\ . \tag{2.6}$$

It is assumed that ξ_{jk} are deterministic functions of S and t. The risk neutral transition densities λ_{jk} are not unique. In practice, we calibrate the parameters in equation (2.5) to market data, consistent with the market's pricing measure. Note that equation (2.5a) is independent of $\mu_j^{\mathbb{P}}$, due to the usual hedging arguments [24].

For computational purposes, equation (2.4) 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)

$$\mathcal{V}(S_{\max}, \tau) \ \ = \ \ \mathcal{V}^*(S_{\max})\ , \forall \tau\ , \tag{2.8}$$

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 [24]. More precisely, the term $\mathcal{V}_k(\xi_{jk} S, \tau)$ in equation (2.5b) is replaced by $\mathcal{V}_k(\min(S_{\max}, \xi_{jk} S), \tau)$.

Remark 2.1 (Viscosity Solution) *Equation (2.4) is a special case of the more general systems of Variational Inequalities (VIs) considered in [15], where it is shown that VIs such as (2.4) have unique, continuous, locally bounded viscosity solutions in the interior of the solution domain. The definition of a viscosity solution must be generalized for systems of PDEs [9, 15, 21, 22]. One of the advantages of the viscosity solution approach is that boundary and initial conditions can be interpreted in the relaxed viscosity sense, which means that the solution may not converge to the initial condition in the usual sense [23, 2]. For example, [14] discusses the case of discontinuous initial conditions which are common in financial applications (i.e. digital options).*

Remark 2.2 (Localized Boundary Conditions) *The effect of the localized boundary condition (2.8) can be made arbitrarily small by selecting $S_{\max}$ sufficiently large [3].*

3 Iterated Optimal Stopping: Continuous Case

The basic idea of iterated optimal stopping [4, 25, 5] is to consider a sequence of functions $(\mathcal{V}_j)^m$, such that

$$\min\left[\ \frac{\partial(\mathcal{V}_j)^{m+1}}{\partial\tau} - \mathcal{L}_j(\mathcal{V}_j)^{m+1} - \lambda_j\mathcal{J}_j(\mathcal{V})^m,\ \ (\mathcal{V}_j)^{m+1} - \mathcal{V}^*\ \right] = 0\ ;\ j = 1,\ldots,K\ . \tag{3.1}$$

Under suitable assumptions, it can be shown that the sequence $(\mathcal{V}_j)^m$ converges monotonically to the solution of equation (2.4) [25].

Note that at each iteration m in equation (3.1), the problem reduces to solving a set of decoupled American type problems, one for each regime. This allows the authors in [25] to deduce properties of the solution $\mathcal{V}_j$. The operator $\mathcal{J}_j$ is replaced by an integral operator in [4, 5] and similar arguments are constructed for jump diffusions.

Remark 3.1 (Interpretation of equation (3.1)) *Problem (3.1) effectively reduces problem (2.4) to a sequence of simpler problems. Each of these simpler problems has known properties. This makes formulation (3.1) attractive from a mathematical point of view. However, we will see that formulation (3.1) is not well-suited to numerical computation.*

4 Discretization

Numerically solving the American option pricing equations in (2.4) is achieved through a finite-difference discretization. A price grid is constructed consisting of a set of $i_{\max}$ nodes

$$\{S_1, S_2, \ldots, S_{i_{\max}}\} \tag{4.1}$$

following a sequence of L timesteps of size $\Delta\tau$

$$\{\tau^0, \tau^1, \ldots, \tau^L\} \tag{4.2}$$

where $\tau^n = n\Delta\tau$.

Denote the numerical solution at (S_i, τ^n) in regime j by $V_{i,j}^n$. A vector of solution values is constructed as follows

$$V^n = [V_{1,1}^n, ..., V_{i_{\max},1}^n, \ldots, V_{1,K}^n, ..., V_{i_{\max},K}^n]'. \tag{4.3}$$

The solution vector has length $N = K \cdot i_{\max}$. For succinctness, a single row index is often used to refer to entries of V^n as follows

$$V_\ell^n = V_{i,j}^n\ \ ;\ \ \ell = (j-1)i_{\max} + i\ \ . \tag{4.4}$$

4.1 $\mathcal{L}$ Operator Discretization

Let $\mathcal{L}_j^h$ denote the discrete form of $\mathcal{L}_j$ in (2.5a). We use a combination of central, forward and backward differencing to generate a discretization of the form

$$(\mathcal{L}_j^h V^n)_{ij} = \alpha_{i,j}V_{i-1,j}^n + \beta_{i,j}V_{i+1,j}^n - (\alpha_{i,j} + \beta_{i,j} + r + \lambda_j)V_{i,j}^n \tag{4.5}$$

Central differencing is used as much as possible with forward backward differencing used only as required to ensure that the positive coefficient condition

$$\alpha_{i,j} \geq 0\quad\ ;\quad\ \beta_{i,j} \geq 0\ \ ;\forall i,j\ , \tag{4.6}$$

is enforced. We refer the reader to [17] for details.

4.2 $\mathcal{J}$ Operator Discretization

Let $\mathcal{J}_j^h$ denote the discrete form of $\mathcal{J}_j$ in (2.5b). This operator is discretized through a linear interpolation approximation as follows

$$[\mathcal{J}_j^h V^n]_{i,j} \quad = \quad \sum_{\substack{k=1 \\ k \neq j}}^{K} \frac{\lambda_{jk}}{\lambda_j} \mathcal{I}_{i,j,k}^h V^n \ , \tag{4.7}$$

where

$$\begin{aligned} \mathcal{I}_{i,j,k}^h V^n &= w V_{m,k}^n + (1-w) V_{m+1,k}^n \\ &\simeq \mathcal{V}_k(\min(S_{i_{\max}}, \xi_{jk} S_i), \tau^n) \ , \end{aligned} \tag{4.8}$$

with $w \in [0,1]$ such that

$$\min(S_{i_{\max}}, \xi_{jk} S_i) = w S_m + (1-w) S_{m+1} \ . \tag{4.9}$$

Note that jumps which extend outside the price grid are approximated by the value at the largest node $S_{i_{\max}}$.

4.3 Imposing the American Constraint

The American constraint is handled implicitly by rewriting (2.4) in direct control form as

$$\max_{\varphi \in \{0,1\}} \Big[\Omega \ \varphi(\mathcal{V}^* - \mathcal{V}_j) - (1-\varphi) \Big(\frac{\partial \mathcal{V}_j}{\partial \tau} - \mathcal{L}_j \mathcal{V}_j - \lambda_j \mathcal{J}_j \mathcal{V} \Big) \Big] = 0 \ , \tag{4.10}$$

where we have introduced a scaling parameter $\Omega > 0$. In exact arithmetic, the introduction of the scaling factor Ω in equation (4.10) has no effect. However, any iterative algorithm used to solve equation (4.10) requires comparison to two terms which have different units. In order to remedy this situation, we use the scaling suggested in [19]

$$\Omega = \frac{C}{\Delta \tau} \ , \tag{4.11}$$

where $C > 0$ is a dimensionless constant.

Equation (4.10) is discretized using fully-implicit ($\theta = 1$) or Crank-Nicolson ($\theta = 0.5$) timestepping

$$\begin{aligned} (1-\varphi_{i,j}^{n+1}) \Big(\frac{V_{i,j}^{n+1}}{\Delta \tau} \quad - \quad & \theta \mathcal{L}_j^h V_{i,j}^{n+1} \Big) + \Omega \ \varphi_{i,j}^{n+1} V_{i,j}^{n+1} \\ = \quad & (1-\varphi_{i,j}^{n+1}) \frac{V_{i,j}^n}{\Delta \tau} + \Omega \ \varphi_{i,j}^{n+1} \mathcal{V}_i^* + (1-\varphi_{i,j}^{n+1}) \lambda_j \theta [\mathcal{J}_j^h V^{n+1}]_{i,j} \\ & \quad + (1-\varphi_{i,j}^{n+1})(1-\theta) \big[\mathcal{L}_j^h V_{i,j}^n + \lambda_j [\mathcal{J}_j^h V^n]_{i,j} \big] \quad ; \quad i < i_{\max} \\ V_{i,j}^{n+1} \quad = \quad & \mathcal{V}_i^* \quad ; \quad i = i_{\max} \ , \end{aligned} \tag{4.12a}$$

where

$$\begin{aligned} \{\varphi_{i,j}^{n+1}\} \quad \in \quad \operatorname*{arg\,max}_{\varphi \in \{0,1\}} & \Big\{ \Omega \ \varphi(\mathcal{V}_i^* - V_{i,j}^{n+1}) - (1-\varphi) \Big(\frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta \tau} \\ & - \theta \big(\mathcal{L}_j^h V_{i,j}^{n+1} + \lambda_j [\mathcal{J}_j^h V^{n+1}]_{i,j} \big) - (1-\theta) \big(\mathcal{L}_j^h V_{i,j}^n + \lambda_j [\mathcal{J}_j^h V^n]_{i,j} \big) \Big) \Big\} \ , \end{aligned} \tag{4.12b}$$

and $\varphi_\ell = 1$ indicates early exercise is optimal at node ℓ, otherwise $\varphi_\ell = 0$.

Remark 4.1 (Convergence to the viscosity solution) *It is straightforward to show, using the methods in [17] that scheme (4.12), is unconditionally l_∞ stable ($\theta = 1$), monotone and consistent, and hence converges to the viscosity solution of equation (2.4).*

5 General Form of Equations

Let Q be a vector of controls defined by

$$Q = [\varphi_{1,1}, ..., \varphi_{i_{\max},1}, \ldots, \varphi_{1,K}, ..., \varphi_{i_{\max},K}]' \, . \tag{5.1}$$

The discrete equations (4.12) can then be written as a nonlinear matrix problem

$$\begin{aligned} &\mathcal{A}^*(Q)\, U = \mathcal{C}(Q, V^n) \\ &\text{with } Q_\ell = \underset{Q_s \in \{0,1\}}{\arg\max} \Big[-\mathcal{A}^*(Q) U + \mathcal{C}(Q, V^n) \Big]_\ell \end{aligned} \tag{5.2}$$

where U is the desired solution V^{n+1} at the next time level and A^* is an $N \times N$ matrix. For computational purposes, we split the A^* matrix as follows

$$\mathcal{A}^*(Q) = \mathcal{A}(Q) - \mathcal{B}(Q) \tag{5.3}$$

with $\mathcal{A}(Q)$ containing the coefficients of nodes coupled within the same regime, while $\mathcal{B}(Q)$ contains the coefficients that couple nodes in different regimes. This splitting will be useful when constructing numerical algorithms to solve the nonlinear equations in Section 6.

The matrix coefficients for nodes $i < i_{\max}$ can be deduced from equations (4.12) as shown below

$$\begin{aligned} [\mathcal{A}(Q)U]_\ell = [\mathcal{A}U]_\ell &= (1-\varphi_\ell)\left(\frac{U_\ell}{\Delta\tau} - \theta \mathcal{L}_j^h U_\ell \right) + \varphi_\ell \Omega \; U_\ell \\ [\mathcal{B}(Q)U]_\ell = [\mathcal{B}U]_\ell &= (1-\varphi_\ell)\lambda_j \theta [\mathcal{J}_j^h U]_\ell \\ [\mathcal{C}(Q, V^n)]_\ell = \quad C_\ell &= (1-\varphi_\ell)\frac{V_\ell^n}{\Delta\tau} + \varphi_\ell \Omega \; \mathcal{V}_i^* \\ &\quad + (1-\varphi_\ell)(1-\theta)\big[\mathcal{L}_j^h V_\ell^n + \lambda_j [\mathcal{J}_j^h V^n]_\ell\big] \; . \end{aligned} \tag{5.4}$$

The boundary condition at node $i = i_{\max}$ requires that

$$[\mathcal{A}U]_\ell = U_{i_{\max},j} \; ; \; [\mathcal{B}U]_\ell = 0 \; ; \; \mathcal{C}_\ell = \mathcal{V}_{i_{\max}}^* \; ; \; \ell = (j-1)i_{\max} + i_{\max}. \tag{5.5}$$

6 Iterative Methods

Two numerical methods are presented to solve the nonlinear matrix problem in (5.2). A local policy iteration is compared to an iterated optimal stopping iteration proposed by [4]. It will be useful to note the following result.

Lemma 6.1 (Uniqueness of Solution of (5.2)) *There exists a unique solution to equation (5.2).*

Proof. Note that the positive coefficient condition (4.6) guarantees that $\mathcal{A}^*(Q)$ is an M-matrix. In addition, the construction of $\mathcal{C}(Q, V^n)$ in equation (5.4) ensures that $\mathcal{C}(Q, V^n)$ is bounded for any finite mesh. Existence and uniqueness follow from the results in [7, 20]. □

6.1 Local Policy Iteration

In this approach, we directly discretize equation (2.4). Following the approach of [30], the matrix splitting in (5.3) is used to solve the American control problem with the regime coupling terms lagged behind one iteration. This local policy method is outlined in Algorithm 6.1. This method was suggested in [27] for

Algorithm 6.1 Local Policy Iteration

```
1: $(V^0)^0 = \text{payoff}$
2: for $n = 0, 1, 2, \ldots, L-1$ do
3:     $(V^{n+1})^0 = V^n$
4:     for $k = 0, 1, 2, \ldots$ until converge do
5:         Solve : $\max_{Q_\ell \in \{0,1\}} \left\{ -\mathcal{A}(Q)(V^{n+1})^{k+1} + \mathcal{B}(Q)(V^{n+1})^k + \mathcal{C}(Q, V^n) \right\} = 0$
6:         if $k > 0$ and $\max_\ell \frac{|(V_\ell^{n+1})^{k+1} - (V_\ell^{n+1})^k|}{\max\left[scale, |(V_\ell^{n+1})^{k+1}|\right]} < tolerance_{outer}$ then
7:             $V^{n+1} = (V^{n+1})^{k+1}$;  break from the iteration
8:         end if
9:     end for
10: end for
```

American options under jump diffusion. Note that we write Algorithm 6.1 for all timesteps, although the actual iteration occurs only within each timestep. This will facilitate comparison with the iterated optimal stopping algorithm. We remark that a related idea for impulse control problems was suggested in [10, 11].

The relative convergence tolerance in Line 7 in Algorithm 6.1 uses a *scale* factor to ensure that unrealistic accuracy criteria is not required for very small option values. Typically $scale = 1$ for options priced in dollars. Line 5 in Algorithm 6.1 is solved by iterating on $(V^{n+1})^{k+1}$. We use Algorithm 6.2 to solve for $(V^{n+1})^{k+1}$.

Algorithm 6.2 Inner Iteration (Policy Iteration)

```
1: $U^0 = (V^{n+1})^k$
2: for $m = 0, 1, 2, \ldots$ until converge do
3:     $Q^m = \arg\max_{Q_\ell \in \{0,1\}} \left\{ -A(Q)U^m + B(Q)(V^{n+1})^k + C(Q, V^n) \right\}$
4:     Solve:   $A(Q^m)U^{m+1} = B(Q^m)(V^{n+1})^k + C(Q^m, V^n)$
5:     if $k > 0$ and $\max_\ell \frac{|U_\ell^{m+1} - U_\ell^m)|}{\max\left[scale, |U_\ell^{m+1})|\right]} < tolerance_{inner}$ then
6:         break from the iteration
7:     end if
8: end for
9: $(V^{n+1})^{k+1} = U^{m+1}$
```

For American options, Line 5 in Algorithm 6.1 reduces to solving a Linear Complementarity Problem (LCP). In general, even for a tridiagonal LCP problem, an iterative method is required [16] for the local American problem. In the special case of a simple put or call, only a single iteration is necessary [8], 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 and Schwartz method [8] to solve the local American problem, as in [27], for the local policy iteration. However, the standard Brennan and Schwartz algorithm [8] cannot be directly applied to more general problems, such as the American butterfly, which we will use as a test case.

Theorem 6.1 *Suppose $(V^{n+1})^k$ is the exact solution of Line 5 in Algorithm 6.1, and let $E^k = (V^{n+1})^k - V^{n+1}$ where V^{n+1} is the exact solution to equation* (5.2)*. If the matrices $\mathcal{A}$ and $\mathcal{B}$ are given by equation* (5.4) *then the local policy iteration in Algorithm 6.1 converges at the rate*

$$\frac{\| E^{k+1} \|_\infty}{\| E^k \|_\infty} \leq \frac{\theta \hat{\lambda} \Delta\tau}{1 + \theta(r + \hat{\lambda})\Delta\tau} \qquad \textit{where } \hat{\lambda} = \max_j \lambda_j \quad . \tag{6.1}$$

Proof. See [19]. □

Remark 6.1 *Result (6.1) is independent of the technique used to solve line 5 in Algorithm 6.1.*

Remark 6.2 *In Theorem 6.1, we have assumed that $(V^{n+1})^k$ is the exact solution of Line 5 in Algorithm 6.1. In general, this will not be the case if an iterative method such as Algorithm 6.2 is used to solve Line 5 in Algorithm 6.1. A more detailed estimate for the convergence rate taking this into account is given in Appendix A.*

6.2 Iterated Optimal Stopping: Discretization

The philosophy behind this approach is to discretize equation (3.1). In [25], the authors numerically solve a sequence of iterated optimal stopping problems to price an American put under a regime switching model. This *global-in-time* iteration is seen in Algorithm 6.3. Line 4 is solved in the same manner as described in Algorithm 6.2.

Algorithm 6.3 Iterated Optimal Stopping

```
1: $(V^n)^0 =$ payoff;  $n = 0, ..., L$
2: for $k = 0, 1, 2, \ldots$ until converge do
3:     for $n = 0, 1, \ldots, L-1$ do
4:         Solve : $\max_{Q_\ell \in \{0,1\}} \Big[ -A(Q)(V^{n+1})^{k+1} + B(Q)(V^{n+1})^k + C(Q, (V^n)^{k+1}) \Big] = 0$
5:     end for
6:     if $k > 0$ and $\max_{\ell,n} \frac{|(V_\ell^{n+1})^{k+1} - (V_\ell^{n+1})^k|}{\max\left[scale, |(V_\ell^{n+1})^{k+1}|\right]} < tolerance_{outer}$ then
7:         break from the iteration
8:     end if
9: end for
```

By storing the solutions at all timesteps in a single vector and constructing two $N(L+1) \times N(L+1)$ matrices, Algorithm 6.3 can be rewritten in the same form as Algorithm 6.1. As with Theorem 6.1 we can determine the following convergence bound for the iterated optimal stopping iteration.

Theorem 6.2 *Assume that $(V^{n+1})^k$ is the exact solution to Line 4 in Algorithm 6.3. Let $E^k = \max_n \parallel (V^{n+1})^k - V^{n+1} \parallel_\infty$ where V^{n+1} is the exact solution to equation (5.2) at timestep $n+1$. If the matrices $\mathcal{A}$ and $\mathcal{B}$ are given by equation (5.4) and fully implicit time stepping ($\theta = 1$) is used, then the iterated optimal stopping iteration converges at the rate*

$$\frac{E^{k+1}}{E^k} \leq \left(1 - \frac{1}{\left[1 + \Delta\tau(\hat{\lambda} + r)\right]^L} \right) \left(\frac{\hat{\lambda}}{\hat{\lambda} + r} \right) \tag{6.2}$$

Proof. See [19]. □

Remark 6.3 *As the timestep size decreases, the global convergence bound (6.2) becomes*

$$\lim_{\substack{\Delta\tau \to 0 \\ L \to \infty \\ L\Delta\tau = T}} \left(1 - \left[1 + \Delta\tau(\hat{\lambda} + r)\right]^{-L} \right) \left(\tfrac{\hat{\lambda}}{\hat{\lambda}+r} \right) = \left(1 - e^{-T(\hat{\lambda}+r)} \right) \left(\tfrac{\hat{\lambda}}{\hat{\lambda}+r} \right) \quad . \tag{6.3}$$

Hence, timestep refinements will only have a limited impact on the convergence of the iterated optimal stopping iteration. Contracts spanning longer periods of time have a theoretically worse convergence bound. Additionally, storage of solution vectors at every time level is required.

The iterated optimal stopping convergence bound in (6.2) is worse than the local policy bound in (6.1) since

$$\frac{\hat{\lambda}\Delta\tau}{1+(r+\hat{\lambda})\Delta\tau} \leq \left(1-\Big[1+\Delta\tau(\hat{\lambda}+r)\Big]^{-L}\right)\left(\frac{\hat{\lambda}}{\hat{\lambda}+r}\right) \quad . \tag{6.4}$$

Proposition 6.1 (Extension to Crank-Nicolson Case) *Provided that*

$$\Delta\tau \leq 2(\Delta\tau)_e \tag{6.5}$$

where $(\Delta\tau)_e$ is the maximum stable explicit timestep size, then we can extend the above convergence analysis for the Crank Nicolson case ($\theta = 1/2$)

$$\frac{E^{k+1}}{E^k} \leq \left(1-\left[\frac{1-(1-\theta)\Delta\tau(\hat{\lambda}+r)}{1+\theta\Delta\tau(\hat{\lambda}+r)}\right]^L\right)\left(\frac{\hat{\lambda}}{\hat{\lambda}+r}\right) \ . \tag{6.6}$$

Proof. This follows the same steps as used in [19]. □

Remark 6.4 *In the limit as $\Delta t \rightarrow 0$, then bound (6.6) becomes*

$$\lim_{\substack{\Delta\tau\rightarrow 0\\ L\rightarrow\infty\\ L\Delta\tau=T}} \left(1-\left[\frac{1-(1-\theta)\Delta\tau(\hat{\lambda}+r)}{1+\theta\Delta\tau(\hat{\lambda}+r)}\right]^L\right)\left(\frac{\hat{\lambda}}{\hat{\lambda}+r}\right) = \left(1-e^{-T(\hat{\lambda}+r)}\right)\left(\tfrac{\hat{\lambda}}{\hat{\lambda}+r}\right) \quad . \tag{6.7}$$

This has the same limit as $\Delta\tau \rightarrow 0$ as in equation (6.3). Condition (6.5) is required to ensure that iterated optimal stopping form of the global matrix is an M matrix [19]. This is a very severe condition in practice, and is not practically useful (that is, it would be better to simply use an explicit method in this case).

Remark 6.5 (Rate of Convergence) *A similar bound on the rate of convergence for the case of an Asian option under jump diffusion was obtained in [5], which is a special case of the more general result in [19]. It is straightforward to obtain obtain a similar convergence estimate for American Asian options, again using the method in [19]. In [5], this is referred to as* exponential convergence*. In the usual terminology of iterative methods, this would be refered to as linear convergence.*

In [25], the rather weaker estimate

$$\frac{E^{k+1}}{E^k} \leq \left(\frac{\hat{\lambda}}{\hat{\lambda}+r}\right) \tag{6.8}$$

was obtained.

Proposition 6.2 (Uniqueness of Solution: Algorithms 6.1 and 6.3) *If the matrix $\mathcal{A}^*(Q)$ is an M-matrix, then if local policy iteration and iterated optimal stopping converge, they converge to the same solution.*

Proof. From Algorithm 6.1 and Algorithm 6.3, we have that at convergence

$$\begin{aligned} &\mathcal{A}^*(Q)\,V^{n+1} = \mathcal{C}(Q, V^n) \\ &\text{with } Q_\ell = \operatorname*{arg\,max}_{Q_s\in\{0,1\}}\Big[-\mathcal{A}^*(Q)V^{n+1} + \mathcal{C}(Q, V^n)\Big]_\ell \end{aligned} \tag{6.9}$$

		American Put		**American Butterfly**	
Refinement	Timesteps	Nodes	Unknowns	Nodes	Unknowns
0	25	29	58	51	153
1	50	57	114	101	303
2	100	113	226	201	603
3	200	225	450	401	1203
4	400	449	898	801	2403
5	800	897	1794	1601	4803
6	1600	1793	3586	3201	9603
7	3200	3585	7170	6401	19203

Table 1: Grid and timestep refinement levels used during numerical tests. On each refinement, a new grid point is placed halfway between all old grid points and the number of timesteps is doubled. A constant timestep size is used.

From Lemma 6.1, the solution of equation (6.9) is unique. □

Remark 6.6 (Implementation Issues) *The same algorithm is used to solve line 5 in Algorithm 6.1 as used in line 4 in Algorithm 6.3. Consequently, Algorithm 6.3 can be converted to Algorithm 6.1 by simply interchanging the n and k loops. This is a trivial implementation change, which has a significant effect on the properties of the algorithm.*

Remark 6.7 (Jump Diffusions) *The iterated optimal stopping algorithm has been proposed as a method for pricing American options under jump diffusions [4]. The general analysis in [19] can be used to obtain convergence rate estimates for local policy iteration and iterated optimal stopping for this problem as well. Similar to the regime switching case, the convergence bounds for iterated optimal stopping compare unfavorably with local policy iteration.*

7 Numerical Examples

The comparably poor upper bound on the convergence rate and heavy memory requirements would suggest that iterated optimal stopping is less computationally efficient compared to local policy iteration. Nonetheless, several authors have used this type of iteration recently [4, 5, 25]. We therefore provide numerical comparisons of the local policy and iterated optimal stopping iterations in this section to verify that the observed performance of these two algorithms is consistent with the worst case upper bounds. Two numerical examples are treated in this section, an American put contract based on the data used in [25], and an American butterfly contract based on data from [19].

The sequence of grid and timestep refinements can be seen in Table 1. The price grids consist of a fine mesh nearby the strikes and an increasingly coarser grid further away from the strike. On each grid refinement, new fine grid nodes are placed between each two coarse grid nodes, and the timesteps are doubled. A direct method is used to solve the linear tridiagonal matrix in line 4 of Algorithm 6.2.

A scale factor of $\Omega = 10^6/\Delta\tau$ and outer convergence tolerance of 10^{-6} is used for all experiments (line 6 in Algorithms 6.1 and 6.3). The inner iteration tolerance in Algorithm 6.2 was 10^{-9}.

7.1 Validation

In this section we show the convergence of the discretization as the grid is refined. Table 2 shows results for two of the American put cases reported in [25]. This is a two regime example. We use fully implicit timestepping with iterated optimal stopping. The parameter values for these results are shown in the table

	$(\sigma_1, \sigma_2) = (0.3, 0.5)$		$(\sigma_1, \sigma_2) = (0.5, 0.4)$	
Refinement	Value (Regime 1)	Value (Regime 2)	Value (Regime 1)	Value (Regime 2)
0	0.596066588	1.220613143	1.247602448	0.933862220
1	0.623233337	1.231744588	1.258122381	0.951342835
2	0.627143687	1.236397726	1.263093502	0.955611738
3	0.629053164	1.238462238	1.265163537	0.957249428
4	0.629544512	1.239313670	1.266042273	0.957915213
5	0.629756273	1.239696598	1.266438090	0.958209403
6	0.629846395	1.239878850	1.266626785	0.958348152
7	0.629888511	1.239968194	1.266719449	0.958416059
Le and Wang:	0.6300	1.2381	1.2647	0.9586

Table 2: American Put: convergence as the grid and timesteps tend to zero. Iterated optimal stopping is used with fully implicit timesteps. Parameter values are $r = 0.2, T = 1, Strike = 10, \lambda_{1,2} = 0.05, \lambda_{2,1} = 0.15$. Jump amplitudes are 1 and option values are at $S = 10$. Le and Wang results are from [25].

Strike (Put), K	10
Risk free rate, r	.05
Maximum grid price, $S_{i_{\max}}$	500
Transition probability 1 → 2, $\lambda_{1,2}$	3
Transition probability 2 → 1, $\lambda_{2,1}$	2
Jump amplitudes, $\xi_{1,2}, \xi_{2,1}$	1.0, 1.0
Volatilities, σ_1, σ_2	0.3, 0.4

Table 3: Input parameters and data for the American put problem.

caption. Our results are comparable with those in [25]. However, there is no report of grid/timestep size sensitivity or convergence criteria in [25].

7.2 American Put

A two-state model with no jumps in the asset price is used to price an American put whose payoff is given by $V^* = \max(K - S, 0)$ where K is the strike. The model parameters are shown in Table 3.

Remark 7.1 (Termination of Iteration) *Note that the termination criteria in line* 6 *of Algorithm 6.1 and Algorithm 6.3 do not precisely correspond to the bounds in equations (6.1) and (6.2), since in practice we do not have the exact solution available. The termination criteria in Algorithms 6.1 and 6.3 can be regarded as practical implementations. However, we expect (as will be verified in the numerical tests) that the trends in the number of iterations will be roughly consistent with the bounds in equations (6.1) and (6.2).*

Refinement results using the local policy iteration with fully implicit and Crank-Nicolson timesteps are shown in Tables 4 and 6 respectively. The number of iterations required for convergence is reported for various grid/timestep refinements and contract lengths.

Similar results using the iterated optimal stopping iteration are in Tables 5 and 7. The table heading "Inner Iterations" refers to those iterations required to solve lines 5 and 4 in Algorithms 6.1 and 6.3 respectively.

Remark 7.2 (Condition (6.5)) *For all the Crank-Nicolson tests, we violate condition (6.5). This has no effect on the local policy iteration algorithm, but convergence of the iterated optimal stopping algorithm cannot be guaranteed if condition (6.5) is not satisfied.*

	Refinement	Value	Outer Iterations per Timestep	Inner Iterations per Outer Iteration	Normalized CPU Time
$T = 1$	0	1.149561815	6.00	2.10	1
	1	1.165728338	5.00	2.12	2
	2	1.171152560	4.00	2.11	4
	3	1.173223677	3.21	2.11	8
	4	1.174102036	3.00	2.11	22
	5	1.174505970	3.00	2.09	68
	6	1.174699595	2.45	2.11	200
	7	1.174795968	2.05	2.11	713
$T = 10$	0	2.480493130	15.6	2.06	3
	1	2.527224461	10.4	2.07	4
	2	2.545702990	7.28	2.10	7
	3	2.552125529	5.45	2.12	13
	4	2.554365849	4.38	2.15	30
	5	2.555244216	3.76	2.16	84
	6	2.555626744	3.07	2.18	248
	7	2.555804716	3.00	2.18	925

Table 4: American Put — Refinement results for the local policy iteration using fully implicit timesteps. Option values are at $S = 10$ in Regime 1. One unit of CPU time is based on level zero, $T = 1$. Compare with Table 5.

	Refinement	Value	Outer Iterations	Inner Iterations per Timestep per Outer Iteration	Normalized CPU Time
$T = 1$	0	1.149561951	14	1.98	4
	1	1.165728413	14	1.98	8
	2	1.171152525	13	2.04	17
	3	1.173224277	13	2.12	43
	4	1.174102017	13	2.28	125
	5	1.174505855	13	2.52	417
	6	1.174700149	13	2.89	1583
	7	1.174796138	13	3.46	6809
$T = 10$	0	2.480497995	59	1.90	15
	1	2.527228409	54	1.90	30
	2	2.545706581	51	2.01	67
	3	2.552129517	49	2.17	167
	4	2.554369087	49	2.46	498
	5	2.555247451	48	2.98	1739
	6	2.555628049	48	3.86	7290
	7	2.555804652	48	5.15	35274

Table 5: American put — Refinement results for the iterated optimal stopping iteration using fully implicit timesteps. Option values are at $S = 10$ in Regime 1. CPU times are normalized so that the time for local policy iteration, level zero is one unit (see Table 4). Compare with Table 4.

	Refinement	Value	Outer Iterations	Inner Iterations	Normalized
			per Timestep	per Outer Iteration	CPU Time
$T = 1$	0	1.158047827	5.08	2.09	1
	1	1.170297442	4.06	2.13	2
	2	1.173605701	3.69	2.11	4
	3	1.174518713	3.02	2.09	8
	4	1.174777835	3.00	2.08	23
	5	1.174855570	3.00	2.06	73
	6	1.174879679	2.15	2.04	191
	7	1.174888084	2.01	2.07	713
$T = 10$	0	2.494986976	11.4	2.07	2
	1	2.535628033	7.80	2.08	3
	2	2.550159207	5.94	2.11	6
	3	2.554443725	4.79	2.12	12
	4	2.555558848	4.02	2.15	30
	5	2.555855693	3.18	2.16	80
	6	2.555939143	3.01	2.15	266
	7	2.555963088	2.98	2.15	1015

Table 6: American Put — Refinement results for the local policy iteration using Crank-Nicolson timesteps. Option values are at $S = 10$ in Regime 1. One unit of CPU time is based on level zero, $T = 1$. Compare with Table 7.

	Refinement	Value	Outer Iterations	Inner Iterations	Normalized
				per Timestep	CPU Time
				per Outer Iteration	
$T = 1$	0	1.158047915	11	2.04	3
	1	1.170297512	10	2.04	6
	2	1.173605959	10	2.04	14
	3	1.174518781	10	2.06	35
	4	1.174777806	10	2.15	100
	5	1.174855530	10	2.34	331
	6	1.174879899	10	2.59	1233
	7	1.174888119	10	3.01	5286
$T = 10$	0	2.494988694	37	2.02	10
	1	2.535630069	33	1.99	20
	2	2.550160719	32	2.01	44
	3	2.554444966	31	2.16	111
	4	2.555558751	30	2.42	325
	5	2.555856723	30	2.83	1139
	6	2.555939003	30	3.44	4597
	7	2.555962940	30	4.40	21570

Table 7: American Put — Refinement results for the iterated optimal stopping iteration using Crank-Nicolson timesteps. Option values are at $S = 10$ in Regime 1. CPU times are normalized so that the time for local policy iteration, level zero is one unit (see Table 6). Compare with Table 6.

As expected from the convergence bound in (6.1), the local policy method requires fewer outer iterations as the grid and timesteps are refined. This holds for both the shorter $T = 1$ contract and the longer $T = 10$ contract. Hence when solving larger problems, fewer outer iterations are needed.

In contrast, the iterated optimal stopping iteration exhibits several undesirable characteristics. The number of outer iterations approaches a fixed non-zero value as the grid and timestep is refined. This value gets increasingly larger for longer contract lengths in accordance with (6.3). The number of inner iterations increases with the refinement level. This is a bit surprising, since the same algorithm is used to solve line 5 in Algorithm 6.1 and line 4 in Algorithm 6.3. A detailed examination revealed a large increase in the number of inner iterations within the first few outer iterations as grid refinements were made. This may be due to the poor initial estimate for the solution in the case of iterated optimal stopping. As a result, the CPU run times grow quite rapidly with every grid refinement as compared to the local policy run times.

Remark 7.3 (Complexity of Both Methods) *Since Algorithm 6.2 is the same for both methods, the ratio of the number of floating point operations for both methods is*

$$\frac{\textit{Flops for Iterated Optimal Stopping}}{\textit{Flops for Local Policy Iteration}} = \frac{\textit{Outer Iterations} \times \textit{Inner Iterations} \;:\; \textit{Iterated Stopping}}{\textit{Outer Iterations} \times \textit{Inner Iterations} \;:\; \textit{Local Policy}} . \tag{7.1}$$

The ratio of CPU times in the Tables is approximately equal to the ratio of complexities in equation (7.1), for the finer grids (the timings are not accurate for the coarse grids).

7.3 American Butterfly

An American butterfly contract is priced with a payoff given by

$$V^* = \max(S - K_1, 0) - 2\max(S - (K_1 + K_2)/2, 0) + \max(S - K_2, 0) \tag{7.2}$$

where K_1 and K_2 represent the two strikes. Data and parameter values for the butterfly example are shown below. We assume the existence of an American contract with payoff (7.2), which can only be early exercised as a unit. This contract has been used as severe test case by several authors [1, 29, 26].

$$\begin{gathered} K_1 = 90 \quad ; \quad K_2 = 110 \qquad ; \qquad r = 0.02 \quad ; \quad S_{i_{\max}} = 5000 \\ \boldsymbol{\lambda} = \begin{bmatrix} -3.2 & 0.2 & 3.0 \\ 1.0 & -1.08 & .08 \\ 3.0 & 0.2 & -3.2 \end{bmatrix} \; ; \; \boldsymbol{\xi} \;=\; \begin{bmatrix} 1.0 & 0.90 & 1.1 \\ 1.2 & 1.0 & 1.3 \\ 0.95 & 0.8 & 1.0 \end{bmatrix} \; ; \; \boldsymbol{\sigma} = \begin{bmatrix} .2 \\ .15 \\ .30 \end{bmatrix} . \end{gathered} \tag{7.3}$$

Note that in this case, the solutions will not have continuous $(\mathcal{V}_j)_S$. The payoff is also not convex, hence this precludes the analysis in [25]. However, this does not cause any difficulty if we pose the solution to problem (2.4) in terms of viscosity solutions.

Refinement results for the butterfly payoff using the local policy iteration with fully implicit and Crank-Nicolson timestepping are displayed in Tables 8 and 10 respectively. Accompanying iterated optimal stopping refinement results are in Tables 9 and 11. Note that the same issues affecting the iterated optimal stopping method discussed in Section 7.2 persist when using the second order Crank-Nicolson discretization. The method requires significantly longer runtimes as compared to the local policy iteration, particularly for highly refined grids and longer contract lengths.

In view of Remark 7.1, we carried out additional tests to check the convergence bound in (6.2) for the American Butterfly example. In these tests, an approximation to the successive error ratio E^{k+1}/E^k was computed. The exact option values are approximated with a numerical solution generated using a convergence tolerance of 10^{-12}. The approximate error terms E^k were then calculated at each iteration and timestep. The average and maximum ratio of successive error estimates for a fully implicit iterated optimal stopping iteration are shown in Table 12. The observed error ratios tend to the bound (6.2) as the expiry time increases.

	Refinement	Value	Outer Iterations	Inner Iterations	Normalized
			per Timestep	per Outer Iteration	CPU Time
$T = 0.5$	0	4.414610687	5.64	2.14	1
	1	4.437635032	4.96	2.15	2
	2	4.450644836	4.00	2.19	5
	3	4.456010904	3.47	2.21	13
	4	4.458284154	3.02	2.24	42
	5	4.459342999	3.00	2.24	156
	6	4.459853289	3.00	2.23	717
	7	4.460104008	2.26	2.23	2710
$T = 10$	0	8.503526680	20.0	2.04	3
	1	8.391584342	12.6	2.07	5
	2	8.371971829	8.60	2.10	9
	3	8.370349958	6.35	2.12	22
	4	8.370656431	5.10	2.15	64
	5	8.371011766	4.12	2.19	199
	6	8.371245326	3.42	2.23	803
	7	8.371375743	3.05	2.24	3414

Table 8: American Butterfly — Refinement results for the local policy iteration using fully implicit timesteps. Option values are at $S = 93$ in Regime 2. CPU times are normalized so that the time for local policy iteration, level zero is one unit. Compare with Table 9.

	Refinement	Value	Outer Iterations	Inner Iterations	Normalized
				per Timestep	CPU Time
				per Outer Iteration	
$T = 0.5$	0	4.414610844	12	2.21	3
	1	4.437635066	12	2.37	7
	2	4.450644870	12	2.65	21
	3	4.456011081	12	3.10	73
	4	4.458284196	12	3.76	288
	5	4.459343003	12	4.77	1295
	6	4.459853288	12	6.35	7618
	7	4.460104147	12	8.74	45699
$T = 10$	0	8.503527951	71	1.99	16
	1	8.391586147	65	2.08	36
	2	8.371973661	62	2.26	99
	3	8.370351440	60	2.67	328
	4	8.370657097	59	3.35	1290
	5	8.371012428	58	4.56	6015
	6	8.371246794	58	6.28	36450
	7	8.371376073	58	8.76	222080

Table 9: American Butterfly — Refinement results for the iterated optimal stopping iteration using fully implicit timesteps. Option values are at $S = 93$ in Regime 2. CPU times are normalized so that the time for local policy iteration, level 0 is one unit (see Table 8). Compare with Table 8.

	Refinement	Value	Outer Iterations	Inner Iterations	Normalized
			per Timestep	per Outer Iteration	CPU Time
$T = 0.5$	0	4.440162686	5.08	2.12	1
	1	4.451422139	4.06	2.15	2
	2	4.457862480	3.94	2.15	6
	3	4.459717103	3.08	2.17	15
	4	4.460172252	3.01	2.17	52
	5	4.460299471	3.00	2.18	197
	6	4.460336267	2.61	2.06	767
	7	4.460347552	2.09	2.21	3123
$T = 10$	0	8.521803224	14.0	2.05	2
	1	8.400544555	9.40	2.09	4
	2	8.376389091	6.84	2.11	9
	3	8.372576548	5.21	2.14	24
	4	8.371770123	4.25	2.17	70
	5	8.371587805	3.84	2.19	244
	6	8.371527683	3.10	2.22	947
	7	8.371517241	3.01	2.22	4226

Table 10: American Butterfly — Refinement results for the local policy iteration using Crank-Nicolson timesteps. Option values are at $S = 93$ in Regime 2. CPU times are normalized so that the time for local policy iteration, level zero is one unit. Compare with Table 11.

	Refinement	Value	Outer Iterations	Inner Iterations	Normalized
				per Timestep	CPU Time
				per Outer Iteration	
$T = 0.5$	0	4.440162700	10	2.17	3
	1	4.451422164	9	2.33	7
	2	4.457862497	9	2.56	19
	3	4.459717136	9	2.90	66
	4	4.460172252	9	3.42	256
	5	4.460299466	9	4.22	1144
	6	4.460336333	9	5.39	6429
	7	4.460347592	9	7.18	36853
$T = 10$	0	8.521803835	44	2.13	12
	1	8.400545174	40	2.19	28
	2	8.376389721	38	2.41	78
	3	8.372577132	37	2.73	258
	4	8.371770660	36	3.33	999
	5	8.371588095	36	4.11	4494
	6	8.371527985	36	5.32	25490
	7	8.371517265	36	7.20	148238

Table 11: American Butterfly — Refinement results for the iterated optimal stopping iteration using Crank-Nicolson timesteps. Option values are at $S = 93$ in Regime 2. CPU times are normalized so that the time for local policy iteration, level zero is one unit (see Table 10). Compare with Table 10.

	Refinement	Convergence Bounds		Approximate Error Ratio Optimal Stopping	
		Local Policy	Optimal Stopping	Avg Value	Max Value
$T = 0.5$	0	0.060	0.785	0.241	0.349
	1	0.031	0.790	0.231	0.365
	2	0.016	0.793	0.224	0.362
	3	0.008	0.794	0.221	0.362
	4	0.004	0.794	0.219	0.362
$T = 10$	0	0.559	0.994	0.819	0.940
	1	0.389	0.994	0.793	0.925
	2	0.242	0.994	0.779	0.922
	3	0.138	0.994	0.771	0.920
	4	0.074	0.994	0.767	0.920

Table 12: Approximate error ratios, E^{k+1}/E^k, for the iterated optimal stopping iteration. American Butterfly case. The approximate error ratio values are computed using fully implicit timesteps. The convergence bounds also correspond to fully implicit timesteps (see (6.1) and (6.2)). The exact solution was approximated by generating a solution with a convergence tolerance of 10^{-12}.

8 Conclusions

Analysis of the iterated optimal stopping algorithm proposed in [25] for American options under regime switching shows that this technique is inferior in terms of worst case bounds on the rate of convergence compared with local policy iteration [27, 19]. Numerical tests indicate that the observed performance of both these methods is consistent with the bounds on the convergence rates, using a common type of convergence test. In addition, iterated optimal stopping requires considerably more storage compared to local policy iteration.

We also note that iterated optimal stopping has been proposed for jump diffusion problems [4, 5]. The analysis of the convergence rates for these problems is similar to that for regime switching, and can be easily carried out using the general form for the discretized equations as discussed in [19]. The analysis shows that the convergence bounds for iterated optimal stopping are worse than for local policy iteration.

Since it is a simple matter to convert an iterated optimal stopping implementation to local policy iteration (this simply involves interchanging two loops), we strongly recommend use of local policy iteration.

Appendix

A Error Bound for Local Policy Iteration with Inexact Inner Solution

In this Appendix, we generalize the result in Theorem 6.1 to include the effect of an approximate solution to Line 5 in Algorithm 6.1. We need only generalize the steps used in [19].

If V^{n+1} is a solution to equation (5.2) then

$$\max_{Q'}\left\{-\mathcal{A}(Q')V^{n+1} + \mathcal{B}(Q')V^{n+1} + \mathcal{C}(Q', V^n)\right\} = 0 \ , \tag{A.1}$$

while from Algorithm 6.1, we have

$$\max_{Q}\left\{-\mathcal{A}(Q)(V^{n+1})^{k+1} + \mathcal{B}(Q)(V^{n+1})^{k} + \mathcal{C}(Q, V^n)\right\} = \mathcal{E}^k. \tag{A.2}$$

The term $\mathcal{E}^k$ in equation (A.2) takes into account that that we may not necessarily have the exact solution to Line 5 in Algorithm 6.1, if we use Algorithm 6.2.

Subtracting equation (A.1) from equation (A.2) we obtain

$$\begin{aligned}
\mathcal{E}^k &= \max_Q\Big\{-\mathcal{A}(Q)(V^{n+1})^{k+1} + \mathcal{B}(Q)(V^{n+1})^k + \mathcal{C}(Q,V^n)\Big\} - \max_{Q'}\Big\{-\mathcal{A}(Q')V^{n+1} + \mathcal{B}(Q')V^{n+1} + \mathcal{C}(Q',V^n)\Big\} \\
&\leq \max_Q\Big\{-\mathcal{A}(Q)((V^{n+1})^{k+1} - V^{n+1}) + \mathcal{B}(Q)((V^{n+1})^k - V^{n+1})\Big\} .
\end{aligned} \tag{A.3}$$

If $\hat{Q}$ satisfies

$$\hat{Q} \in \arg\max_Q\Big\{\mathcal{A}(Q)((V^{n+1})^{k+1} - V^{n+1}) + \mathcal{B}(Q)((V^{n+1})^k - V^{n+1})\Big\} . \tag{A.4}$$

then, from equation (A.3), we have, $(E^{k+1} = (V^{n+1})^{k+1} - V^{n+1})$

$$\mathcal{A}(\hat{Q})E^{k+1} \leq \mathcal{B}(\hat{Q})E^k - \mathcal{E}^k , \tag{A.5}$$

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

$$\begin{aligned}
E^{k+1} &\leq \mathcal{A}(\hat{Q})^{-1}\mathcal{B}(\hat{Q})E^k - \mathcal{A}(\hat{Q})^{-1}\mathcal{E}^k \ \leq C_1\|E^k\|_\infty \mathbf{e} + C_2\|\mathcal{E}^k\|_\infty \mathbf{e} \\
& \quad C_1 = \max_Q \|\mathcal{A}(Q)^{-1}\mathcal{B}(Q)\|_\infty \\
& \quad C_2 = \max_Q \|\mathcal{A}(Q)^{-1}\|_\infty
\end{aligned} \tag{A.6}$$

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

$$\begin{aligned}
\mathcal{E}^k &= \max_Q\Big\{-\mathcal{A}(Q)(V^{n+1})^{k+1} + \mathcal{B}(Q)(V^{n+1})^k + \mathcal{C}(Q,V^n)\Big\} - \max_{Q'}\Big\{-\mathcal{A}(Q')V^{n+1} + \mathcal{B}(Q')V^{n+1} + \mathcal{C}(Q',V^n)\Big\} \\
&\geq \min_Q\Big\{-\mathcal{A}(Q)((V^{n+1})^{k+1} - V^{n+1}) + \mathcal{B}(Q)((V^{n+1})^k - V^{n+1})\Big\} .
\end{aligned} \tag{A.7}$$

Hence if

$$\bar{Q} \in \arg\min_{Q\in Z}\Big\{-\mathcal{A}(Q)((V^{n+1})^{k+1} - V^{n+1}) + \mathcal{B}(Q)((V^{n+1})^k - V^{n+1})\Big\} , \tag{A.8}$$

then

$$E^{k+1} \ \geq \ \mathcal{A}(\bar{Q})^{-1}\mathcal{B}(\bar{Q})E^k - \mathcal{A}(\hat{Q})^{-1}\mathcal{E}^k \geq -C_1\|E^k\|_\infty \mathbf{e} - C_2\|\mathcal{E}^k\|_\infty \mathbf{e} . \tag{A.9}$$

Equations (A.6) and (A.9) then give

$$\begin{aligned}
\|E^{k+1}\|_\infty &\leq C_1\|E^k\|_\infty + C_2\|\mathcal{E}_{\max}\|_\infty \\
& \qquad \|\mathcal{E}_{\max}\|_\infty = \max_k \|\mathcal{E}^k\|_\infty .
\end{aligned} \tag{A.10}$$

From [19], we have that

$$\begin{aligned}
C_1 &\leq \frac{\theta\hat{\lambda}\Delta\tau}{1+\theta(r+\hat{\lambda})\Delta\tau} \qquad ; \qquad \hat{\lambda} = \max_j \lambda_j \\
&< 1 ,
\end{aligned} \tag{A.11}$$

and C_2 bounded, hence

$$\|E^{k+1}\|_\infty \leq C_1^{k+1}\|E^0\|_\infty + C_2\bigg(\frac{1-C_1^{k+1}}{1-C_1}\bigg)\|\mathcal{E}_{\max}\|_\infty , \tag{A.12}$$

and in view of equation (A.11), we obtain

$$\lim_{k\to\infty} \|E^{k+1}\|_\infty \quad \leq \quad \frac{C_2}{1-C_1}\|\mathcal{E}_{\max}\|_\infty \ . \tag{A.13}$$

Manipulation of Line 4 in Algorithm 6.2 results in

$$\mathcal{A}(Q^m)(U^{m+1}-U^m) \quad = \quad \max_Q \left\{ -\mathcal{A}(Q)U^m + \mathcal{B}(Q)(V^{n+1})^k + \mathcal{C}(Q,V^n) \right\} \tag{A.14}$$

Recall that at convergence of Algorithm 6.2, we have

$$(V^{n+1})^{k+1} = U^{m+1} \ . \tag{A.15}$$

When Algorithm 6.2 terminates, we have

$$\mathcal{E}^k \quad = \quad \max_Q \left\{ -\mathcal{A}(Q)(V^{n+1})^{k+1} + \mathcal{B}(Q)(V^{n+1})^k + \mathcal{C}(Q,V^n) \right\} \ , \tag{A.16}$$

then, from equations (A.14) and (A.16) (with $U^{m+1} = (V^{n+1})^{k+1}$)

$$\begin{aligned} \mathcal{E}^k \quad = \quad & \mathcal{A}(Q^m)(U^{m+1}-U^m) + \max_Q \left\{ -\mathcal{A}(Q)U^{m+1} + \mathcal{B}(Q)(V^{n+1})^k + \mathcal{C}(Q,V^n) \right\} \\ & - \max_Q \left\{ -\mathcal{A}(Q)U^m + \mathcal{B}(Q)(V^{n+1})^k + \mathcal{C}(Q,V^n) \right\} \end{aligned} \tag{A.17}$$

and then

$$|\mathcal{E}^k| \quad \leq \quad |\mathcal{A}(Q^m)(U^{m+1}-U^m)| + \max_Q |A(Q)(U^{m+1}-U^m)| \ , \tag{A.18}$$

which implies

$$\|\mathcal{E}^k\|_\infty \quad \leq \quad 2 \cdot \max_Q \|A(Q)\|_\infty \|U^{m+1}-U^m\|_\infty \ . \tag{A.19}$$

Since $\mathcal{A}(Q^m)$ is bounded, $\|\mathcal{E}^k\|_\infty$ (and hence $\|\mathcal{E}_{\max}\|_\infty$) can be made arbitrarily small by making $tolerance_{inner}$ small in Algorithm 6.2.

References

[1] A. Almendral and C.W. Oosterlee. Accurate evaluation of European and American options under the CGMY process. *SIAM Journal on Scientific Computing*, 29:93–117, 2007.

[2] G. Barles. Convergence of numerical schemes for degenerate parabolic equations arising in finance. In L. C. G. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 1–21. Cambridge University Press, Cambridge, 1997.

[3] G. Barles, CH. Daher, and M. Romano. Convergence of numerical shemes for parabolic eqations arising in finance theory. *Mathematical Models and Methods in Applied Sciences*, 5:125–143, 1995.

[4] E. Bayraktar and H. Xing. Pricing American options for jump diffusions by iterating optimal stopping problems for diffusions. *Mathematical Methods of Operations Research*, 70:505–525, 2009.

[5] E. Bayraktar and H. Xing. Pricing Asian options for jump diffusions. *Mathematical Finance*, 21(1):117–143, January 2011.

[6] M. Bierbrauer, S. Truck, and R. Weron. Modeling electricity prices with regime switching models. In *Computational Science - ICCS 2004*, volume 3039 of *Lecture Notes in Computer Science*, pages 859–867. Springer Berlin / Heidelberg, 2004.

[7] O. Bokanowski, S. Maroso, and H. Zidani. Some convergence results for Howard's algorithm. *SIAM Journal on Numerical Analysis*, 47:3001–3026, 2009.

[8] J. Brennan and E.S. Schwartz. The valuation of American put options. *Journal of Finance*, 32:449–462, 1977.

[9] A. Briani, F. Camilli, and H. Zidani. Approximation schemes for monotone systems of nonlinear second order differential equations: Convergence result and error estimate. *Differential Equations and Applications*, 4:297–317, 2012.

[10] J. Chancelier, B. Øksendal, and A. Sulem. Combined stochastic control and optimal stopping, and application to numerical approximation of combined stochastic and impulse control. Proc. Steklov Inst. Math. 237, 140-163 (2002) and Tr. Mat. Inst. Steklova 237, 149-172 (2002)., 2002.

[11] J.P. Chancelier, M. Messaoud, and A. Sulem. A policy iteration algorithm for fixed point problems with nonexpansive operators. *Mathematical Methods in Operations Research*, 65:239–259, 2007.

[12] S. Chen and M. Insley. Regime switching in stochastic models of commodity prices: An application to an optimal tree harvesting problem. *Journal of Economic Dynamics and Control*, 36:201–219, 2012.

[13] Z. Chen and P.A. Forsyth. Implications of a regime switching model on natural gas storage valuation and optimal operation. *Quantitative Finance*, 10:159–176, 2010.

[14] R. Cont and E. Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential levy models. *SIAM Journal on Numerical Analysis*, 43:1596–1626, 2005.

[15] S. Crepey. About pricing equations in finance. In A.R. Carmona, editor, *Paris-Princeton Lectures on Mathematical Finance*, pages 63–203. Springer, Berlin, 2010. Lecture Notes in Mathematics 2003.

[16] C.W. Cryer. The efficient solution of linear complemetarity problems for tridiagonal Minkowski matrices. *ACM Transactions on Mathematical Software*, 9:199–214, 1983.

[17] P. A. Forsyth and G. Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Compuational Finance*, 11 (Winter):1–44, 2008.

[18] M. Hardy. A regime switching model of long term stock returns. *North American Actuarial Journal*, 5:41–53, 2001.

[19] Y. Huang, P.A. Forsyth, and G. Labahn. Methods for pricing American options under regime switching. *SIAM Journal on Scientific Computing*, 33(5):2144–2168, 2011.

[20] Y. Huang, P.A. Forsyth, and G. Labahn. Combined fixed point and policy iteration for Hamilton-Jacobi-Bellman equations in finance. *SIAM Journal on Numerical Analysis*, 50:1849–1860, 2012.

[21] H. Ishii and S. Koike. Viscosity solutions for monotone systems of second order elliptic PDEs. *Communications in Partial Differential Equations*, 16:1095–1128, 1991.

[22] H. Ishii and S. Koike. Viscosity solutions of a system of nonlinear elliptic PDEs arising in switching games. *Funkcialaj Ekvacioj*, 34:143–155, 1991.

[23] E. Jakobsen. Monotone schemes. In R. Cont, editor, *Encyclopedia of Quantitative Finance*, pages 1253–1263. Wiley, New York, 2010.

[24] J.S. Kennedy. *Hedging contingent claims in markets with jumps.* PhD thesis, School of Computer Science, University of Waterloo, 2007.

[25] H. Le and C. Wang. A finite horizon optimal stopping problem with regime switching. *SIAM Journal on Control and Optimization*, 48(8):5193–5213, 2010.

[26] R.H. Nochetto, T. van Petersdorff, and C.S. Zhang. A posteriori error analysis for a class of integral equations with variational inequalities. *Numerische Mathematik*, 116:519–552, 2010.

[27] S. Salmi and J. Toivanen. An iterative method for pricing American options under jump diffusion models. *Applied Numerical Mathematics*, 61:821–831, 2011.

[28] P. Wilmott. *Paul Wilmott on Quantitative Finance.* John Wiley & Sons Ltd., West Sussex, U.K., 2000.

[29] J.H. Witte and C. Reisinger. On the penalization error for American options in a jump model. Working paper, University of Oxford, 2010.

[30] H. Yang. A numerical analysis of American options with regime switching. *Journal of Scientific Computing*, 44(1):69–91, 2010.