SIAM J. NUMER. ANAL.
Vol. 50, No. 4, pp. 1861–1882

COMBINED FIXED POINT AND POLICY ITERATION FOR HAMILTON–JACOBI–BELLMAN EQUATIONS IN FINANCE*

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

Abstract. Implicit methods for Hamilton–Jacobi–Bellman (HJB) partial differential equations give rise to highly nonlinear discretized algebraic equations. The classic policy iteration approach may not be efficient in many circumstances. In this article, we derive sufficient conditions to ensure convergence of a combined fixed point policy iteration scheme for the solution of discretized equations. Numerical examples are included for a singular stochastic control problem arising in insurance (a guaranteed minimum withdrawal benefit), where the underlying risky asset follows a jump diffusion, and an American option assuming a regime switching process.

Key words. HJB equation, fully implicit, fixed point policy iteration, relaxation, singular control, penalty method, regime switching

AMS subject classifications. 65M12, 93C20

DOI. 10.1137/100812641

1. Introduction.

A number of financial pricing problems are naturally modeled in terms of solving nonlinear partial differential equations (PDEs). This is often the case for problems which arise in the context of optimal stochastic control [25, 28, 29], in which case the nonlinear PDEs are typically Hamilton–Jacobi–Bellman (HJB) equations. Some examples include natural gas storage [33], insurance products [26, 6, 14], asset allocation [34, 15], and optimal trade execution [1].

Solutions to nonlinear HJB equations are not necessarily unique, and one must take care to provide numerical procedures which ensure convergence to the viscosity solution [5, 4]. In order to ensure both numerical stability and convergence, implicit methods can be chosen over explicit methods. Implicit methods result in a nonlinear system of algebraic equations at each timestep. Solving these nonlinear equations is often the computational bottleneck.

One popular approach for solving the nonlinear equations resulting from a fully implicit discretization of HJB equations is based on the idea of policy iteration [21, 7, 25, 18, 8]. Policy iteration proceeds by solving a linear system at every step and then finding the control which gives the best local solution. Policy iteration is particularly effective when the linear system is sparse or well structured and hence easy to solve.

It has been known for some time that policy iteration can be viewed as a form of Newton iteration [30, 32, 8]. An alternative approach, known as value iteration, can be seen to be a type of nonlinear relaxation [25]. In this paper, we consider a combination of both methods.

Although our main focus here is on financial applications, where we solve systems of nonlinear algebraic equations arising after discretization of HJB equations, the final algebraic problem is similar to that arising in infinite horizon Markovian

*Received by the editors October 25, 2010; accepted for publication (in revised form) May 15, 2012; published electronically July 3, 2012. This work was supported by Credit Suisse, New York and the Natural Sciences and Engineering Research Council of Canada.
http://www.siam.org/journals/sinum/50-4/81264.html

†Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (yqhuang@ecemail.uwaterloo.ca).
‡David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (paforsyt@uwaterloo.ca, glabahn@uwaterloo.ca).

dynamic programming problems (MDPs). Hence many of the results we derive here can be applied to MDPs coming from nonfinancial applications [25]. In addition, HJB equations also arise naturally in optimal stochastic control problems [27].

Financial options are typically modeled as functions of risky assets, with the asset prices following geometric Brownian motion. However, it is well known that geometric Brownian motion is inconsistent with market data. Jump diffusion and regime switching are two important approaches, both of which are considered to better model observed risky asset stochastic processes [20, 13]. However, these are precisely cases where the use of policy iteration has efficiency issues. For example, when the underlying stochastic process is a jump diffusion then the policy iteration matrix would be dense [16], and hence the use of a direct solution of each linear system is prohibitive in terms of cost. Difficulties also arise when the underlying stochastic process is modeled using regime switching. In this case the associated linear system at each iteration is sparse, but the sparsity pattern has lost its structure. Using a direct solution method (even with a good ordering technique) no longer turns out to be efficient.

The main goal of this paper is to present an efficient scheme for solving the nonlinear discretized equations which arise from fully implicit discretization of HJB equations. We present a *fixed point policy iteration scheme* for solving the nonlinear discretized equations which arise from fully implicit discretization of HJB equations. We show that our approach converges and that the method is considerably more efficient than making use of full policy iteration. In order to validate our approach we show how this fixed point policy iteration can be used in two specific examples from financial applications. The first example is a singular control formulation of a guaranteed minimum withdrawal benefit (GMWB) [22], where the underlying risky asset follows a jump diffusion process [13]. The second example is based on an American option written on an asset which follows a regime switching process [24].

The main results of this paper are as follows:

- We derive sufficient conditions which ensure convergence of the fixed point policy iteration scheme. These conditions are very natural if we use a monotone discretization to ensure convergence to the viscosity solution.
- We verify that the conditions required for convergence are satisfied for the GMWB and regime switching examples.
- We observe that in some formulations of the control problem [8], the nonlinear optimization objective function admits an arbitrary scaling factor. We derive sufficient conditions for the convergence of the fixed point policy iteration which impose bounds on this scaling factor.
- We include numerical experiments which demonstrate that the fixed point policy iteration is more efficient than various alternative algorithms.

We emphasize here that our analysis is based on a very general framework. Although the numerical examples used in this paper have a finite control set, our analysis applies as well to cases where the admissible set of controls is infinite. For example, we do not require that the discretized equations be continuous functions of the control (see [37] for a situation where this occurs). As such our results can be applied to a wide variety of discretized HJB equations.

The proof of the convergence of the fixed point scheme for American options under jump diffusion in [16] is a special case of the more general result obtained here. The approach in this paper is also simpler than the method used in [12]. In addition, we do not rely on a special choice for the initial iterate as in [30].

2. Methods for solving algebraic equations. In [18] a number of problems in financial modeling were presented in a general form as nonlinear HJB problems. These problems were then solved by implicitly discretizing the associated PDE and then solving the resulting discrete algebraic equations. For the applications addressed in [18] an efficient method for solving the associated algebraic systems made use of a (Newton-like) policy iteration scheme. However, in some cases policy iteration has significant efficiency drawbacks. In particular this happens when the risky assets follow a stochastic process which includes a Poisson jump process. In this section we describe a new procedure, called *fixed point policy iteration*, which provides a method for overcoming these computational bottlenecks.

2.1. Preliminaries. The algebraic equations in [18] can be represented in the form

$$\sup_{Q \in Z} \left\{ -\mathbb{A}(Q)V + \mathcal{C}(Q) \right\} = 0, \tag{2.1}$$

with $\mathbb{A}$ an $N \times N$ matrix and $V, \mathcal{C}$ vectors of length N. Here Q is an indexed set of N controls, where each $Q_l \in Z$, with Z the set of admissible controls. We make the following assumption.

Assumption 2.1.

(a) The set of admissible controls Z is compact.

(b) The matrices and vectors have the property that $[\mathbb{A}(Q)]_{\ell,m}$ and $[\mathcal{C}(Q)]_\ell$ depend only on Q_ℓ.

Assumptions (a) and (b) are typically satisfied for discretized HJB equations.

In general, we do not want to assume that the objective function

$$F(Q, V) = -\mathbb{A}(Q)V + \mathcal{C}(Q) \tag{2.2}$$

is a continuous function of the control Q. For example, in order to ensure monotonicity when discretizing HJB equations, one often uses central/upstream differencing, with central differencing used as much as possible [37], which results in a discontinuous objective function. In order to handle the case where $F(Q, V)$ is a discontinuous function of Q, we make use of its upper semicontinuous envelope. If the ith row of $F(Q, V)$ is given by

$$[F(Q, V)]_i = -\sum_j \mathbb{A}_{i,j}(Q_i)V_j + \mathcal{C}_i(Q_i), \tag{2.3}$$

then the upper semicontinuous envelope $\bar{F}(Q, V)$ for fixed V is given by ($\forall Q_i \in Z$)

$$\left[\bar{F}(Q, V)\right]_i = \limsup_{\substack{q \to Q_i \\ q \in Neigh(Q_i)}} \left\{ -\sum_j \mathbb{A}_{i,j}(q)V_j + \mathcal{C}_i(q) \right\} \equiv -\sum_j \mathbb{A}^*_{i,j}(Q_i, V)V_j + \mathcal{C}^*_i(Q_i, V), \tag{2.4}$$

where $Neigh(Q_i)$ is a closed neighborhood of Q_i (but containing Q_i). For a fixed V, the upper semicontinuous envelope is determined by the coefficients $\mathbb{A}$ and $\mathcal{C}$. As an example, if Q_i contains a single control, with Z a compact subset of $\mathbb{R}$, then

$$-\sum_j \mathbb{A}^*_{i,j}(Q_i, V)V_j + \mathcal{C}^*_i(Q_i, V) \equiv \max \begin{cases} \lim\limits_{q \to Q_i^+} -\sum\limits_j \mathbb{A}_{i,j}(q)V_j + \mathcal{C}_i(q), \\ \lim\limits_{q \to Q_i^-} -\sum\limits_j \mathbb{A}_{i,j}(q)V_j + \mathcal{C}_i(q), \\ \qquad -\sum_j \mathbb{A}_{i,j}(Q_i)V_j + \mathcal{C}_i(Q_i). \end{cases} \tag{2.5}$$

Note that since Z is compact, then $\mathbb{A}$ and $\mathcal{C}$ are related to $\mathbb{A}^*$ and $\mathcal{C}^*$ by

$$\sup_{Q\in Z}\left\{-\mathbb{A}(Q)V+\mathcal{C}(Q)\right\}=\max_{Q\in Z}\left\{-\mathbb{A}^*(Q,V)V+\mathcal{C}^*(Q,V)\right\}. \tag{2.6}$$

As such (2.1) is interpreted as

$$\begin{aligned}&\mathbb{A}^*(\hat{Q},V)V=\mathcal{C}^*(\hat{Q},V)\\ &\text{with}\quad \hat{Q}_\ell=\arg\max_{Q\in Z}\left[-\mathbb{A}^*(Q,V)V+\mathcal{C}^*(Q,V)\right]_\ell.\end{aligned} \tag{2.7}$$

Remark 2.1 (Z a finite set). If the set of admissible controls is a finite set, then trivially $\mathbb{A}^*(Q,V)=\mathbb{A}(Q)$ and $\mathcal{C}^*(Q,V)=\mathcal{C}(Q)$.

The following will be needed in the next section.

LEMMA 2.1. *Suppose* $Q^Y\in\arg\max_{Q\in Z}\{-\mathbb{A}^*(Q,Y)Y+\mathcal{C}^*(Q,Y)\}$. *Then for any control* $\hat{Q}$ *and vector* $\hat{Y}$ *we have*

$$-\mathbb{A}^*(Q^Y,Y)Y+\mathcal{C}^*(Q^Y,Y)\geq-\mathbb{A}^*(\hat{Q},\hat{Y})Y+\mathcal{C}^*(\hat{Q},\hat{Y}). \tag{2.8}$$

Proof. The result follows from (2.6) coupled with the inequalities

$$\begin{aligned}&-\mathbb{A}^*(Q^Y,Y)Y+\mathcal{C}^*(Q^Y,Y)\\ &\quad=\sup_{Q\in Z}\left\{-\mathbb{A}(Q)Y+\mathcal{C}(Q)\right\}\geq\limsup_{Q\to\hat{Q};\,Q\in Neigh(\hat{Q})}\left\{-\mathbb{A}(Q)Y+\mathcal{C}(Q)\right\}\\ &\quad\geq-\mathbb{A}^*(\hat{Q},\hat{Y})Y+\mathcal{C}^*(\hat{Q},\hat{Y})\end{aligned}$$

for a given $\hat{Q}$ and $\hat{Y}$. Here $Neigh(\hat{Q})$ is a closed neighborhood of $\hat{Q}$. ∎

2.2. Policy iteration. Policy iteration is a well-known iterative method for solving problems of type (2.7) [21, 7]. Let V^k denote the kth estimate for V (starting at V^0). The policy iteration approach for solving (2.7) is given in Algorithm 2.1.

ALGORITHM 2.1 Policy iteration

$V^0=$ Initial solution vector of size N ; given *scale* >0, *tolerance* >0
for $k=0,1,2,\ldots$ until converge **do**

$$Q^k_\ell=\arg\max_{Q\in Z}\left\{-\mathbb{A}^*(Q,V^k)V^k+\mathcal{C}^*(Q,V^k)\right\}_\ell$$

Solve the linear system

$$\mathbb{A}^*(Q^k,V^k)V^{k+1}=\mathcal{C}^*(Q^k,V^k)$$

if $k\geq 0$ and $\left(\max_\ell\dfrac{|V_\ell^{k+1}-V_\ell^k|}{\max\left[scale,|V_\ell^{k+1}|\right]}<tolerance\right)$ **then**
break from the iteration
end if
end for

The term *scale* in Algorithm 2.1 is used to ensure that unrealistic levels of accuracy are not required when the value is very small.

There are several possibilities for solving the linear system in the policy iteration method. For example, if $\mathbb{A}^*$ is sparse, then direct or iterative methods (such as preconditioned GMRES [31]) can be used.

2.3. Splitting methods. It is not always the case that one can easily solve the policy iteration matrix $\mathbb{A}^*(Q^k, V^k)$. To this end, we form the splitting $\mathbb{A}^* = \mathcal{A}^* - \mathcal{B}^*$, so that our algebraic equations will now be written as

$$
\begin{aligned}
&(\mathcal{A}^*(Q,V) - \mathcal{B}^*(Q,V))\ V = \mathcal{C}^*(Q,V) \\
&\text{with} \quad Q_\ell = \underset{Q \in Z}{\arg\max} \left[-\Big(\mathcal{A}^*(Q,V) - \mathcal{B}^*(Q,V) \Big) V + \mathcal{C}^*(Q,V) \right]_\ell .
\end{aligned} \tag{2.9}
$$

We assume that this splitting is such that any linear system having $\mathcal{A}^*(Q,V)$ as its coefficient matrix is easy to solve.

2.4. Simple iteration. Using the above notation, at each step of full policy iteration we solve

$$
\Big(\mathcal{A}^*(Q^k,V^k) - \mathcal{B}^*(Q^k,V^k) \Big) V^{k+1} = \mathcal{C}^*(Q^k,V^k). \tag{2.10}
$$

However, as discussed above, it may be very costly to solve (2.10). An obvious alternative is to use an iterative method. If $(V^{k+1})^m$ is the mth estimate for V^{k+1}, then simple iteration for solution of the linear system (2.10) is

$$
\mathcal{A}^*(Q^k,V^k) \left(V^{k+1} \right)^{m+1} = \mathcal{B}^*(Q^k,V^k) \left(V^{k+1} \right)^{m} + \mathcal{C}^*(Q^k,V^k). \tag{2.11}
$$

2.5. Fixed point policy iteration. Instead of solving the linear system to convergence using simple iteration, it is natural to ask whether it suffices to use only a single simple iteration at each nonlinear iterate. In this case we replace policy iteration with what we refer to as fixed point policy iteration.

Algorithm 2.2 Fixed point policy iteration

$V^0 =$ Initial solution vector of size N
for $k = 0, 1, 2, \ldots$ until converge **do**

$$
Q^k_\ell = \underset{Q \in Z}{\arg\max} \left[-\mathcal{A}^*(Q,V^k)V^k + \mathcal{B}^*(Q,V^k)V^k + \mathcal{C}^*(Q,V^k) \right]_\ell
$$

Solve the linear system

$$
[\mathcal{A}^*(Q^k,V^k)]V^{k+1} = \mathcal{B}^*(Q^k,V^k)V^k + \mathcal{C}^*(Q^k,V^k)
$$

if converged **then**
break from the iteration
end if
end for

The above method requires only the solution of the sparse matrix $\mathcal{A}^*(Q^k,V^k)$ and a matrix-vector multiply $\mathcal{B}^*(Q^k,V^k)V^k$ at each nonlinear iteration.

3. Convergence of the fixed point policy iteration. In [16], the convergence of an iterative scheme for a penalty formulation for American options under a jump diffusion process was proven. This same idea was generalized for other HJB problems in [12]. While it is possible to use this approach to prove convergence of scheme (2.2), these proofs are algebraically complex. In the following, we will present a simpler and more general method which proves convergence of Algorithm 2.2.

In order to ensure convergence of our scheme we need to make some basic assumptions which hold for the applications that are of interest.

Condition 3.1. The matrices $\mathcal{A}^*(Q,V), \mathcal{B}^*(Q,V)$ and vector $\mathcal{C}^*(Q,V)$ satisfy the following:

(i) The matrices $\mathcal{A}^*(Q,V)$ and $\mathcal{A}^*(Q,V) - \mathcal{B}^*(Q,V)$ are M matrices.

(ii) The matrices $\mathcal{A}^*(Q,V), \mathcal{B}^*(Q,V)$, the vector $\mathcal{C}^*(Q,V)$, and $\|(\mathcal{A}^*)^{-1}(Q,V)\|_\infty$ are bounded, independent of Q, V.

(iii) There is a constant $C_1 < 1$ such that

$$\|\mathcal{A}^*(Q^k,V^k)^{-1} \cdot \mathcal{B}^*(Q^{k-1},V^{k-1})\|_\infty \le C_1 \quad \text{and } \|\mathcal{A}^*(Q^k,V^k)^{-1} \cdot \mathcal{B}^*(Q^k,V^k)\|_\infty \le C_1. \tag{3.1}$$

Remark 3.1. We remind the reader that a matrix $\mathcal{A}^*$ is an M matrix if the offdiagonals are nonpositive, $\mathcal{A}^*$ is nonsingular, and $(\mathcal{A}^*)^{-1} \ge 0$. A sufficient condition for a matrix to be an M matrix is that the offdiagonals are nonpositive, and each row sum is strictly positive [35]. We will use this result in the following.

Remark 3.2. In order to ensure convergence, the discretizations of our financial problems as in (2.7) need to be monotone, consistent, and ℓ_∞ stable. This requires a positive coefficient discretization resulting in the M matrices of (i) and bounded matrices $\mathcal{A}^*(Q,V)$, $\mathcal{B}^*(Q,V)$ and vector $\mathcal{C}^*(Q,V)$.

Before proving the main result of this section, it will be helpful to note the following proposition and lemmas.

Proposition 3.1 (convergent sequence). *Given a bounded infinite sequence (v_n), such that*

$$v_{k+1} \ge v_k - \alpha\beta^k, \tag{3.2}$$

where $\alpha > 0$ is a constant independent of k and $|\beta| < 1$, then the sequence converges.

Proof. This is a simple case of a result found in [8]. Property (3.2) implies that for any $q > p$ we have

$$v_p \le v_q + \sum_{k=p}^{q-1} \alpha\beta^k. \tag{3.3}$$

Let $s = \liminf v_n$. Then for any $\epsilon > 0$ and any q the definition of lim inf implies that there exists $q^* > q$ such that $v_{q^*} < s + \epsilon$, and so

$$v_p < s + \epsilon + \sum_{k=p}^{q^*-1} \alpha\beta^k \le s + \epsilon + \sum_{k=p}^{\infty} \alpha\beta^k. \tag{3.4}$$

Hence $v_p \le s + \sum_{k=p}^{\infty} \alpha\beta^k$, and so

$$\limsup_p v_p \le s + \lim_{N\to\infty} \sum_{k=N}^{\infty} \alpha\beta^k = s = \liminf_p v_p. \tag{3.5}$$

Since v_p is bounded from above, we obtain convergence to a finite value. ◻

Lemma 3.2 (bounded iterates). *Let matrices $\mathcal{A}^*(Q,V), \mathcal{B}^*(Q,V)$ and vector $\mathcal{C}^*(Q,V)$ satisfy Condition 3.1. Then $\|V^k\|_\infty$ is bounded independent of k.*

Proof. From Algorithm 2.2 we have

$$\|V^{k+1}\|_\infty \le \|\mathcal{A}^*(Q^k,V^k)^{-1}\mathcal{B}^*(Q^k,V^k)\|_\infty \|V^k\|_\infty + \|\mathcal{A}^*(Q^k,V^k)^{-1}\mathcal{C}^*(Q^k,V^k)\|_\infty \le C_1\|V^k\|_\infty + C_2 \tag{3.6}$$

for some constant C_2 independent of k. Iterating (3.6) gives

$$\|V^{k+1}\|_\infty \le C_1^{k+1}\|V^0\|_\infty + C_2\sum_{i=0}^{k} C_1^i \le \|V^0\|_\infty + \frac{C_2}{1-C_1}, \tag{3.7}$$

which follows since $C_1 < 1$. □

LEMMA 3.3 (uniqueness of solution). *Assume the set of controls satisfies Assumption* 2.1 *and that* $\mathcal{A}^*(Q,V)$, $\mathcal{B}^*(Q,V)$, *and* $\mathcal{C}^*(Q,V)$ *satisfy Condition* 3.1. *If the iterative scheme* (2.2) *converges, then it converges to the unique solution of* (2.9).

Proof. Note first that simple manipulation of method (2.2) gives

$$\begin{aligned} \mathcal{A}^*(Q^k,V^k)(V^{k+1}-V^k) &= -\mathcal{A}^*(Q^k,V^k)V^k + \mathcal{B}^*(Q^k,V^k)V^k + \mathcal{C}^*(Q^k,V^k) \\ &= \sup_{Q\in Z}\Big\{-\mathcal{A}(Q)V^k + \mathcal{B}(Q)V^k + \mathcal{C}(Q)\Big\}. \end{aligned} \tag{3.8}$$

Suppose now that $\lim_{k\to\infty} V^k = V^\infty$. Then $\lim_{k\to\infty}\mathcal{A}^*(Q^k,V^k)(V^{k+1}-V^k) = 0$ since $\mathcal{A}^*(Q,V)$ is bounded. Consequently

$$\begin{aligned} 0 &= \lim_{k\to\infty}\sup_{Q\in Z}\Big\{-\mathcal{A}(Q)V^k + \mathcal{B}(Q)V^k + \mathcal{C}(Q)\Big\} \\ &= \sup_{Q\in Z}\Big\{-\mathcal{A}(Q)V^\infty + \mathcal{B}(Q)V^\infty + \mathcal{C}(Q)\Big\}, \end{aligned}$$

since $\sup(\cdot)$ is a continuous function of V^k. Thus V^∞ solves (2.9).

As for uniqueness, suppose there are two solutions X, Y, such that

$$\begin{aligned} \mathbb{A}^*(Q^X,X)X = \mathcal{C}^*(Q^X,X)\ ;\ Q^X \in \arg\max_{Q\in Z}\Big\{-\mathbb{A}^*(Q,X)X + \mathcal{C}(Q,X)\Big\}, \\ \mathbb{A}^*(Q^Y,Y)Y = \mathcal{C}^*(Q^Y,Y)\ ;\ Q^Y \in \arg\max_{Q\in Z}\Big\{-\mathbb{A}^*(Q,Y)Y + \mathcal{C}^*(Q,Y)\Big\}. \end{aligned}$$

The above two equations, along with Lemma 2.1, give

$$\begin{aligned} \mathbb{A}^*(Q^X,X)(X-Y) &= -\mathbb{A}^*(Q^X,X)Y + \mathcal{C}^*(Q^X,X) \\ &\quad - \left[-\mathbb{A}^*(Q^Y,Y)Y + \mathcal{C}^*(Q^Y,Y)\right] \le 0. \end{aligned}$$

This implies $\mathbb{A}^*(Q^X,X)(X-Y)\le 0$. Since $\mathbb{A}^*(Q^X,X)$ is an M matrix, $X-Y\le 0$. Interchanging X and Y also gives $(Y-X)\le 0$, and hence $X=Y$. □

Remark 3.3. Similar uniqueness results (assuming continuous $\mathbb{A}(Q)$) are given in, for example, [8, 25].

THEOREM 3.4 (convergence of scheme). *If the matrices* $\mathcal{A}^*(Q,V),\mathcal{B}^*(Q,V)$ *and vector* $\mathcal{C}^*(Q,V)$ *satisfy Condition* 3.1, *then the scheme* (2.2) *converges to the unique solution of* (2.9) *for any initial iterate* V^k.

Proof. Algorithm 2.2 can be written as

$$\begin{aligned} \mathcal{A}^*(Q^k,V^k)(V^{k+1}-V^k) &= \mathcal{B}^*(Q^{k-1},V^{k-1})(V^k - V^{k-1}) \\ &\quad -\mathcal{A}^*(Q^k,V^k)V^k + \mathcal{B}^*(Q^k,V^k)V^k + \mathcal{C}^*(Q^k,V^k) \\ &\quad - \left[-\mathcal{A}^*(Q^{k-1},V^{k-1})V^k + \mathcal{B}^*(Q^{k-1},V^{k-1})V^k\right. \\ &\quad \left. + \mathcal{C}^*(Q^{k-1},V^{k-1})\right] \\ &= \mathcal{B}^*(Q^{k-1},V^{k-1})(V^k - V^{k-1}) \end{aligned}$$

$$
\begin{aligned}
& -\mathbb{A}^*(Q^k, V^k)V^k + \mathcal{C}^*(Q^k, V^k) \\
& - \left[-\mathbb{A}^*(Q^{k-1}, V^{k-1})V^k + \mathcal{C}^*(Q^{k-1}, V^{k-1})\right] \\
& \geq \mathcal{B}^*(Q^{k-1}, V^{k-1})(V^k - V^{k-1}) \qquad (3.9)
\end{aligned}
$$

where the last inequality follows from Lemma 2.1. Equations (3.9) combined with the fact that $\mathcal{A}^*(Q^k, V^k)$ is an M matrix then implies

$$
V^{k+1} - V^k \geq \left[\mathcal{A}^*(Q^k, V^k)^{-1}\mathcal{B}^*(Q^{k-1}, V^{k-1})\right](V^k - V^{k-1}). \qquad (3.10)
$$

From Condition 3.1

$$
\|\mathcal{A}^*(Q^k, V^k)^{-1}\mathcal{B}^*(Q^{k-1}, V^{k-1})\|_\infty \leq C_1 < 1, \qquad (3.11)
$$

and so we have

$$
\left(V^{k+1} - V^k\right) \geq -C_1^k \|V^1 - V^0\|_\infty \, \mathbf{e}, \qquad (3.12)
$$

where $\mathbf{e} = [1, 1, \ldots, 1]'$. Let $C_3 = \|V^1 - V^0\|_\infty$. Then in component form we have

$$
[V^{k+1}]_\ell \geq [V^k]_\ell - C_1^k C_3. \qquad (3.13)
$$

From Lemma 3.2, the sequence V_i^{k+1} is bounded, hence the iteration converges from Proposition 3.1. In the limit, the iteration converges to the unique solution of (2.9) from Lemma 3.3. □

Remark 3.4 (monotone convergence). We can eliminate condition (3.11) if we require that $(V^1 - V^0) \geq 0$ and $\mathcal{B}(Q) \geq 0$, since then the iteration will generate a monotone nondecreasing sequence from (3.10). Tests in [16] show that enforcing monotone convergence using a special choice for the first iterate converges more slowly than using the natural choice of the solution from the previous step. In addition, numerical experiments indicate that floating point errors are amplified if condition (3.11) is violated, and hence the sequence V^k may not be nondecreasing even if $(V^1 - V^0) \geq 0$.

Remark 3.5 (previous work). Various forms of modified policy iteration have been suggested in the context of infinite horizon Markov chain problems [25]. However, convergence results in [30] require that the initial iterate be selected so as to enforce monotone convergence (as in Remark 3.4). Moreover, we do not require that $\mathcal{A}(Q), \mathcal{B}(Q), \mathcal{C}(Q)$ be continuous functions of the control Q [37].

Condition 3.1 requires bounding a matrix norm of the form

$$
\begin{aligned}
\|A^{-1}B\|_\infty &= \max_{y \neq 0} \frac{\|A^{-1}By\|_\infty}{\|y\|_\infty} \\
&= \max_{y \neq 0} \frac{\|x\|_\infty}{\|y\|_\infty}, \qquad \text{where } Ax = By \qquad (3.14)
\end{aligned}
$$

with A an M matrix. The following will be useful in this regard.

PROPOSITION 3.5. *Suppose $Ax = By$ with A a strictly diagonally dominant M matrix and $B \geq 0$. Then for any ℓ such that $|x_\ell| = \|x\|_\infty$ we have*

$$
\left(\sum_u A_{\ell,u}\right) \|x\|_\infty \leq \left(\sum_u B_{\ell,u}\right) \|y\|_\infty. \qquad (3.15)
$$

Proof. Since $Ax = By$ we have

$$A_{\ell,\ell}x_\ell = -\sum_{u\neq\ell} A_{\ell,u}x_u + \sum_u B_{\ell,u}y_u. \tag{3.16}$$

Taking absolute values on both sides and using the fact that $A_{\ell,u}$ is nonpositive whenever $u \neq \ell$ we have that

$$A_{\ell,\ell}|x_\ell| \leq -\Big(\sum_{u\neq\ell} A_{\ell,u}\Big)\|x\|_\infty + \Big(\sum_u B_{\ell,u}\Big)\|y\|_\infty. \tag{3.17}$$

The result follows since $|x_\ell| = \|x\|_\infty$. ◻

4. Guaranteed minimum withdrawal benefit: Jump diffusion. In this and the following sections we give two examples from computational finance: the guaranteed minimum withdrawal benefit (GMWB) insurance contract and an American option pricing problem.

4.1. Singular control formulation of the GMWB problem. A variable annuity policy is a financial contract between a policyholder and an insurance company which promises a stream of cash flows. For a given initial lump sum payment an insurance company creates an investor risky asset account and guarantees a stream of cash flows. The latter payments come from a second, virtual, guarantee account. The payments are variable, depending on the performance of the risky asset account, with some lower bound. Often these variable annuities have GMWBs which allow the policy holder to cumulatively withdraw at least the total amount originally invested. The control parameter in this case is the withdrawal rate.

We extend the singular control formulation for pricing GMWBs in [14] by assuming that the investor's risky asset account W_t follows a finite activity jump diffusion process (in the risk neutral measure). Thus we have

$$\begin{aligned} dW_t &= (r-\eta-\lambda\rho)W_{t^-}\,dt + \sigma W_{t^-}\,dZ \\ &\quad + W_{t^-}\,d\Big(\sum_{i=1}^{\pi_t}(\xi_i - 1)\Big) + dA \quad \text{if } W_{t^-} > 0, \end{aligned} \tag{4.1}$$

$$dW_t = 0 \quad \text{if } W_{t^-} = 0, \tag{4.2}$$

where Z is a Brownian motion, and t^- denotes the instant immediately before t. In addition π_t is a Poisson process with intensity $\lambda > 0$, and ξ_i are positive random variables representing jump amplitudes, with distribution $p(\cdot)$. The processes Z, π_t, ξ_i are assumed to be independent. A is the investor's virtual guarantee withdrawal account. In the above r is the risk-free rate, σ is the volatility, and η the fee charged for the guarantee. We assume that the various ξ follow a log-normal distribution $p(\xi)$ given by

$$p(\xi) = \frac{1}{\sqrt{2\pi}\zeta\xi}\exp\left(-\frac{(\log(\xi)-\nu)^2}{2\zeta^2}\right), \tag{4.3}$$

with parameters ζ and ν, $\rho = E[\xi - 1]$, where $E[\cdot]$ is the expectation, and $E[\xi] = \exp(\nu + \zeta^2/2)$ given the distribution function $p(\xi)$ in (4.3).

For the investor's virtual guarantee account A, let $\gamma \equiv \gamma(t)$ denote the withdrawal rate at time t with $\gamma \in [0,\infty)$. Here an infinite withdrawal rate corresponds to an instantaneous withdrawal of a finite amount. The policy guarantees that the sum of withdrawals throughout the policy's life is equal to the premium paid up front, which is denoted by ω_0. As a result, we have $A(0) = \omega_0$, and

$$A(t) = \omega_0 - \int_0^t \gamma(u)du, \quad A(t) \geq 0. \tag{4.4}$$

We assume that we are dealing with a GMWB having a cap on the maximum allowed withdrawal rate without penalty. If G is the contractual withdrawal rate and $\kappa < 1$ is the proportional penalty charge applied on the portion of the withdrawal exceeding G, then the net withdrawal rate $f(\gamma)$ received by the policy holder is

$$f(\gamma) = \begin{cases} \gamma, & 0 \leq \gamma \leq G, \\ G + (1-\kappa)(\gamma - G), & \gamma > G. \end{cases} \tag{4.5}$$

Define $\tau = T - t$, where t is the forward time and T is the expiry time of the contract, and set $V = V(W, A, \tau)$ to be the no-arbitrage value of the guarantee. Generalizing the formulation in [26, 14, 22] to the case with stochastic process (4.1), the value of the guarantee is given from the solution to the following singular control problem:

$$\min\left[\ V_\tau - \mathcal{L}V - \lambda\mathcal{J}V - G\max(\mathcal{F}V, 0)\ ,\ \ \kappa - \mathcal{F}V\ \right] = 0. \tag{4.6}$$

Here the operators $\mathcal{L}, \mathcal{F}, \mathcal{J}$ are defined as

$$\begin{aligned} \mathcal{L}V &= \frac{\sigma^2}{2}W^2 D_{WW}V + (r - \eta - \lambda\rho)WD_W V - (r+\lambda)V, \\ \mathcal{F}V &= 1 - V_W - V_A = 1 - D_W V - D_A V, \\ \mathcal{J}V &= \int_0^\infty V(\xi W, A, \tau)p(\xi)\ d\xi, \end{aligned} \tag{4.7}$$

while D_A, D_W, and D_{WW} denote the usual partial derivative operators. Problem (4.6) is solved on the computational domain

$$(W, A, \tau) \in [0, W_{\max}) \times [0, \omega_0] \times [0, T]. \tag{4.8}$$

At expiry time $\tau = 0$, the value of the contract is

$$V(W, A, \tau = 0) = \max\Big[W, (1-\kappa)A\Big]. \tag{4.9}$$

Other boundary conditions are

$$\begin{aligned} \min\Big[V_\tau - rV - G\max(1 - V_A, 0), \kappa - (1 - V_A)\Big] &= 0,\ W = 0, \\ V(W_{\max}, A, \tau) &= e^{-\eta\tau}W_{\max}\,,\ \ W = W_{\max}, \\ V_{WW} &\to 0\,,\ \ W \to W_{\max}, \\ V_\tau &= \mathcal{L}V - \lambda\mathcal{J}V,\ \ A = 0. \end{aligned} \tag{4.10}$$

No boundary condition is required at $A = \omega_0$. For details concerning the derivation of (4.6), we refer readers to [26, 9, 11, 14, 22].

As discussed in [14, 22], we can reformulate problem (4.6) in penalized form as

$$V_\tau^\varepsilon = \mathcal{L}V^\varepsilon + \lambda \mathcal{J}V^\varepsilon + \max_{\substack{\varphi\in\{0,1\},\psi\in\{0,1\}\\ \varphi\psi=0}} \left[\varphi G \mathcal{F} V^\varepsilon + \psi\left(\frac{(\mathcal{F}V^\varepsilon - \kappa)}{\varepsilon} + \kappa G\right)\right]. \tag{4.11}$$

The basic idea of the penalty method is to discretize (4.11) and let $\varepsilon \to 0$ as the mesh and timesteps tend to zero. In the case of no jumps ($\lambda = 0$), then it is shown in [22] that this will converge to the viscosity solution of (4.6) as the mesh and timesteps vanish.

Note that we are working here in an incomplete market, so that the equivalent martingale pricing measure is not in general unique. As in [3], in practice the parameters of (4.11) are obtained by calibration to traded prices of options. This means that the parameters of (4.11) correspond to those from the market's pricing measure.

4.2. Discretization of the GMWB problem. In order to solve the singular control problem from the last subsection we discretize our problem over a finite grid in the $W \times A$ plane. Define a set of nodes in the W direction $\{W_1, W_2, \ldots, W_{i_{\max}}\}$ and in the A direction $\{A_1, A_2, \ldots, A_{j_{\max}}\}$. Denote the nth timestep by $\tau^n = n\Delta\tau$ and let $V_{i,j}^n$ be the approximate solution of (4.11) at (W_i, A_j, τ^n). Let $\mathcal{L}^h, \mathcal{J}^h, \mathcal{F}^h, D_W^h, D_A^h$ be the discrete forms of the operators $\mathcal{L}, \mathcal{J}, \mathcal{F}, D_W, D_A$, respectively. We discretize (4.11) using fully implicit timestepping and central, forward, and backward differencing so that the positive coefficient condition is satisfied [37, 18, 22]. For efficiency, central differencing is used as much as possible [37].

The final discretized equations then become

$$\begin{aligned} &V_{i,j}^{n+1} - \Delta\tau \mathcal{L}^h V_{i,j}^{n+1} + \varphi_{i,j}^{n+1} G[D_A^h V_{i,j}^{n+1} + D_W^h V_{i,j}^{n+1}]\Delta\tau \\ &\quad + \frac{\psi_{i,j}^{n+1}}{\varepsilon}[D_A^h V_{i,j}^{n+1} + D_W^h V_{i,j}^{n+1}]\Delta\tau \\ &\quad = \varphi_{i,j}^{n+1} G \Delta\tau + \psi_{i,j}^{n+1}\Delta\tau\left[\frac{1-\kappa}{\varepsilon} + \kappa G\right] + \lambda\Delta\tau[\mathcal{J}^h V^{n+1}]_{i,j} + V_{i,j}^n, \end{aligned} \tag{4.12}$$

where

$$\begin{aligned} \{\varphi_{i,j}^{n+1}, \psi_{i,j}^{n+1}\} \in \operatorname*{arg\,max}_{\substack{\varphi\in\{0,1\},\psi\in\{0,1\}\\ \varphi\psi=0}} &\Bigg\{\varphi\ G[1 - D_A^h V_{i,j}^{n+1} - D_W^h V_{i,j}^{n+1}] \\ &+ \psi\left[\frac{1 - D_A^h V_{i,j}^{n+1} - D_W^h V_{i,j}^{n+1} - \kappa}{\varepsilon} + \kappa G\right]\Bigg\}. \end{aligned} \tag{4.13}$$

As discussed in [22], $\varepsilon = C_4 \Delta\tau$, where C_4 is a constant. The boundary conditions in this case translate into the discrete equations

$$V_{i_{\max},j} = e^{-\eta\tau^n} W_{\max}. \tag{4.14}$$

The integral term $\mathcal{J}V$ is discretized via transformation into a correlation integral combined with a use of the midpoint rule as described in detail in [17].

4.3. Associated general linear form. Let $N = i_{\max} \times j_{\max}$ be the size of the grid and set

$$V^n = [V_{1,1}^n, \ldots, V_{i_{\max},1}^n, \ldots, V_{1,j_{\max}}^n, \ldots, V_{i_{\max},j_{\max}}^n]'. \tag{4.15}$$

We can represent the linear relationships given in (4.12) in matrix form as follows. Define square $N \times N$ matrices $\mathcal{A}, \mathcal{B}$ and a vector $\mathcal{C}$ of size N by

$$\begin{aligned} [\mathcal{A}(\varphi_\ell^k, \psi_\ell^k)U]_\ell &= [\mathcal{A}^k U]_\ell = U_\ell - \Delta\tau \mathcal{L}^h U_\ell + \varphi_\ell^k G[D_A^h U_\ell + D_W^h U_\ell]\Delta\tau \\ &\quad + \frac{\psi_\ell^k}{\varepsilon}[D_A^h U_\ell + D_W^h U_\ell]\Delta\tau, \\ \big[\mathcal{B}(\varphi_\ell^k, \psi_\ell^k)U\big]_\ell &= [\mathcal{B}^k U]_\ell = \lambda \Delta\tau [\mathcal{J}^h U]_\ell, \\ \mathcal{C}(\varphi_\ell^k, \psi_\ell^k)_\ell &= C_\ell^k = \varphi_\ell^k G \Delta\tau + \psi_\ell^k \left[\frac{(1-\kappa)}{\varepsilon} + \kappa G\right]\Delta\tau + V_\ell^n \end{aligned} \tag{4.16}$$

with controls

$$\{\varphi_\ell^k, \psi_\ell^k\} \in \underset{\substack{\varphi_\ell \in \{0,1\}, \psi_\ell \in \{0,1\} \\ \varphi_\ell \psi_\ell = 0}}{\arg\max} \big[-\mathcal{A}(\varphi_\ell, \psi_\ell)U^k + \mathcal{B}(\varphi_\ell, \psi_\ell)U^k + \mathcal{C}(\varphi_\ell, \psi_\ell)\big]_\ell . \tag{4.17}$$

If we write U and Q as

$$\begin{aligned} U &= [U_{1,1}, \ldots, U_{i_{\max},1}, \ldots, U_{1,j_{\max}}, \ldots, U_{i_{\max},j_{\max}}]', \\ Q &= [q_{1,1}, \ldots, q_{i_{\max},1}, \ldots, q_{1,j_{\max}}, \ldots, q_{i_{\max},j_{\max}}]' \end{aligned} \tag{4.18}$$

with $q_{i,j}$ coming from the set

$$\big\{(\varphi, \psi) \;\mid\; \varphi \in \{0,1\}, \psi \in \{0,1\}, \varphi\psi = 0\big\}, \tag{4.19}$$

then the discretized equations (4.12) become

$$\sup_{Q \in Z}\left\{-\mathcal{A}(Q)V^{n+1} + \mathcal{B}(Q)V^{n+1} + \mathcal{C}(Q)\right\} = 0. \tag{4.20}$$

Remark 4.1. Notice that any vector index $1 \le \ell \le N$ corresponds to a grid node (i, j) via

$$\ell = i + (j-1)i_{\max} \text{ with } 1 \le i \le i_{\max} \text{ and } 1 \le j \le j_{\max}. \tag{4.21}$$

Recall that in order to ensure convergence to the viscosity solution of (4.6), the discretization must be monotone, consistent, and l_∞ stable [5]. A positive coefficient discretization guarantees monotonicity [18]. The positive coefficient condition can be defined in terms of the matrices $\mathcal{A}, \mathcal{B}$ as follows.

DEFINITION 4.1 (positive coefficient condition). *A positive coefficient discretization generates matrices* $\mathcal{A}, \mathcal{B}$ *having the properties*

$$(\mathcal{A} - \mathcal{B})_{\ell,m} \begin{cases} > 0, & \ell = m, \\ \le 0, & \ell \ne m. \end{cases}$$

Remark 4.2. The discretization of the jump term $\mathcal{J}V$ (4.7) as in [17] results in a dense matrix $\mathcal{B}$. However, the method of discretization used in that paper implies that vector product $\mathcal{B}V^n$ can be computed efficiently in $O(N \log N)$ operations using a fast Fouries transform (FFT).

PROPOSITION 4.2. *Suppose a positive coefficient discretization (Definition* 4.1*) is used and the jump operator* $\mathcal{J}^h$ *is discretized using the method in* [17]*. Then the following hold*:

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

(b) *Suppose row ℓ corresponds to grid node (i, j) as in* (4.21). *Then the ℓth row sums for $\mathcal{A}(Q^k)$ and $\mathcal{B}(Q^k)$ are*

$$Row\_Sum_{\ \ell}\ (\mathcal{A}(Q^k)) = \begin{cases} 1 + (r+\lambda)\Delta\tau, & 1 < i < i^*, \\ 1 + r\Delta\tau, & i = 1,\ i = i^*, \dots, i_{\max} - 1, \\ 1, & i = i_{\max}, \end{cases}$$

$$Row\_Sum_{\ \ell}\ (\mathcal{B}(Q^k)) \leq \begin{cases} \lambda\Delta\tau, & 1 < i < i^*, \\ 0, & \textit{otherwise}. \end{cases} \tag{4.22}$$

Here linear behavior of the solution is assumed for $i \geq i^*$ [17].

(c) *The matrices $(\mathcal{A}(Q) - \mathcal{B}(Q))$ and $\mathcal{A}(Q)$ in* (4.20) *are strictly diagonally dominant M matrices.*

Proof. The construction of $\mathcal{B}(Q^k)$, using the discretization of $\mathcal{J}V$ as detailed in [17], implies that

$$\sum_{\mu} [\mathcal{J}^h]_{\ell,\mu} \leq 1 \text{ and } [\mathcal{J}^h]_{\ell,\mu} \geq 0. \tag{4.23}$$

This holds since $p(\xi)$ in (4.7) is a probability density function. When the grid node (i, j) satisfies $i > i^*$, then the ℓth row of $\mathcal{B}(Q^k)$ is identically zero. This gives (a) and the second part of (b).

In order to prove the remaining part of (b) we note that the row sum is the same as $[\mathcal{A}(Q^k)e]_\ell$ with $e = [1, \dots, 1]'$. Since $D^h_{WW} 1 = D^h_W 1 = D^h_A 1 = 0$ we see that $\mathcal{L}^h 1 = -(r+\lambda)$. Thus $[\mathcal{A}(Q^k)e]_\ell = 1 + (r+\lambda)\Delta\tau$ for $1 < i < i^*$. A similar argument shows that $[\mathcal{A}(Q^k)e]_\ell = 1 + r\Delta\tau$ for $i = 1; i^* \leq i < i_{\max}$. When $i = i_{\max}$ then the corresponding row is just the ℓth identity row (since it is just a boundary assignment) and hence its row sum is just unity. Statement (c) follows since the offdiagonals of $\mathcal{A}(Q) - \mathcal{B}(Q)$ and $\mathcal{A}(Q)$ are nonpositive (since the discretization is monotone [18]), and from (b) the row sums are strictly positive. ∎

Remark 4.3 (efficient implementation). It is interesting to observe that in order to ensure a positive coefficient discretization, the D^h_A operator in (4.12) is always backward differenced. As a result, the solution for $V^{n+1}_{i,j}$ for fixed j depends only on $V^{n+1}_{i,j-1}$. An efficient implementation using this idea is described precisely in [22].

5. Regime switching: American options. A second method for extending geometric Brownian motion (GBM) is by use of a regime switching model. Regime switching models have been applied to insurance [20], electricity markets [19, 36], natural gas [2], and optimal forestry management [10].

This is considered to better model observed risky asset stochastic processes [20], particularly for options having a longer time frame. It also has the useful property of being computationally inexpensive when compared to a full stochastic volatility jump diffusion model. In this section we also show that our methods can be used with both fully implicit or Crank–Nicolson timestepping.

5.1. Modeling American options under regime switching processes. Let $\sigma^j, j = 1, \dots, K$, be a finite set of discrete volatilities for our model. Shifts between these states are controlled by a continuous Markov chain. Under the risk neutral measure, the stochastic process for the underlying asset S in regime j is

$$dS = (r - \rho_j)\ S\ dt + \sigma^j\ S\ dZ + \sum_{k=1}^{K} (\xi_{jk} - 1)\ S\ dX_{jk}, \tag{5.1}$$

where Z is a Brownian motion and X is a continuous K state Markov chain

$$dX_{jk} = \begin{cases} 1 \text{ with probability} & \lambda_{jk}\ dt + \delta_{jk}, \\ 0 \text{ with probability} & 1 - \lambda_{jk}\ dt - \delta_{jk}. \end{cases} \tag{5.2}$$

Here ξ_{jk} are assumed to be nonrandom. It is understood that there can be only one transition over any infinitesimal time interval, and that Z and X are independent. It is also assumed that $\lambda_{jk} \geq 0, j \neq k$. When a transition from $j \to k$ occurs, then the asset price jumps $S \to \xi_{jk} S$. In addition, we define

$$\lambda_{jj} = -\sum_{\substack{k=1 \\ k \neq j}}^{K} \lambda_{jk}, \quad \rho_j = \sum_{\substack{k=1 \\ k \neq j}}^{K} \lambda_{jk} (\xi_{jk} - 1), \quad \lambda_j = \sum_{\substack{k=1 \\ k \neq j}}^{K} \lambda_{jk}. \tag{5.3}$$

For notational completeness, $\xi_{jj} = 1$.

Let $V_j(S, \tau)$ be the no-arbitrage value of our contingent claim in regime j where as usual we have $\tau = T - t$, so we are working backward in time. Define the following differential operators:

$$\begin{aligned} \mathcal{L}_j V_j &= \frac{\sigma_j^2 S^2}{2} D_{SS} V_j + (r - \rho_j) S D_S V_j - (r + \lambda_j) V_j, \\ \mathcal{J}_j V &= \sum_{\substack{k=1 \\ k \neq j}}^{K} \frac{\lambda_{jk}}{\lambda_j} V_k(\xi_{jk} S, \tau). \end{aligned} \tag{5.4}$$

The price of an American option in regime j is then given by [23]

$$\min \left[\ V_{j,\tau} - \mathcal{L}_j V_j - \lambda_j \mathcal{J}_j V, \quad V_j - V^*\ \right], \tag{5.5}$$

where V^* is the payoff. The risk neutral transition densities λ_{jk} are not unique. In practice, we calibrate the parameters in (5.5) to market data, consistent with the market's pricing measure.

5.2. Regime switching: Direct control approach. We can formulate (5.5) as a control problem, as in [8], where we introduce a scaling parameter $\Omega > 0$,

$$\max_{\phi \in \{0,1\}} \left[\Omega\ \phi (V^* - V_j) - (1 - \phi)(V_{j,\tau} - \mathcal{L}_j V_j - \lambda_j \mathcal{J}_j V) \right] = 0. \tag{5.6}$$

Equation (5.6) is discretized on the computational domain $(S, \tau) \in [0, S_{\max}] \times [0, T]$. 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 is

$$V(S, \tau = 0) = V^*(S). \tag{5.7}$$

We truncate any jumps which would require data outside the computational domain. The resulting error is small in regions of interest if $S_{\max}$ is sufficiently large [23].

5.2.1. Discretizing the regime switching direct control formulation. Define a set of nodes $\{S_1, S_1, \ldots, S_{i_{\max}}\}$, and denote the nth timestep by $\tau^n = n\Delta\tau$. Let $V_{i,j}^n$ be the approximate solution of (5.6) at (S_i, τ^n), regime j, and define vectors V^n as in (4.15), that is,

$$V^n = [V_{1,1}^n, \ldots, V_{i_{\max},1}^n, \ldots, V_{1,K}^n, \ldots, V_{i_{\max},K}^n]'. \tag{5.8}$$

Let $\mathcal{L}_j^h, \mathcal{J}_j^h$ be the discrete form of the operators $\mathcal{L}_j, \mathcal{J}_j$. As usual we use central, forward, and backward differencing to ensure a positive coefficient discretization [18], with central differencing used as much as possible. Linear interpolation is used to discretize $\mathcal{J}_j^h$,

$$[\mathcal{J}_j^h V^n]_{i,j} = \sum_{\substack{k=1 \\ k \neq j}}^{K} \frac{\lambda_{jk}}{\lambda_j} I_{i,j,k}^h V^n, \tag{5.9}$$

where $I_{i,j,k}^h V^n \simeq V_k(\min(S_{\max}, \xi_{jk} S_i), \tau^n)$ and

$$I_{i,j,k}^h V^n = w V_{\alpha,k}^n + (1-w) V_{\alpha+1,k}^n, \qquad w \in [0,1]. \tag{5.10}$$

Using fully implicit ($\theta = 1$) or Crank–Nicolson ($\theta = 1/2$) timestepping, the discrete form of (5.6) is then

$$\begin{aligned} (1 - \phi_{i,j}^{n+1}) \left(V_{i,j}^{n+1} - \Delta\tau\theta \mathcal{L}_j^h V_{i,j}^{n+1} \right) + \Omega\ \phi_{i,j}^{n+1} \Delta\tau V_{i,j}^{n+1} \\ = (1 - \phi_{i,j}^{n+1}) V_{i,j}^n + \Omega\ \phi_{i,j}^{n+1} \Delta\tau V_i^* + (1 - \phi_{i,j}^{n+1}) \lambda_j \Delta\tau\theta [\mathcal{J}_j^h V^{n+1}]_{i,j} \\ + (1 - \phi_{i,j}^{n+1})(1-\theta) \left[\Delta\tau \mathcal{L}_j^h V_{i,j}^n + \lambda_j \Delta\tau [\mathcal{J}_j^h V^n]_{i,j} \right], \end{aligned} \tag{5.11}$$

where

$$\begin{aligned} \{\phi_{i,j}^{n+1}\} \in \underset{\phi \in \{0,1\}}{\arg\max} \Bigg\{ \Omega\ \phi (V_i^* - V_{i,j}^{n+1}) - (1-\phi) \Bigg(\frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta\tau} \\ - \theta \big(\mathcal{L}_j^h V_{i,j}^{n+1} + \lambda_j [\mathcal{J}_j^h V^{n+1}]_{i,j} \big) - (1-\theta) \big(\mathcal{L}_j^h V_{i,j}^n + \lambda_j [\mathcal{J}_j^h V^n]_{i,j} \big) \Bigg) \Bigg\}. \end{aligned} \tag{5.12}$$

5.2.2. General form of the direct control regime switching model. Define vectors U as in (4.15) and let matrices $\mathcal{A}, \mathcal{B}$ and vector $\mathcal{C}$ be defined as

$$\begin{aligned} [\mathcal{A}(\phi_\ell^k) U]_\ell &= [\mathcal{A}^k U]_\ell = (1 - \phi_\ell^k) \left(U_\ell - \Delta\tau\theta \mathcal{L}_j^h U_\ell \right) + \phi_\ell^k \Omega\ \Delta\tau U_\ell, \\ \left[\mathcal{B}(\phi_\ell^k) U \right]_\ell &= [\mathcal{B}^k U]_\ell = (1 - \phi_\ell^k) \lambda_j \Delta\tau\theta [\mathcal{J}_j^h V^{n+1}]_\ell, \\ \mathcal{C}(\phi_\ell^k) &= C_\ell^k = (1 - \phi_\ell^k) V_\ell^n + \phi_\ell^k \Omega\ \Delta\tau V_i^* \\ &\quad + (1 - \phi_\ell^k)(1-\theta) \left[\Delta\tau \mathcal{L}_j^h V_\ell^n + \lambda_j \Delta\tau [\mathcal{J}_j^h V^n]_\ell \right], \end{aligned} \tag{5.13}$$

where as before the index ℓ corresponds to the grid node (i, j). Define a vector of controls Q as in (4.18), with $q_\ell = \phi_\ell$, with admissible controls Z,

$$Z_\ell = \big\{ \phi \ \mid \ \phi \in \{0,1\} \big\}. \tag{5.14}$$

The final discretized equations are then in the general form

$$\sup_{Q \in Z} \Big\{ -\mathcal{A}(Q) V^{n+1} + \mathcal{B}(Q) V^{n+1} + \mathcal{C}(Q) \Big\} = 0. \tag{5.15}$$

The positive coefficient condition then results in the following.

PROPOSITION 5.1. *Suppose the discretization* (5.11) *satisfies the positive coefficient condition (see Definition* 4.1*), and linear interpolation is used in* (5.9). *Then the following hold*:

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

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

$$\textit{Row\_Sum}_{\ \ell}\ (\mathcal{A}(Q^k)) = \begin{cases} (1-\phi_\ell^k)\big(1+\theta(r+\lambda_j)\Delta\tau\big) + \phi_\ell^k \Omega\ \Delta\tau, & i < i_{\max}, \\ 1, & i = i_{\max}, \end{cases}$$

$$\textit{Row\_Sum}_{\ \ell}\ (\mathcal{B}(Q^k)) = \begin{cases} (1-\phi_\ell^k)\lambda_j \Delta\tau\theta, & i < i_{\max}, \\ 0, & i = i_{\max}. \end{cases} \tag{5.16}$$

(c) *The matrices $(\mathcal{A}(Q) - \mathcal{B}(Q))$ and $\mathcal{A}(Q)$ in* (5.15) *are strictly diagonally dominant M matrices.*

Proof. In this case part (a) follows from the representation (5.10) since here λ_{jk}, λ_j and the coefficients of I_{ijk}^h are nonnegative for all i, j, k (since $w \in [0,1]$). The row sum of $\mathcal{A}(Q^k)$ also follows as in Proposition 4.2 since again one can see using the operator form that $\mathcal{L}_j^h 1 = -(r+\lambda_j)$ for all j. Thus if $e = [1, \ldots, 1]'$, then $[\mathcal{A}(Q^k)e]_\ell = (1-\phi_\ell^k)(1+\theta(r+\lambda_j)\Delta\tau) + \phi_\ell^k \Omega \Delta\tau$ for $i < i_{\max}$. The row sum of $\mathcal{B}(Q^k)$ is computed using the fact that the representation (5.10) always sums to unity since this adds the coefficients coming from Lagrange interpolation. The case when $i = i_{\max}$ is a consequence of the Dirichlet boundary condition at this node. As in Proposition 4.2, part (c) follows from the use of a positive coefficient discretization, since from (b) the row sums of $(\mathcal{A}(Q) - \mathcal{B}(Q))$ and $\mathcal{A}(Q)$ are strictly positive and the offdiagonals are nonpositive. ∎

6. Verification of Condition 3.1. In this section we show that the previous two problems all satisfy Condition 3.1 (with perhaps a suitable scaling), and hence the fixed point policy iteration scheme converges. For our examples, Z is a finite set, and hence from Remark 2.1, we have that $\mathcal{A}^* = \mathcal{A}$, $\mathcal{B}^* = \mathcal{B}$, and $\mathcal{C}^* = \mathcal{C}$. Therefore we need only verify that Condition 3.1 is valid if we replace $\mathcal{A}^*(Q,V)$, $\mathcal{B}^*(Q,V)$, and $\mathcal{C}^*(Q,V)$, by $\mathcal{A}(Q)$, $\mathcal{B}(Q)$, and $\mathcal{C}(Q)$.

In all cases we need only verify Condition 3.1 (iii) since the property of being strictly diagonally dominant M matrices has been verified in Propositions 4.2 and 5.1. $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are clearly bounded for any finite grid size.

LEMMA 6.1. *If the discretization for the GMWB problem satisfies the conditions required for Proposition* 4.2*, then this discretization satisfies Condition* 3.1.

Proof. For this problem, $\mathcal{B}(Q^k)$ is independent of Q^k, and hence we need only show that

$$\|\mathcal{A}(Q^k)^{-1}\mathcal{B}(Q^k)\|_\infty \leq C_1 \tag{6.1}$$

for some constant $C_1 < 1$. If

$$\mathcal{A}(Q^k)x = \mathcal{B}(Q^k)y, \tag{6.2}$$

then, for the GMWB problem, Proposition 3.5 combined with Proposition 4.2 implies that

$$\frac{\|x\|_\infty}{\|y\|_\infty} \leq \frac{\lambda\Delta\tau}{1+(r+\lambda)\Delta\tau}, \tag{6.3}$$

so that $C_1 < 1$ as required. To prove that $\|\mathcal{A}(Q)^{-1}\|_\infty$ is bounded independently of Q, we repeat the above argument setting $\mathcal{B}$ to the identity matrix. ∎

LEMMA 6.2. *If the discretization of the American option under regime switching, using the direct control method in section* 5.2*, satisfies the preconditions for Proposition* 5.1*, and* $\Omega > \max_j \lambda_j \theta$*, then this discretization satisfies Condition* 3.1*.*

Proof. Suppose

$$\mathcal{A}(Q^k)x = \mathcal{B}(Q^k)y \tag{6.4}$$

and that $|x_\ell| = \|x\|_\infty$ with index ℓ corresponding to node (i, j). If $i < i_{\max}$ and $\phi_\ell = 0$, then Propositions 3.5 and 5.1 imply that

$$\frac{\|x\|_\infty}{\|y\|_\infty} \le \frac{\theta \lambda_j \Delta\tau}{1 + \theta(r + \lambda_j)\Delta\tau}. \tag{6.5}$$

Otherwise when $i = i_{\max}$ or $\phi_\ell = 1$, then $\|\mathcal{B}(Q^k)\|_\infty = 0$ and so $\|x\|_\infty = 0$. In either case bound (6.5) holds giving a constant $C_1 < 1$ satisfying $\|\mathcal{A}(Q^k)^{-1}\mathcal{B}(Q^k)\|_\infty \le C_1$.

Suppose now that

$$\mathcal{A}(Q^k)x = \mathcal{B}(Q^{k-1})y \tag{6.6}$$

and that $|x_\ell| = \|x\|_\infty$ with index ℓ corresponding to grid node (i, j). If $i < i_{\max}$, $\phi_\ell^{k-1} = 0$, and $\phi_\ell^k = 1$, then

$$\frac{\|x\|_\infty}{\|y\|_\infty} \le \frac{\theta \lambda_j}{\Omega}. \tag{6.7}$$

But Ω is an arbitrary scaling of (5.6). Hence we can choose

$$\Omega > \max_j \lambda_j \theta, \tag{6.8}$$

in which case $\frac{\|x\|_\infty}{\|y\|_\infty} \le C_1$ with $C_1 < 1$. In all other cases, $C_1 < 1$ unconditionally. Repeating the above argument setting $\mathcal{B}$ to the identity shows that $\|\mathcal{A}^{-1}(Q)\|_\infty$ is bounded independently of Q. ∎

Remark 6.1 (scaling factor: (5.6)). At first glance, it appears unnatural to introduce an arbitrary scaling factor in (5.6), only to have it be used to satisfy condition (6.7). However, if $\phi_\ell^{k-1} = 0, \phi_\ell^k = 1$, then the units of row ℓ of $\mathcal{A}^k$ and row ℓ of $\mathcal{B}^k$ are not the same. Hence we can violate or satisfy conditions (6.7) simply by rescaling the time units. However, choosing a scaling factor which satisfies conditions (6.7) means that this same scaling factor must be used in the optimization step (5.12) in Algorithm 2.2. Consequently, choosing different scaling factors will result, in general, in different choices for ϕ_ℓ^k at each iteration.

7. Numerical examples.

In this section, several numerical examples are presented using both the fixed point policy iteration scheme in (2.2) and the full policy iteration scheme in algorithm (2.1). The results show that the fixed point policy iteration scheme requires significantly smaller computational cost compared to the full policy scheme.

7.1. GMWB.

The contract parameters from the problem in [11] are given in Table 7.1. Table 7.2 gives the mesh size and timestep parameters. In the localized computational domain, we set $W_{\max} = 1000\omega_0$. The penalty parameter is set to $\varepsilon = \Delta\tau 10^{-2}/\omega_0$ [22].

TABLE 7.1
A sample GMWB contract parameters used in the numerical experiments.

Parameter	Value
Expiry time T	10.0 years
Interest rate r	0.05
Maximum no-penalty withdrawal rate G	10/year
Withdrawal penalty κ	0.10
Initial lump-sum premium ω_0	100
Initial guarantee account balance $A(0)$	100
Initial personal annuity account balance $W(0)$	100
Jump diffusion parameters (ζ, ν, λ)	$(0.45, -0.9, 0.1)$

TABLE 7.2
Grid and timestep data for convergence experiments. At each refinement, new fine grid nodes are introduced between each two coarse grid nodes, and the timesteps are halved.

Refine level	W nodes	A nodes	Timesteps
1	125	111	120
2	249	221	240
3	497	441	480
4	993	881	960
5	1985	1761	1920

Table 7.3 presents the fair insurance fee η charged by the insurance company computed by solving the equation $V(\eta; W = \omega_0, A = \omega_0, \tau = T) = \omega_0$ [22]. Newton iteration is used to solve this equation with the convergence tolerance

$$\frac{|\eta^{k+1} - \eta^k|}{\max(\eta^{k+1}, \eta^k)} < 10^{-8}, \tag{7.1}$$

where η^k is the kth iterate.

Our actual implementation of the nonlinear iteration (2.2) takes advantage of the structure of this problem as described in Remark 4.3. Using fully implicit timestepping, Table 7.4 presents the convergence results for the GMWB value with respect to two volatility values, assuming the no-arbitrage insurance fee is imposed. We compared the fixed point policy (2.2) and full policy iteration scheme (2.1). A simple iteration (2.11) method was used to solve the policy iteration matrix. The nonlinear convergence tolerance for the policy and fixed point policy iteration is given by

$$\max_{\ell} \frac{|\hat{V}_\ell^{k+1} - \hat{V}_\ell^k|}{\max(scale, |\hat{V}_\ell^{k+1}|)} < 10^{-8}. \tag{7.2}$$

A relative update tolerance of 10^{-8} was also used for the simple iteration (2.11).

TABLE 7.3
Convergence study for the fair insurance fee η value with jump diffusions. Contract parameters are given in Table 7.1. Ratio is the ratio of successive changes in the solution as the mesh is refined.

Refine	$\sigma = 0.2$		$\sigma = 0.3$	
level	Fair fee	Ratio	Fair fee	Ratio
1	0.034427	N/A	0.046890	N/A
2	0.032854	N/A	0.045789	N/A
3	0.032439	3.79	0.045536	4.34
4	0.032329	3.78	0.045471	3.91
5	0.032297	3.37	0.045452	3.35

Table 7.4

Iteration and convergence experiments for the GMWB guarantee value at $t = 0$ and $W = A = \omega_0 = 100$ using the fixed point policy and full policy schemes. Contract parameters are given in Table 7.1. Total itns/step refers to the average number of iterations per timestep to solve the equation. Outer itns/step refers to the average number of outer iterations in the full policy iteration scheme. Ratio is the ratio of successive changes in the solution as the mesh/timesteps are refined. Since the fair insurance fee is imposed, the numerical solution should converge to $Value = \omega_0 = 100$. All methods used the same number of timesteps. Fully implicit timestepping is used.

Refine level	Value	Total itns/step		Outer itns/step	Ratio
		Fixed pt policy	Full policy	Full policy	
$\sigma = 0.2, \eta = 0.032297$					
1	100.6090	4.67	10.16	3.88	N/A
2	100.1775	4.57	9.32	3.92	N/A
3	100.0471	4.33	9.08	3.98	3.31
4	100.0108	4.21	8.64	4.02	3.59
5	99.9999	4.08	8.04	4.05	3.32
$\sigma = 0.3, \eta = 0.045452$					
1	100.3375	4.91	10.94	4.18	N/A
2	100.0842	4.84	10.19	4.32	N/A
3	100.0213	4.64	9.89	4.38	4.03
4	100.0049	4.65	9.47	4.45	3.83
5	100.0000	4.44	8.81	4.42	3.34

These two schemes show no difference in computed values to seven digits. However the fixed point policy scheme requires less than half the iterations required by the full policy iteration. The computational cost for these methods is dominated by the FFTs required to carry out the dense matrix-vector multiply, hence the CPU time is proportional to the number of iterations.

Table 7.4 also shows the number of outer iterations required by full policy iteration. The convergence ratio refers to the ratio of successive changes in the solution as the mesh and timesteps are reduced by two. This ratio indicates that better than linear convergence is obtained due to the maximal use of central differencing as much as possible for the V_W term [37, 22].

7.2. Regime switching. In this section, we will consider a numerical example for the regime switching, American option example described in section 5. We consider a case with three regimes $1, 2, 3$. The transition probability array $\boldsymbol{\lambda}$, jump amplitudes $\boldsymbol{\xi}$, and volatilities $\boldsymbol{\sigma}$ are given in (7.3). Other data are given in Table 7.5.

$$\boldsymbol{\lambda} = \begin{bmatrix} -3.5613 & 0.2405 & 3.3208 \\ 1.1279 & -1.2008 & 0.0729 \\ 2.9882 & 0.2025 & -3.1907 \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{bmatrix} 1.0 & 0.9095 & 1.0279 \\ 1.2502 & 1.0 & 1.6512 \\ 0.9693 & 0.7732 & 1.0 \end{bmatrix}, \quad \boldsymbol{\sigma} = \begin{bmatrix} 0.2 \\ 0.15 \\ 0.30 \end{bmatrix}. \tag{7.3}$$

Table 7.6 shows the grid and timestep data used for a convergence study for this problem. At each grid refinement, new fine grid nodes are added between each coarse grid node, and the timestep control parameter is halved. Crank–Nicolson variable timestepping is used.

Table 7.5

Data for the regime switching, American problem.

Expiry time	0.50	Strike K	100
Payoff	put	Risk free rate r	0.02
Exercise	American	Scaling parameter	$10^6/\Delta\tau$

Table 7.6

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.

Refinement	S nodes	Timesteps	Unknowns
0	51	37	153
1	101	74	303
2	201	145	603
3	401	287	1203
4	801	571	2403
5	1601	1139	4803
6	3201	2273	6603

Table 7.7

Number of fixed point policy iterations per timestep. All methods used the same total number of timesteps. Crank–Nicolson timestepping used. American option, fixed point policy iteration, value at $t = 0, S = 100$, regime 1. Ratio is the ratio of successive changes as the mesh/timesteps are refined.

Refinement	$\Omega = 10$	$\Omega = 1/(\Delta\tau)$	$\Omega = 10^6/(\Delta\tau)$	Value	Ratio
0	5.60	5.60	5.60	6.8261328	
1	4.84	4.84	4.84	6.8292905	
2	4.33	4.33	4.33	6.8300983	3.9
3	3.96	3.86	3.82	6.8303228	3.6
4	4.22	3.90	3.61	6.8303765	4.2
5	4.32	3.79	3.12	6.8303906	3.8
6	5.1	4.29	3.0	6.8303941	4.1

Recall that we introduced a scaling factor Ω in (5.12). A natural choice for a scaling factor is $\Omega = C/(\Delta\tau)$, where C is a dimensionless constant selected so as to satisfy (6.8). In the examples in this section, the coarse grid timestep is such that condition (6.8) is satisfied for $C \geq 1$.

Table 7.7 shows that the number of iterations per step for the direct control method (for fine grids) is sensitive to the choice of scaling factor. All methods gave the same computed values to eight digits. Table 7.7 also indicates that the method is approximately second order.

We compared fixed point policy iteration with some other approaches. Policy iteration (2.1) was used, and the sparse matrix $(\mathcal{A} - \mathcal{B})$ was solved using a direct method, based on minimum degree ordering for $((\mathcal{A}-\mathcal{B})+(\mathcal{A}-\mathcal{B})')$. The convergence tolerance for the policy iteration is given in (7.2). Table 7.8 shows several other possible methods. A GMRES iterative solver, using a level zero ILU preconditioner

Table 7.8

Comparison of full policy iteration (2.1) *using a direct solve, full policy iteration with an iterative solution (GMRES), full policy iteration with simple iteration* (2.11)*, and fixed point policy iteration* (2.2)*, refinement level* 5*. Regime switching, American option, penalty formulation. All methods used the same number of timesteps. Crank–Nicolson timestepping used.*

Linear solution method	Outer iterations per step	Inner iterations per step	CPU time (normalized)
	Full policy iteration (2.1)		
Direct (min degree)	2.39	NA	54.3
GMRES (ILU(0))[31]	2.39	4.64	15.0
Simple iteration (2.11)	2.39	4.89	1.3
	Fixed point policy iteration (2.2)		
Direct	3.12	NA	1.0

[31], was used to solve the $(\mathcal{A}-\mathcal{B})$ matrix, in conjunction with the full policy iteration (2.1). In addition, full policy iteration was also used with the $(\mathcal{A}-\mathcal{B})$ matrix solved using a simple iteration (2.11). A convergence tolerance based on a relative update condition ($< 10^{-8}$) for the inner iteration was used in both cases. We compared these methods with the fixed point policy iteration scheme (2.2). Table 7.8 shows that fixed point policy iteration requires the least CPU time.

8. Conclusion. We have developed a fixed point policy iteration scheme for solving discretized HJB equations. This method is particularly useful if the risky asset (in a financial application) follows a jump diffusion or regime switching process.

We have determined sufficient conditions which ensure that this iteration scheme converges. In the penalty formulation case, these conditions are typically satisfied if a monotone discretization method is used, which is normally required in order to ensure convergence to the viscosity solution.

In the case that the discrete equations are solved using the approach in [8], convergence of the fixed point policy iteration can only be guaranteed if the discretized optimization problem satisfies a scaling condition. It is always possible to select a scaling parameter which satisfies this condition. It is interesting to observe that, for the direct control approach [8], the convergence rate is sensitive to the scaling of the nonlinear equations. This does not appear to have been observed previously, and merits further study.

Our numerical tests show that the fixed point policy iteration method is more efficient than a variety of alternative strategies. We have used a very general approach to prove the convergence of the fixed point policy iteration. In the case where the admissible set of controls is infinite, we do not require that the discretized equations at each node be a continuous function of the control (this may arise if we use central differencing as much as possible for monotone schemes). We also do not require a special choice for the initial iterate. Hence, the fixed point policy iteration scheme can be applied to a wide variety of discretized HJB equations.

REFERENCES

[1] R. Almgren and N. Chriss, *Optimal execution of portfolio transactions*, J. Risk, 3 (2000/2001), pp. 5–39.

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

[3] L. Andersen and J. Andreasen, *Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing*, Rev. Derivatives Res., 4 (2000), pp. 231–262.

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

[5] G. Barles and P. Souganidis, *Convergence of approximation schemes for fully nonlinear equations*, Asympto. Anal., 4 (1991), pp. 271–283.

[6] D. Bauer, A. Kling, and J. Russ, *A universal pricing framework for guaranteed minimum benefits in variable annuities*, ASTIN Bull., 38 (2008), pp. 621–651.

[7] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1961.

[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] Z. Chen and P. A. Forsyth, *A numerical scheme for the impulse control formulation for pricing variable annuities with a guaranteed minimum withdrawal benefit (GMWB)*, Numer. Math., 109 (2008), pp. 535–569.

[10] 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, 36 (2012), pp. 201–219.

[11] Z. Chen, K. Vetzal, and P. Forsyth, *The effect of modelling parameters on the value of GMWB guarantees*, Insurance Math. Econom., 43 (2008), pp. 165–173.
[12] S. Clift, *Linear and Non-linear Monotone Methods for Valuing Financial Options under-two Factor, Jump Diffusion Models*, Ph.D. thesis, School of Computer Science, University of Waterloo, 2007.
[13] R. Cont and P. Tankov, *Financial Modelling with Jump Processes*, Chapman and Hall, Boca Raton, FL, 2004.
[14] M. Dai, Y. K. Kwok, and J. Zong, *Guaranteed minimum withdrawal benefit in variable annuities*, Math. Finance, 18 (2008), pp. 595–611.
[15] M. Dai and Y. Zhong, *Penalty methods for continuous time portfolio selection with transaction costs*, J. Comput. Finance, 13 (2010), pp. 1–31.
[16] Y. d'Halluin, P. 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. Forsyth, and K. Vetzal, *Robust numerical methods for contingent claims under jump diffusion processes*, IMA J. Numer. Anal., 25 (2005), pp. 87–112.
[18] P. A. Forsyth and G. Labahn, *Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance*, J. Comput. Finance, 11 (2008), pp. 1–44.
[19] N. Haldrup and M. Ø. Nielsen, *A regime switching long memory model for electricity prices*, J. Econometrics, 135 (2006), pp. 349–376.
[20] M. Hardy, *A regime switching model of long term stock returns*, North American Actuarial J., 5 (2001), pp. 41–53.
[21] R. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA, 1960.
[22] Y. Huang and P. Forsyth, *Analysis of a penalty method for pricing a guaranteed minimum withdrawal benefit (GMWB)*, IMA J. Numer. Anal., 32 (2012), pp. 320–351.
[23] J. Kennedy, *Hedging Contingent Claims in Markets with Jumps*, Ph.D. thesis, School of Computer Science, University of Waterloo, 2007.
[24] A. Khaliq and R. Liu, *New numerical scheme for pricing American options with regime switching*, Int. J. Theor. Appl. Finance, 12 (2009), pp. 319–340.
[25] H. Kushner, *Numerical Methods for Stochastic Control Problems in Continuous Time*, 2nd ed., Springer, New York, 2001.
[26] M. A. Milevsky and T. S. Salisbury, *Financial valuation of guaranteed minimum withdrawal benefits*, Insurance Math. Econom., 38 (2006), pp. 21–38.
[27] B. Oksendal and A. Sulem, *Applied stochastic control of jump diffusions*, Springer, New York, 2007.
[28] H. Pham, *On some recent aspects of stochastic control and their applications*, Probab. Surveys, 2 (2005), pp. 506–549.
[29] H. Pham, *Continuous-time Stochastic Control and Optimization with Financial Applications*, Springer, Berlin, 2009.
[30] M. Puterman and M. Shin, *Modified policy iteration for discounted Markov decision problems*, Management Sci., 24 (1978), pp. 1127–1137.
[31] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[32] M. S. Santos and J. Rust, *Convergence properties of policy iteration*, SIAM J. Control Optim., 42 (2004), pp. 2094–2115.
[33] M. Thompson, M. Davison, and H. Rasmussen, *Natural gas storage valuation and optimization: A real options application*, Naval Res. Logist., 56 (2009), pp. 226–238.
[34] A. Tourin and T. Zariphopoulou, *Viscosity solutions and numerical schemes for investment/consumption models with transaction costs*, in Numerical Methods in Finance, L. C. G. Rogers and D. Talay, eds., Cambridge University Press, Cambridge, UK, 1997, pp. 245–269.
[35] R. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1961.
[36] M. Wahib, Z. Lin, and C. Edirisingh, *Pricing swing options in the electricity market under regime switching uncertainty*, Quant. Finance, 10 (2010), pp. 975–994.
[37] J. Wang and P. A. Forsyth, *Maximal use of central differencing for Hamilton–Jacobi–Bellman PDEs in finance*, SIAM J. Numer. Anal., 46 (2008), pp. 1580–1601.