

## CONVERGENCE PROPERTIES OF POLICY ITERATION\*

MANUEL S. SANTOS<sup>†</sup> AND JOHN RUST<sup>‡</sup>

**Abstract.** This paper analyzes asymptotic convergence properties of policy iteration in a class of stationary, infinite-horizon Markovian decision problems that arise in optimal growth theory. These problems have continuous state and control variables and must therefore be discretized in order to compute an approximate solution. The discretization may render inapplicable known convergence results for policy iteration such as those of Puterman and Brumelle [*Math. Oper. Res.*, 4 (1979), pp. 60–69]. Under certain regularity conditions, we prove that for piecewise linear interpolation, policy iteration converges quadratically. Also, under more general conditions we establish that convergence is superlinear. We show how the constants involved in these convergence orders depend on the grid size of the discretization. These theoretical results are illustrated with numerical experiments that compare the performance of policy iteration and the method of successive approximations.

**Key words.** policy iteration, method of successive approximations, quadratic and superlinear convergence, complexity, computational cost

**AMS subject classifications.** 49M15, 65K05, 90C30, 93B40

**DOI.** 10.1137/S0363012902399824

**1. Introduction.** The goal of this paper is to provide new insights into the convergence properties of *policy iteration*, an algorithm developed by Bellman (1955, 1957) and Howard (1960) for solving stationary, infinite-horizon Markovian dynamic programming (MDP) problems. Policy iteration has some rough similarities to the simplex algorithm of linear programming (LP). Just as the simplex algorithm generates a sequence of improving trial solutions to the LP problem (along with their associated costs), policy iteration generates an improving sequence of decision rules to the MDP problem (along with their associated value functions). Also, similarly to the simplex algorithm, policy iteration has been found to converge to the optimal solution in a remarkably small number of iterations. Typically fewer than 10 to 20 policy iterations are required to find the optimal solution. But analogously to LP, where the number of possible vertices increases exponentially fast as the number of variables  $M$  and constraints  $N$  increases, the number of possible decision rules of an MDP problem with  $M$  states and  $N$  actions in each state is  $N^M$ , which also grows exponentially fast in  $M$ . Klee and Minty (1972) have constructed worst-case families of LP problems where the simplex algorithm visits a large and exponentially increasing number of vertices before converging to the optimal solution. Are there also “worst-case” families of MDP problems where policy iteration visits an exponentially increasing number of trial decision rules before converging to the optimal solution?

Although we do not answer this question in the present paper, we consider an example due to Tsitsiklis (2000) of a family of finite state MDP problems with  $M$  states and two actions in each state where roughly  $M$  policy iteration steps are required to find the optimal solution. While this example may not represent the “worst case,” it is a fairly pessimistic result that suggests that policy iteration will not be an

---

\*Received by the editors March 12, 2002; accepted for publication (in revised form) August 9, 2003; published electronically February 18, 2004.

<http://www.siam.org/journals/sicon/42-6/39982.html>

<sup>†</sup>Department of Economics, Arizona State University, Tempe, AZ 85287 (Manuel.Santos@asu.edu).

<sup>‡</sup>Department of Economics, University of Maryland, College Park, MD 20742 (jrust@gemini.econ.umd.edu).

efficient method for solving all types of MDP problems. The reason is that each policy iteration step is expensive: it involves (among other computations) a solution of a linear system with  $M$  equations in  $M$  unknowns at a total cost of  $O(M^3)$  arithmetic operations using standard linear equation solvers.

This paper was motivated by a desire to characterize those families of MDP problems for which only a small number of policy iteration steps are required to find the solution. Puterman and Brumelle (1979) were among the first to analyze the convergence properties of policy iteration for MDP problems with continuous state and action spaces. They showed that policy iteration is mathematically equivalent to *Newton's method*, and so under certain regularity conditions the policy iteration algorithm displays a quadratic order of convergence. The shortcoming of Puterman and Brumelle's abstract infinite-dimensional space analysis is that the generalized version of policy iteration that they describe is not actually computationally feasible in problems where there are an infinite number of states and actions. Their analysis assumed that the exact value functions are computed at each policy evaluation step, and this requires solving an infinite-dimensional system of linear equations. Furthermore, they impose a Lipschitz order condition which is not easily verifiable in their framework.

We analyze a computationally feasible version of policy iteration for a class of optimal growth problems with a continuum of states and decisions. We establish sufficient conditions under which the policy iteration solution for discretized versions of this problem exhibits a quadratic rate of convergence and a global operative estimate with a convergence rate equal to 1.5. We study how the constants involved in the convergence orders depend on the grid size of the discretization. Also, under more general conditions we show that convergence is superlinear. Thus, desirable convergence properties that Puterman and Brumelle demonstrated for an abstract, idealized version of policy iteration also hold for a practical, computationally feasible version that can be applied to solve actual problems in economics and finance. To the best of our knowledge, this is the first numerical algorithm with such convergence properties.

Despite the scant theoretical work and evaluations of the performance of policy iteration in economic models, computational economists (e.g., see Chapter 12 of Judd (1998)) have been well aware of this numerical procedure. Rust (1987) applies policy iteration to solve a durable good replacement problem. A recent paper by Benitez-Silva et al. (2001) offers an extensive numerical evaluation of policy iteration in general economic models. Unfortunately, there is little guidance about the conditions under which one should use policy iteration as opposed to a variety of alternative algorithms including the *method of successive approximations*. This latter method can be shown to be globally convergent since the value function to an MDP problem is the fixed point to a contraction mapping. Section 6 contains a computational illustration that compares successive approximations and our computationally feasible version of policy iteration for an optimal growth problem whose solution can also be computed analytically.

The paper is structured as follows. In section 2 we present some background on the policy iteration algorithm, reviewing known results in the finite and infinite state cases, respectively. In section 3, we focus our analysis on a class of multidimensional optimal growth problems. Then, in section 4 we describe our computationally feasible version of the policy iteration algorithm using bilinear interpolation. In section 5 we state and prove the main results on the convergence properties of our algorithm. Section 6 provides a comparison of the performance of both policy iteration and the method of successive approximations in the case of an optimal growth problem that

admits an analytic solution. This enables us to evaluate the uniform approximation errors of the two algorithms as well as compare their relative computational speed. The computational results are consistent with theoretical predictions. We conclude in section 7 with a discussion of our main findings.

**2. Background.** As noted in the introduction, policy iteration is a commonly used algorithm for solving stationary, infinite-horizon MDP problems. The MDP problem is mathematically equivalent to computing the fixed point  $V^*$  to *Bellman's equation*

$$(2.1) \quad V^* = \Gamma(V^*),$$

where  $\Gamma$  is the *Bellman operator* given by

$$(2.2) \quad \Gamma(V)(s) \equiv \max_{a \in A(s)} \left[ u(s, a) + \beta \int V(s') p(ds'|s, a) \right],$$

and  $\beta \in (0, 1)$  is the discount factor,  $u(s, a)$  represents the utility or payoff earned in state  $s$  when action  $a$  is taken, and  $p(ds'|s, a)$  is a conditional probability distribution for next period's state  $s'$  given the current state  $s$  and action  $a$ . As is well known (see, e.g., Blackwell (1965)), under mild regularity conditions the Bellman operator is a contraction mapping, and hence its fixed point  $V^*$ , the *value function*, is unique. The solution to Bellman's equation is of considerable interest because it has been shown that from  $V^*$  we can compute the corresponding *optimal decision rule or policy function*  $\alpha^*$  given by

$$(2.3) \quad \alpha^*(s) \equiv \arg \max_{a \in A(s)} \left[ u(s, a) + \beta \int V^*(s') p(ds'|s, a) \right].$$

The policy iteration algorithm computes  $V^*$  via a sequence of trial value functions  $\{V_n\}$  and decision rules  $\{\alpha_n\}$  under an alternating sequence of *policy improvement* and *policy evaluation* steps. The policy improvement step computes an improved policy  $\alpha_n$  implied by the previous value function  $V_n$ :

$$(2.4) \quad \alpha_n(s) = \arg \max_{a \in A(s)} \left[ u(s, a) + \beta \int V_n(s') p(ds'|s, a) \right] \quad (\text{policy improvement}).$$

The policy evaluation step computes a new value function  $V_{n+1}$  implied by policy  $\alpha_n$ :

$$(2.5) \quad V_{n+1}(s) = \left[ u(s, \alpha_n(s)) + \beta \int V_{n+1}(s') p(ds'|s, \alpha_n(s)) \right] \quad (\text{policy evaluation}).$$

Policy iteration continues until  $V_{n+1} = V_n$ ,  $\alpha_{n+1} = \alpha_n$ , or the difference between two successive value functions or decision rules is less than a prescribed solution tolerance. Under fairly general conditions, policy iteration can be shown to generate a monotonically improving sequence of trial value functions,  $V_{n+1} \geq V_n$ . In the case of MDP problems where both the state space  $S$  and action sets  $A(s)$  are finite, there is only a finite number of possible policies bounded by  $N^M$ , where  $M$  is the number of states and  $N$  is the maximum number of possible actions in the constraint sets  $A(s)$ . This, together with the fact that policy iteration is monotonic (and thus cannot cycle) implies that the policy iteration algorithm will converge to the true fixed point in a finite number of steps (assuming all arithmetic operations are carried

out exactly, without any rounding error). Although the number of potential policies is vast (for example a stopping problem with two choices and 1000 states has  $2^{1000}$  or about  $1.07 \times 10^{301}$  possible policies), it has been observed in most computational examples that policy iteration converges in a remarkably small number of iterations, typically less than 20. Furthermore, the number of policy iteration steps appears to be independent of the number of states and decisions.

The most commonly used alternative to policy iteration is the method of successive approximations

$$(2.6) \quad V_{n+1} = \Gamma(V_n).$$

As is well known, this algorithm is globally convergent (an implication of the fact that  $\Gamma$  is a contraction mapping), but it converges at a geometric rate. The error after  $\widehat{T}$  successive approximation steps is  $O(\beta^{\widehat{T}}/(1-\beta))$ . Since each successive approximation step requires  $O(M^2N)$  arithmetic operations (and thus each iteration is an order faster than policy iteration), as the discount factor  $\beta \rightarrow 1$ , the number of successive approximation steps required to obtain acceptable accuracy rises rapidly, calling for a huge number of computations. Hence, when  $\beta$  is close to 1 (as in the calibration of models with quarterly data or smaller time intervals), the total computational burden of doing a small number of more expensive policy iteration steps may be less than the total amount of work involved in doing a large number of less expensive successive approximation steps.

Of course, this advice will not hold if the number of policy iteration steps increases sufficiently rapidly with  $M$ . Unfortunately, the following counterexample (Tsitsiklis (2000)) shows that in general the number of policy iteration steps cannot be bounded by a constant that is independent of  $M$ . Consider an MDP problem with  $M + 1$  states  $s \in \{0, 1, \dots, M\}$ . In each state there are two possible actions,  $a \in \{-1, +1\}$ , with the interpretation that  $a = -1$  corresponds to “moving one state to the left” and  $a = +1$  corresponds to “moving one state to the right.” This can be mapped into a transition probability given by  $p(ds'|s, a)$  that puts probability mass 1 on state  $s' = s - 1$  if  $a = -1$  and probability mass 1 on state  $s' = s + 1$  if  $a = +1$ . States  $s = 0$  and  $s = M$  are zero-cost-absorbing states regardless of the action taken, so that we have

$$(2.7) \quad u(s, a) = 0 \quad \text{if } s = 0 \text{ or } s = M.$$

For states  $s = 1, 2, \dots, M - 2$ , the payoff equals  $-1$  for moving left and  $-2$  for moving right:

$$(2.8) \quad u(s, a) = \begin{cases} -1 & \text{if } a = -1, \\ -2 & \text{if } a = +1. \end{cases}$$

Finally, for state  $s = M - 1$  there is a reward of  $2M$  for moving right and a reward of  $-1$  for moving left:

$$(2.9) \quad u(s, a) = \begin{cases} -1 & \text{if } a = -1, \\ 2M & \text{if } a = +1. \end{cases}$$

Consider policy iteration starting from the initial value function  $V_0(s) = 0$  for all  $s$ . Then it is easy to see that the optimal policy  $\alpha_0$  implied by this value function is

$\alpha_0(s) = -1$  for states  $s = 1, \dots, M-2$  and  $\alpha_0(M-1) = +1$  (observe that the choice of policy is irrelevant at the absorbing states  $s = 0$  and  $s = M$ ). It is not difficult to check that when  $\beta = 1$ , the value  $V_1$  associated with this policy  $\alpha_0$  is given by

$$(2.10) \quad V_1(s) = \begin{cases} 0 & \text{if } s = 0, M, \\ -s & \text{if } s \in \{1, 2, \dots, M-2\}, \\ 2M & \text{if } s = M-1. \end{cases}$$

This is the value function generated by the first policy iteration step. For the second step, notice that optimal policy  $\alpha_1$  implied by  $V_1$  is to go left at states  $s = 1, 2, \dots, M-3$ , but to go right at states  $s = M-2$  and  $s = M-1$ . This new policy  $\alpha_1$  differs from the initial policy  $\alpha_0$  in only one state,  $s = M-2$ , where it is now optimal to go right because the updated value  $V_1$  assigns a higher value  $V_1(M-1) = 2M$  to the penultimate state  $s = M-1$ . Moreover, after each succeeding policy iteration  $n = 2, \dots, M-1$ , the updated decision rule  $\alpha_n$  differs from the previous policy  $\alpha_{n-1}$  in state  $s = M-n-1$ , flipping the optimal action from left to right. This process continues for  $M-1$  policy iteration steps until the optimal policy  $\alpha^*(s) = +1$  for all  $s$ ; i.e., it is optimal to always move to the right. The optimal value function  $V^*(s)$  for this policy is then given by

$$(2.11) \quad V^*(s) = \begin{cases} 0 & \text{if } s = 0, M, \\ 2(s+1) & \text{if } s \in \{1, 2, \dots, M-1\}. \end{cases}$$

It should be stressed that the same type of result occurs for discounted problems with  $\beta \in (0, 1)$  as long as  $\beta$  is sufficiently close to 1.

In summary, this counterexample provides a family of MDP problems with  $M+1$  states in which a total of  $M-1$  policy iteration steps are required to converge to the optimal solution starting from an initial guess  $V_0 = 0$ . Therefore, it will not be possible to provide a bound on the number of steps that is independent of the number of states  $M$ . But an alternative analysis of the convergence of policy iteration, initiated by Puterman and Brumelle (1979), does suggest that under additional conditions only a small number of policy iterations should be required to solve the MDP problem and this number should be essentially independent of the number of states  $M$ . Puterman and Brumelle were among the first to notice that policy iteration is mathematically equivalent to Newton's method for solving nonlinear functional equations. Then, building on some generalized results on the quadratic convergence of the Newton iteration with supports replacing derivatives, Puterman and Brumelle showed that for some constant  $L$  the iteration (2.4)–(2.5) satisfies the quadratic convergence bound

$$(2.12) \quad \|V_{n+1} - V^*\| \leq \frac{\beta L}{(1-\beta)} \|V_n - V^*\|^2,$$

where  $\|V\|$  is a norm in the space of functions  $V$ . There are, however, two limitations to this result. For the case of finite state, finite action MDP problems, the quadratic convergence bound is not effective. First, (2.12) may hold only in a neighborhood of the fixed-point solution  $V^*$ . Consequently, the errors  $\|V_n - V^*\|$  tend to be either very large until  $V_n$  is in a “domain of attraction” of  $V^*$ , in which case the error immediately falls to 0 after one further policy iteration step. In more technical terms, following the analysis below one can show that for the case of finite state, finite action MDP problems, constant  $L$  is equal to zero if there is no policy switch after a small

perturbation in the value function, but  $L$  becomes undefined (i.e.,  $L = \infty$ ) at points with multiple maxima in which a small perturbation in the value function leads to a sudden switch in the optimal action. Therefore, for MDP problems with a finite number of states and actions the Puterman and Brumelle results cannot be applied. The second limitation was noted in the introduction; namely, the abstract infinite-dimensional version of policy iteration in (2.4)–(2.5) is not computationally feasible for problems where there are infinite numbers of states. In these problems some sort of discretization procedure must be employed. And it seems crucial to understand how constant  $L$  will depend on the discretization procedure.

**3. A reduced form model of economic growth.** For expositional convenience and for our later computational purposes, we now consider a reduced-form version of our MDP problem (2.1)–(2.2) that encompasses most standard models of economic growth (cf. Stokey, Lucas with Prescott (1989)). As in most of the economics literature, we distinguish between endogenous and exogenous growth variables. This is commonly assumed in macroeconomic applications, where Bellman’s equation is usually expressed in the following form:

$$(3.1) \quad V^*(k_0, z_0) = \max_{k_1} v(k_0, k_1, z_0) + \beta \int_Z V^*(k_1, z')Q(dz', z_0)$$

subject to (s.t.)  $(k_0, k_1, z_0) \in \Omega$ ,  
 $(k_0, z_0)$  fixed,  $0 < \beta < 1$ .

Here,  $k$  is a vector of endogenous state variables in a set  $K$ , which may include several types of capital stocks and measures of wealth. Also,  $z$  is a vector made up of stochastic exogenous variables such as some indices of productivity or market prices. This latter random vector lies in a set  $Z$ , and is governed by a probability law  $Q$  which is assumed to be weakly continuous. As is typical in economic growth theory (cf. Stokey, Lucas with Prescott (1989)), this formulation is written in reduced-form in the sense that the action or control variables are not explicitly laid out. It should be understood that for every  $(k_0, k_1, z_0)$  an optimal control has been selected from which the one-time payoff  $v(k_0, k_1, z_0)$  can be defined.

The technological constraints of this economy are represented by a given feasible set  $\Omega \subset K \times K \times Z$ , which is the graph of a continuous correspondence  $\Gamma : K \times Z \rightarrow K$ . The intertemporal objective is characterized by a return function  $v$  and a given discount factor  $0 < \beta < 1$ . Then,  $s = (k, z)$  is the vector of state variables lying in the set  $S = K \times Z$ . Let  $(S, \mathbb{S})$  denote a measurable space.

As discussed in Stokey, Lucas with Prescott (1989) (see p. 240), the optimization problem of the preceding section is clearly more general. That is, with an appropriate choice of the action space and laws of motion for the state variables, our MDP problem (3.1) can be embedded in the framework of functional equations (2.1)–(2.2). The converse is not true. Our main interest, however, is to apply policy iteration to a standard macroeconomic setting. This optimization problem will become handy for our numerical experiments in section 6 of a neoclassical growth model with leisure. Moreover, it becomes transparent from our arguments below that our results can be extended to related MDP problems since the analysis centers on the following specific assumptions.

*Assumption 1.* The set  $S = K \times Z \subset \mathbb{R}^\ell \times \mathbb{R}^m$  is compact, and  $\mathbb{S}$  is the Borel  $\sigma$ -field. For each fixed  $z$  the set  $\Omega_z = \{(k, k') \mid (k, k', z) \in \Omega\}$  is convex.

*Assumption 2.* The mapping  $v : \Omega \rightarrow \mathbb{R}$  is continuous. Also, there exists  $\eta > 0$  such that for every  $z$  function  $v(k, k', z) + \frac{\eta}{2}\|k'\|^2$  is concave in  $(k, k')$ .

These assumptions are fairly standard. In Assumption 2,  $\|\cdot\|$  denotes the Euclidean norm. The asserted uniform concavity of  $v$  will be needed only for some stronger versions of our results. Under these conditions the value function  $V^*(k_0, z_0)$ , given in (3.1), is well defined and jointly continuous in  $K \times Z$ . Moreover, for each fixed  $z_0$  the mapping  $V^*(\cdot, z_0)$  is concave. The optimal value  $V^*(k_0, z_0)$  is attained at a unique point  $k_1$  given by the policy function  $k_1 = g(k_0, z_0)$  characterizing the set of optimal solutions.

**4. Policy iteration in a space of piecewise linear interpolations.** In what follows we shall be concerned with a policy iteration algorithm in a space of piecewise bilinear interpolations. Most of our results below apply to other interpolation schemes provided that (a) the operator is monotone, and (b) the operator has a fixed point. These two conditions may be quite restrictive for some approximation schemes such as polynomial and spline interpolations, but they hold true for several forms of piecewise linear interpolation.

We would like to remark that piecewise linear interpolation has certain computational advantages over the usual discretization procedure in which the domain and functional values are restricted to a fixed grid of prespecified points. Indeed, discretizations based on functional evaluations over a discrete set of points are not computationally efficient for smooth problems. Most calculations may become awkward under a discrete state space, and hence the corresponding algorithms are usually rather slow. For instance, the most powerful maximization procedures make use of the information provided by the values of the functional derivatives. These powerful techniques may still be applied under piecewise linear interpolation (cf. Santos (1999)). Therefore, piecewise linear interpolation preserves the aforementioned properties of monotonicity and existence of a fixed point for the operator, and at the same time allows for the use of efficient searching procedures over a continuous state space.

**4.1. Formulation of the numerical algorithm.** Let us assume that both  $K$  and  $Z$  are convex polyhedra. This does not entail much loss of generality in most economic applications. Let  $\{S^j\}$  be a finite family of *simplices*<sup>1</sup> in  $K$  such that  $\cup_j S^j = K$  and  $\text{int}(S^i) \cap \text{int}(S^j) = \emptyset$  for every pair  $S^i, S^j$ . Also, let  $\{D^i\}$  be a finite family of simplices in  $Z$  such that  $\cup_i D^i = Z$  and  $\text{int}(D^i) \cap \text{int}(D^j) = \emptyset$  for every pair  $D^i, D^j$ . Define the *grid size* or *mesh level* as

$$h = \max_{j,i} \text{diam} \{S^j, D^i\}.$$

Let  $(k^j, z^i)$  be a generic vertex of the triangulation. Then, every  $k$  and  $z$  can be represented as a convex combination of  $\{k^j\}$  and  $\{z^i\}$ . More precisely, for  $k \in S^j$  and  $z \in D^i$  there is a unique set of nonnegative weights  $\lambda_j(k)$  and  $\varphi_i(z)$ , with  $\sum_j \lambda_j(k) = 1$  and  $\sum_i \varphi_i(z) = 1$ , such that

$$(4.1) \quad k = \sum_j \lambda_j(k) k^j \quad \text{and} \quad z = \sum_i \varphi_i(z) z^i \quad \text{for all } k^j \in S^j \text{ and } z^i \in D^i.$$

We next define a finite-dimensional space of numerical functions compatible with the simplex structure  $\{S^j, D^i\}$ . Each element  $V^h$  is determined by its nodal values

<sup>1</sup>A simplex  $S^j$  in  $\mathbb{R}^\ell$  is the set of all convex combinations of  $\ell + 1$  given points (cf. Rockafellar (1970)). Thus, a simplex in  $\mathbb{R}^1$  is an interval, a simplex in  $\mathbb{R}^2$  would be a triangle, and a simplex in  $\mathbb{R}^3$  would be a tetrahedron.

$\{V^h(k^j, z^i)\}$  and is extended over the whole domain by the bilinear interpolation

$$V^h(k, z) = \sum_j \lambda_j(k) \left[ \sum_i \varphi_i(z) V^h(k^j, z^i) \right].$$

Note that the interpolation is first effected in space  $Z$  and then in space  $K$ . This interpolation ordering is appropriate for carrying out the numerical integrations and maximizations outlined below. Let the function space

$$(4.2) \quad \mathcal{V}^h = \left\{ V^h : K \times Z \longrightarrow \mathbb{R} \left| \begin{array}{l} V^h(k, z) = \sum_j \lambda_j(k) \left[ \sum_i \varphi_i(z) V^h(k^j, z^i) \right] \\ \text{for } \lambda_j \text{ and } \varphi_i \text{ satisfying (4.1)} \end{array} \right. \right\}.$$

It follows that  $\mathcal{V}^h$  is a Banach space when equipped with the norm

$$(4.3) \quad \|V^h\| = \max_{(k^j, z^i) \in K \times Z} |V^h(k^j, z^i)| \quad \text{for } V^h \in \mathcal{V}^h.$$

We now consider the following algorithm for policy iteration in space  $\mathcal{V}^h$ .

- (i) *Initial step:* Select an accuracy level  $\varepsilon$  and an initial guess  $V_0^h$ .
- (ii) *Policy improvement step:* Find  $k_1 = g_n^h(k_0^j, z_0^i)$  that solves

$$(4.4) \quad B^h(V_n^h)(k_0^j, z_0^i) \equiv -V_n^h(k_0^j, z_0^i) + \max_{k_1} v(k_0^j, k_1, z_0^i) + \beta \int_Z V_n^h(k_1, z') Q(dz', z_0^i)$$

for each vertex point  $(k_0^j, z_0^i)$ .

- (iii) *Policy evaluation step:* Find  $V_{n+1}^h(k_0^j, z_0^i)$  that solves

$$(4.5) \quad V_{n+1}^h(k_0^j, z_0^i) = v(k_0^j, g_n^h(k_0^j, z_0^i), z_0^i) + \beta \int_Z V_{n+1}^h(g_n^h(k_0^j, z_0^i), z') Q(dz', z_0^i)$$

for each vertex point  $(k_0^j, z_0^i)$ .

- (iv) *End of iteration:* If  $\|V_{n+1}^h - V_n^h\| \leq \varepsilon$ , stop; else, increment  $n$  by 1 and return to step (ii).

Observe that in step (ii) for a given  $V_n^h$  we first carry out the integration operation and then find the optimal policy  $g_n^h$ . And in step (iii) for a given  $g_n^h$  we search for a fixed-point solution  $V_{n+1}^h$ . Technically, a natural approach for solving (4.5) is to use numerical integration and then limit the search for the fixed point to a finite system of linear equations (cf. Dahlquist and Bjorck (1974) (pp. 396-397)). Regarding step (iv), the iteration process will stop once the term  $\|V_{n+1}^h - V_n^h\|$  falls within a prespecified accuracy level,  $\varepsilon$ .

The solution to (4.5) can be written as

$$(4.6) \quad v_{g_n^h} = [I - \beta P_{g_n^h}] V_{n+1}^h,$$

where  $v_{g_n^h}$  represents the utility implied by policy function  $g_n^h$ , that is,  $v_{g_n^h}(k_0^j, z_0^i) = v(k_0^j, g_n^h(k_0^j, z_0^i), z_0^i)$  for each vertex point  $(k_0^j, z_0^i)$ , and  $P_{g_n^h}$  is the Markov (conditional expectation) operator defined by

$$(4.7) \quad \begin{aligned} P_{g_n^h}(V(k_0^j, z_0^i)) &= \int_Z V(g_n^h(k_0^j, z_0^i), z') Q(dz', z_0^i) \\ &= \sum_{j'} \lambda_{j'}(g_n^h(k_0^j, z_0^i)) \left[ \int_Z \sum_{i'} \varphi_{i'}(z') V(k_0^{j'}, z_0^{i'}) Q(dz', z_0^i) \right]. \end{aligned}$$

It should be understood that in this latter expression  $g_n^h(k_0^j, z_0^i)$  belongs to some simplex  $S^{j'}$ , and  $z'$  is integrated out over simplices of the form  $D^{i'}$ .

Observe that  $P_{g_n^h}$  defines a linear operator in the space  $\mathcal{V}^h$  of functions  $V^h$  that can be represented by its nodal values  $\{V^h(k_0^j, z_0^i)\}$ . Since this operator is positive and bounded and  $\beta \in (0, 1)$ , it can be shown that the inverse operator  $[I - \beta P_{g_n^h}]^{-1}$  exists and has the following series representation:

$$(4.8) \quad [I - \beta P_{g_n^h}]^{-1} = \sum_{t=0}^{\infty} \beta^t P_{g_n^h}^t.$$

The distance between two operators  $P_g$  and  $P_{\hat{g}}$  will be defined by

$$(4.9) \quad \|P_g - P_{\hat{g}}\| = \max_{\{(k_0^j, z_0^i)\}} \left\{ \sum_{S^{j'}} \left[ \sum_{j'} |\lambda_{j'}(g(k_0^j, z_0^i)) - \lambda_{j'}(\hat{g}(k_0^j, z_0^i))| \right] \right\}.$$

The summation of these absolute differences goes over all weights  $\lambda_{j'}$  and over all simplices  $S^{j'}$ , under the convention that  $\lambda_{j'}(g(k_0^j, z_0^i)) = 0$  if  $g(k_0^j, z_0^i)$  does not belong to  $S^{j'}$ .

If  $\{V_n^h\}_{n \geq 1}$  is a sequence of functions generated by (4.4)–(4.5), one can readily check from these equations that such a sequence satisfies

$$(4.10) \quad V_{n+1}^h = V_n^h + [I - \beta P_{g_n^h}]^{-1} B^h(V_n^h).$$

Moreover, if in the policy improvement step the set of maximizers  $g_n^h$  is unique, then  $-[I - \beta P_{g_n^h}]$  is the derivative of  $B^h$  at  $V_n^h$  when  $P_{g_n^h}$  is considered as a linear operator in the finite-dimensional space  $\mathcal{V}^h$ . Hence, (4.10) implies that policy iteration is equivalent to Newton’s method applied to operator  $B^h$  defined in (4.4). As is well known (e.g., Traub and Woźniakowski (1979)), Newton’s method exhibits locally a quadratic rate of convergence provided that the functional derivative satisfies a certain regular Lipschitz condition. But the uniqueness of the set of maximizers seems fairly stringent for operator  $B^h$ . Indeed, function  $V_{n+1}^h$  is not necessarily concave, since it is obtained as the solution to (4.5). Therefore, there is no guarantee that the maximizer in (4.4) is unique, and consequently that  $B^h$  has a well-defined derivative. To circumvent these technicalities, Puterman and Brumelle (1979) apply an extension of Newton’s method to policy iteration following a familiar procedure with supports replacing derivatives. Even though operator  $B^h$  may not have a well-defined derivative, the following general property follows from the above maximization step: If  $g_n^h$  is a selection of the correspondence of maximizers in (4.4), then it must be the case that  $-[I - \beta P_{g_n^h}]$  is the support of operator  $B^h$  at  $V_n^h$ . More precisely, (4.4) implies that for any other function  $V^h$  in  $\mathcal{V}^h$  the following condition must hold:

$$(4.11) \quad B^h(V^h) - B^h(V_n^h) \geq -[I - \beta P_{g_n^h}][V^h - V_n^h].$$

Of course, one readily sees from (4.11) that if there is a unique set of maximizers  $g_n^h$ , then  $-[I - \beta P_{g_n^h}]$  is the derivative of  $B^h$  at  $V_n^h$ .

**4.2. Existence of a fixed point and monotonicity.** For our later analysis, we need to establish the existence of a unique fixed point for our algorithm and the monotone convergence to such a solution. We begin with the following discretized

version of Bellman’s equation.

$$(4.12) \quad \begin{aligned} V^h(k_0^j, z_0^i) &= \max_{k_1} v(k_0^j, k_1, z_0^i) + \beta \int_Z V^h(k_1, z') Q(dz', z_0^i) \\ &\text{s.t. } (k_0^j, k_1, z_0^i) \in \Omega \\ &\text{for each vertex point } (k_0^j, z_0^i). \end{aligned}$$

Note that this equation needs to be satisfied only at each vertex point  $(k_0^j, z_0^i)$ .

LEMMA 4.1. Equation (4.12) has a unique solution  $V^h$  in  $\mathcal{V}^h$ .

*Proof.* The proof is standard. One just defines the discretized dynamic programming operator  $V_{n+1}^h = T^h(V_n^h)$  given by

$$(4.13) \quad \begin{aligned} V_{n+1}^h(k_0^j, z_0^i) &= \max_{k_1} v(k_0^j, k_1, z_0^i) + \beta \int_Z V_n^h(k_1, z') Q(dz', z_0^i) \\ &\text{s.t. } (k_0^j, k_1, z_0^i) \in \Omega \\ &\text{for each vertex point } (k_0^j, z_0^i). \end{aligned}$$

One immediately sees that  $T^h$  is a contraction mapping in  $\mathcal{V}^h$  with modulus  $0 < \beta < 1$ . By a well-known fixed-point theorem,  $T^h$  has a unique fixed point  $V^h$  in  $\mathcal{V}^h$ .

Notice that  $V^h = T^h(V^h)$  implies that  $B^h(V^h) = 0$ . Therefore, the method of successive approximations as defined by (4.13) allows us to prove the existence of a fixed point for our algorithm as defined by (4.4)–(4.5). We next verify the monotone convergence of the algorithm to the fixed-point solution.  $\square$

LEMMA 4.2. Assume that  $\{V_n^h\}_{n \geq 0}$  is a sequence satisfying (4.4)–(4.5). Then  $V_{n+1}^h \geq V_n^h$  for all  $n \geq 1$ .

*Proof.* From (4.5), consider the following equation:

$$(4.14) \quad V_n^h = v_{g_{n-1}^h} + \beta P_{g_{n-1}^h} V_n^h.$$

That is,  $V_n^h$  is the value function under policy  $g_{n-1}^h$ . Now, call  $g_n^h$  the corresponding set of maximizers over the right-hand side of (4.4) under function  $V_n^h$ . Then,

$$(4.15) \quad v_{g_n^h} + \beta P_{g_n^h} V_n^h \geq V_n^h.$$

Moreover, a further application of this procedure for function  $V_n^h$  on the left-hand side of (4.15) yields

$$v_{g_n^h} + \beta P_{g_n^h} v_{g_n^h} + \beta^2 [P_{g_n^h}]^2 V_n^h \geq V_n^h.$$

Hence, after  $t$  iterations, we obtain

$$(4.16) \quad \sum_{s=0}^t \beta^s [P_{g_n^h}]^s v_{g_n^h} + \beta^{t+1} [P_{g_n^h}]^{t+1} V_n^h \geq V_n^h.$$

Now, letting  $t \rightarrow \infty$ , it follows from (4.6)–(4.8) that the left-hand side of (4.16) converges to  $V_{n+1}^h$ . Therefore,  $V_{n+1}^h \geq V_n^h$ .  $\square$

Remark 4.3. Assume that  $\{V_n^h\}_{n \geq 1}$  is a sequence satisfying (4.4)–(4.5). Let  $T^h$  be the discretized dynamic programming operator as defined by (4.13). Then, from the previous method of proof one can readily establish that  $V_{n+1}^h \geq T^h(V_n^h) \geq V_n^h$ . In this sense, the policy iteration algorithm converges faster to the fixed-point solution than the method of successive approximations generated by operator  $T^h$ .

**5. Convergence properties of the numerical algorithm.** We now establish some convergence properties of our policy iteration algorithm. We begin with a global result for concave interpolations in which the convergence order is equal to 1.5 and the constant involved in the convergence order is relatively easy to estimate. Among other applications, this result may be useful in placing an upper bound on the number of policy iteration steps over a well-defined convergence region to reach a tolerance level  $\varepsilon$ . Then, the same strategy of proof used for this global convergence result will be applied to address further local convergence properties. Thus for concave interpolations we prove quadratic convergence of the algorithm, and for a more general setting of continuous functions we prove that convergence is superlinear. The constants involved in these convergence orders are shown to depend on the mesh level of the discretization.

**5.1. Global convergence in a space of concave functions.** For present purposes, we shall assume that either the fixed point  $V^h(k, z)$  is a concave function in  $k$  or the sequence  $\{V_n^h(k, z)\}_{n \geq 1}$  generated by policy iteration is a sequence of concave functions in  $k$ .

In what follows, for a real-valued function the norm  $\|V_n^h\|$  is as defined in (4.3), and the distance between operators  $\|P_g - P_{\hat{g}}\|$  is as defined in (4.9). Also, for an  $\ell$ -dimensional function  $g = (g_1, \dots, g_r, \dots, g_\ell)$  let

$$(5.1) \quad \|g\| = \max_{0 \leq r \leq \ell} \left| \max_{(k_0^j, z_0^i)} g_r(k_0^j, z_0^i) \right|.$$

The following simple result will be very useful.

LEMMA 5.1. *Let  $h$  be the grid size of triangulation  $\{S^j, D^i\}$ . Then, there exists a constant  $\kappa$  that depends on the grid configuration and the dimension  $\ell$  such that  $\|P_g - P_{\hat{g}}\| \leq \frac{\kappa}{h} \|g - \hat{g}\|$ .*

As presently shown, constant  $\kappa$  depends on the uniformity of the grid; also,  $\frac{\kappa}{h}$  converges to  $\infty$  as  $h$  goes to 0. Therefore, the constants involved in our convergence results below will depend on the uniformity of the grid and on the mesh level,  $h$ , of the discretization. These constants get unbounded as  $h$  goes to zero.

*Proof of Lemma 5.1.* The proof becomes more transparent for the case  $\ell = 1$ . In this simple case, we can show that

$$(5.2) \quad \kappa = \frac{2h}{h_0},$$

where  $h_0 = \min_r \{k^{r+1} - k^r\}$  is the minimum distance over all pairs of adjacent grid points. Thus, for a uniform grid we obtain  $\kappa = 2$ .

To verify (5.2), consider two points  $k_1 = g(k_0^j, z_0^i)$  and  $\widehat{k}_1 = \widehat{g}(k_0^j, z_0^i)$ . If  $k_1$  and  $\widehat{k}_1$  are contained in the same grid interval  $[k^r, k^{r+1}]$ , then  $k = \lambda(k)k^r + (1 - \lambda(k))k^{r+1}$  for  $k = k_1, \widehat{k}_1$ . Hence,

$$(5.3) \quad \begin{aligned} & |\lambda(k_1) - \lambda(\widehat{k}_1)| + |(1 - \lambda(k_1)) - (1 - \lambda(\widehat{k}_1))| \\ &= 2|\lambda(k_1) - \lambda(\widehat{k}_1)| \leq \frac{2|k_1 - \widehat{k}_1|}{k^{r+1} - k^r} \leq \frac{2|k_1 - \widehat{k}_1|}{h_0}. \end{aligned}$$

If  $k_1 < \widehat{k}_1$  belong to different grid intervals, then  $k^r < k_1 < k^{r+1} < \dots < k^{r+n} < \widehat{k}_1 < k^{r+n+1}$  for some integers  $r$  and  $n$ . Notice that for  $n > 1$  we get from (4.9) that

$\|P_g - P_{\hat{g}}\| = 2$ . Thus, for this case the result is trivially satisfied for  $\kappa$  in (5.2). If  $n = 1$ , then (4.9) amounts to

$$(5.4) \quad |\lambda(k_1)| + |(1 - \lambda(k_1)) - \lambda(\widehat{k}_1)| + |1 - \lambda(\widehat{k}_1)|.$$

By continuity, for  $\widehat{k}_1 = k^{r+1}$  the bound in (5.3) is also valid for (5.4). From  $\widehat{k}_1 = k^{r+1}$  this bound can be extended for all  $\widehat{k}_1$  in the interval  $[k^{r+1}, k^{r+2}]$ , since  $\lambda(\widehat{k}_1)$  has Lipschitz constant bounded by  $1/h_0$ . Consequently,

$$|\lambda(k_1)| + |(1 - \lambda(k_1)) - \lambda(\widehat{k}_1)| + |1 - \lambda(\widehat{k}_1)| \leq \frac{2|k_1 - \widehat{k}_1|}{h_0}.$$

Therefore, in all cases the stated result holds true for  $(k_0^j, z_0^j)$ , and  $(k_0^j, z_0^j)$  is an arbitrarily chosen vertex point.

The proof in the multidimensional case is very similar. Let us assume that the domain  $K$  is subdivided into a family of simplices  $\{S^j\}$  such that  $\bigcup_j S^j = K$  and  $\text{int}(S^i) \cap \text{int}(S^j) = \emptyset$  for every pair  $S^i, S^j$ . Then, every point  $k$  in  $S^j$  has a unique representation as a convex combination of the vertex points,  $k = \sum_{j=1}^{\ell+1} \lambda_j k_0^j$ . Moreover, every  $\lambda_j$  is a Lipschitz function on  $K$ , and we can find a uniform Lipschitz constant that applies for all  $\lambda_j$ . Therefore, given any pair of points  $k_1 = g(k_0^j, z_0^j)$  and  $\widehat{k}_1 = \widehat{g}(k_0^j, z_0^j)$ , the proof proceeds in the same way as above.  $\square$

**PROPOSITION 5.2.** *Let  $\{V_n^h(k, z)\}_{n \geq 1}$  be a sequence of functions generated by (4.4)–(4.5), and assume that every function  $V_n^h$  is concave in  $k$ . Let  $\{g_n^h\}_{n \geq 1}$  be the corresponding sequence of policy functions. Then, under Assumptions 1 and 2 there exists a constant  $L$  such that for any pair of functions  $V_n^h, V_{n+1}^h$ , it must hold that  $\|P_{g_n^h} - P_{g_{n+1}^h}\| \leq L \|V_n^h - V_{n+1}^h\|^{1/2}$ .*

*Proof.* First, the contraction property of operator  $T^h$  implies that

$$\|v_{g_n^h} + \beta P_{g_n^h} V_n^h - v_{g_{n+1}^h} - \beta P_{g_{n+1}^h} V_{n+1}^h\| \leq \beta \|V_n^h - V_{n+1}^h\|.$$

Moreover, by (4.3) and (4.7), we have  $\|\beta P_{g_{n+1}^h} V_n^h - \beta P_{g_{n+1}^h} V_{n+1}^h\| \leq \beta \|V_n^h - V_{n+1}^h\|$ . Hence, an application of the triangle inequality yields

$$(5.5) \quad \|v_{g_n^h} + \beta P_{g_n^h} V_n^h - v_{g_{n+1}^h} - \beta P_{g_{n+1}^h} V_n^h\| \leq 2\beta \|V_n^h - V_{n+1}^h\|.$$

Now, as is well known (e.g., see Lemma 3.2 of Santos (2000)) by the concavity of  $V_n^h$  in  $k$ , the convexity of  $\Omega$ , and the postulated concavity of  $v$  in Assumption 2, we can assert from (5.5) that

$$(5.6) \quad \|g_n^h - g_{n+1}^h\| \leq 2 \left(\frac{\beta}{\eta}\right)^{1/2} \|V_n^h - V_{n+1}^h\|^{1/2}.$$

Therefore, a straightforward application of Lemma 5.1 proves the result for  $L = \frac{2\kappa}{h} \left(\frac{\beta}{\eta}\right)^{1/2}$ .  $\square$

It should be stressed that in the preceding proof only one value function  $V_n^h$  needs to be concave. Hence, the following result is an easy consequence of the previous arguments.

**COROLLARY 5.3.** *Let  $V^h$  be the fixed point of the discretized Bellman equation (4.12), and assume that  $V^h(k, z)$  is a concave function in  $k$ . Let  $\{V_n^h\}_{n \geq 1}$  be a sequence of (not necessarily concave) functions generated by (4.4)–(4.5). Then, under*

Assumptions 1 and 2, there exists a constant  $L$  such that

$$\|P_{g^h} - P_{g_n^h}\| \leq L \|V^h - V_n^h\|^{1/2} \quad \text{for all } n.$$

*Remark 5.4.* Observe that for  $\ell = 1$  the fixed-point solution  $V^h(k, z)$  is in fact a concave function in  $k$ . Indeed, for  $\ell = 1$  the concavity of  $V^h(k, z)$  can be established by the method of successive approximations.

*Remark 5.5.* Note that in Corollary 5.3 operator  $P_{g_n^h}$  is not necessarily unique, since there may not be a unique set of maximizers  $g_n^h$ . But the result holds true for any  $P_{g_n^h}$ , and the distance between two optimal policies must be arbitrarily small for a large enough  $n$ .

All the basic ingredients are now in place to demonstrate that the algorithm displays a convergence rate equal to 1.5 (cf. Puterman and Brumelle (1979)).

**THEOREM 5.6.** *Assume that the conditions of Proposition 5.2 are satisfied. Then*

$$\|V_{n+1}^h - V_n^h\| \leq \frac{\beta L}{1 - \beta} \|V_n^h - V_{n-1}^h\|^{1.5} \quad \text{for all } n.$$

*Proof.* First, for any two functions  $V_n^h$  and  $V_{n-1}^h$  we must have (see (4.11))

$$(5.7) \quad B^h(V_{n-1}^h) - B^h(V_n^h) \geq -[I - \beta P_{g_n^h}][V_{n-1}^h - V_n^h].$$

Moreover, a further application of (5.7) yields

$$\begin{aligned} B^h(V_n^h) - [I - \beta P_{g_n^h}][V_{n-1}^h - V_n^h] &\leq B^h(V_{n-1}^h) \\ &\leq B^h(V_n^h) - [I - \beta P_{g_{n-1}^h}][V_{n-1}^h - V_n^h]. \end{aligned}$$

Now, subtracting  $B^h(V_n^h) - [I - \beta P_{g_n^h}][V_{n-1}^h - V_n^h]$  from each of the three terms, we obtain

$$(5.8) \quad \begin{aligned} 0 &\leq B^h(V_{n-1}^h) - B^h(V_n^h) + [I - \beta P_{g_n^h}][V_{n-1}^h - V_n^h] \\ &\leq ([I - \beta P_{g_n^h}] - [I - \beta P_{g_{n-1}^h}])[V_{n-1}^h - V_n^h]. \end{aligned}$$

Then, for any  $V_{n+1}^h$  and  $V_n^h$  satisfying (4.4)–(4.5) we must have

$$(5.9) \quad \begin{aligned} \|V_{n+1}^h - V_n^h\| &= \|[I - \beta P_{g_n^h}]^{-1} B^h(V_n^h)\| \\ &\leq \|[I - \beta P_{g_n^h}]^{-1}\| \|B^h(V_n^h)\| \\ &= \|[I - \beta P_{g_n^h}]^{-1}\| \|B^h(V_n^h) - B^h(V_{n-1}^h) + [I - \beta P_{g_{n-1}^h}][V_n^h - V_{n-1}^h]\| \\ &\leq \|[I - \beta P_{g_n^h}]^{-1}\| \|[I - \beta P_{g_n^h}] - [I - \beta P_{g_{n-1}^h}]\| \|V_n^h - V_{n-1}^h\|. \end{aligned}$$

Here, both equalities come from (4.10). Also, the first inequality follows from the definition of the norm, and the last inequality is a consequence of (5.8).

Therefore, from (5.9) we obtain

$$(5.10) \quad \|V_{n+1}^h - V_n^h\| \leq \|[I - \beta P_{g_n^h}]^{-1}\| \|[I - \beta P_{g_n^h}] - [I - \beta P_{g_{n-1}^h}]\| \|V_n^h - V_{n-1}^h\|.$$

Finally, Proposition 5.2 together with (5.10) implies that

$$\|V_{n+1}^h - V_n^h\| \leq \frac{\beta L}{1 - \beta} \|V_n^h - V_{n-1}^h\|^{1.5}. \quad \square$$

THEOREM 5.7. *Assume that the conditions of Corollary 5.3 are satisfied. Then,*

$$\|V^h - V_{n+1}^h\| \leq \frac{\beta L}{1 - \beta} \|V^h - V_n^h\|^{1.5} \quad \text{for all } n.$$

*Proof.* From the monotonicity of policy iteration we have  $0 \leq V^h - V_{n+1}^h$ . Then,

$$\begin{aligned} 0 &\leq V^h - V_{n+1}^h = V^h - V_n^h - [I - \beta P_{g_n^h}]^{-1} B^h(V_n^h) \\ &\leq [I - \beta P_{g_n^h}]^{-1} [I - \beta P_{g_n^h}][V^h - V_n^h] - [I - \beta P_{g_n^h}]^{-1} [I - \beta P_{g^h}][V^h - V_n^h] \\ &= [I - \beta P_{g_n^h}]^{-1} ([I - \beta P_{g_n^h}] - [I - \beta P_{g^h}])[V^h - V_n^h]. \end{aligned}$$

Here, the first equality comes from (4.10); the inequality is a consequence of the maximization involved in operator  $B^h$  (cf. (4.11)) and the fact that  $B^h(V^h) = 0$ ; and after a simple factorization we get the last equality.

Now, taking norms and applying Proposition 5.2 it follows that

$$\begin{aligned} \|V^h - V_{n+1}^h\| &\leq \|[I - \beta P_{g_n^h}]^{-1}\| \|[I - \beta P_{g_n^h}] - [I - \beta P_{g^h}]\| \|V^h - V_n^h\| \\ &\leq \frac{\beta L}{1 - \beta} \|V^h - V_n^h\|^{1.5}. \quad \square \end{aligned}$$

**5.2. Local convergence properties.** Suitable variations of the preceding arguments will now allow us to establish further convergence properties near the fixed-point solution  $V^h$ .

**5.2.1. Quadratic convergence.** To guarantee the quadratic convergence of policy iteration we need the following strengthening of Corollary 5.3.

PROPOSITION 5.8. *Let  $V^h$  be the fixed point of the discretized Bellman equation (4.12). Assume that  $V^h(k, z)$  is a concave function in  $k$ . For each vertex point  $(k_0^j, z_0^i)$ , assume that  $g^h(k_0^j, z_0^i)$  is not a grid point in the family of simplices  $\{S^j\}$ . Let  $\{V_n^h\}_{n \geq 1}$  be a sequence of functions generated by (4.4)–(4.5). Then, under Assumptions 1 and 2, there are constants  $\widehat{L}$  and  $\widehat{N}$  such that*

$$\|P_{g^h} - P_{g_n^h}\| \leq \widehat{L} \|V^h - V_n^h\| \quad \text{for all } n \geq \widehat{N}.$$

After some obvious adjustments in the power estimates of the proof of Theorem 5.7, we now get the convergence result in Theorem 5.9.

THEOREM 5.9. *Assume that the conditions of Proposition 5.8 are satisfied. Then,*

$$\|V^h - V_{n+1}^h\| \leq \frac{\beta \widehat{L}}{1 - \beta} \|V^h - V_n^h\|^2 \quad \text{for all } n \geq \widehat{N}.$$

*Remark 5.10.* One may argue that Theorem 5.9 is a stronger result than Theorem 5.7. But Theorem 5.7 may be of interest for numerical applications<sup>2</sup> as constant  $L$  is easier to estimate. Also, note that constant  $L = 2 \frac{\kappa}{h} (\frac{\beta}{\eta})^{1/2}$  is  $O(\frac{1}{h})$ , whereas the proof of Proposition 5.8 below reflects that constant  $\widehat{L}$  is  $O(\frac{1}{h^2})$ . Finally, the arguments

<sup>2</sup>For instance, we can define the region of convergence  $R_\alpha = \{V \in \mathcal{V}^h \mid \frac{\beta L}{1 - \beta} \|V - V^h\|^{0.5} \leq 1 - \alpha\}$ , where  $V^h$  is the fixed-point solution and  $\alpha > 0$ . Then, for all  $V$  in  $R_\alpha$  we can find an upper bound on the number of policy iteration steps that are necessary to reach a certain tolerance level  $\varepsilon$ .

leading to the proof of Theorem 5.9 are heavily dependent on certain properties of piecewise linear approximations and the assumed interiority of the solution. In contrast, the arguments involved in the proof of Theorem 5.7 seem to be less specific.

*Proof of Proposition 5.8.* Let every function  $V^h$  in  $\mathcal{V}^h$  be represented by a finite-dimensional vector  $u^h$  that lists all nodal values  $\{V^h(k_0^j, z_0^i)\}$ . Then, we may rewrite optimization problem (4.12) as follows:

$$(5.11) \quad \begin{aligned} \max_{k_1} \psi(k_0^j, z_0^i, k_1, u^h) &= \max_{k_1} v(k_0^j, k_1, z_0^i) + \beta \int_Z V^h(k_1, z') Q(dz', z_0^i) \\ \text{s.t. } (k_0^j, k_1, z_0^i) &\in \Omega. \end{aligned}$$

In what follows, there is no restriction of generality to focus on a single vertex point  $(k_0^j, z_0^i)$ . The mapping  $\psi(k_0^j, z_0^i, \cdot, \cdot)$  has the following properties:

- (i) The mapping  $\psi(k_0^j, z_0^i, \cdot, \cdot)$  is continuous in  $(k_1, u)$ .
- (ii) For  $(k_0^j, z_0^i, u^h)$  function  $\psi(k_0^j, z_0^i, k_1, u^h) + \frac{\eta}{2} \|k_1\|^2$  is concave in  $k_1$ .
- (iii) Let  $D_3\psi(k_0^j, z_0^i, k_1, u; v)$  be the directional derivative of function  $\psi$  at  $(k_0^j, z_0^i, k_1, u)$  with respect to  $k_1$  in the direction  $v$ . Let  $B_\delta(u^h) = \{u \mid \|u - u^h\| < \delta\}$ . Then, for some small  $\delta > 0$ , and for all  $\|v\| = 1$  and all  $k_1$  sufficiently close to  $k_1^h = g^h(k_0^j, z_0^i)$ , there is a constant  $H > 0$  such that  $|D_3\psi(k_0^j, z_0^i, k_1, u; v) - D_3\psi(k_0^j, z_0^i, k_1, u^h; v)| \leq H\|u - u^h\|$  for every  $u$  in  $B_\delta(u^h)$ .

The continuity of  $\psi(k_0^j, z_0^i, \cdot, \cdot)$  follows from standard arguments. In (ii) the curvature parameter  $\eta$  of Assumption 2 still applies as the fixed-point solution  $V^h(k_1, z')$  is assumed to be concave in  $k_1$ . The Lipschitz property in (iii) comes from the fact that small changes in the nodal values lead to bounded variations in the slopes or directional derivatives of a piecewise linear function. In fact, for a given choice of norm  $\|u - u^h\|$  constant  $H$  will depend on the form of the triangulation  $\{S^j\}$ , especially on  $1/h_0$  (where  $h_0$  is the minimum distance between two grid points).

Now, the proof of this proposition will result from some simple extensions of standard arguments<sup>3</sup> (e.g., Fleming and Rishel (1975) (p. 170) and Montrucchio (1987) (p. 263)). For fixed  $(k_0^j, z_0^i)$  let  $k_1^h$  be the unique maximum point in (5.11) and let  $\hat{k}_1$  be a maximum solution under  $\psi(k_0^j, z_0^i, k_1, \hat{u})$  for  $\hat{u}$  in  $B_\delta(u^h)$ .

By Assumption 2 and the presupposed concavity of  $V^h(k_1, z')$  in  $k_1$ , we have

$$(5.12) \quad \psi(k_0^j, z_0^i, k_1^h, u^h) - \psi(k_0^j, z_0^i, \hat{k}_1, u^h) \geq -D_3\psi(k_0^j, z_0^i, k_1^h, u^h; \hat{k}_1 - k_1^h) + \frac{\eta}{2} \|\hat{k}_1 - k_1^h\|^2.$$

Since the objective reaches the maximum value at  $k_1^h$ , the directional derivative in (5.12) is nonpositive. Moreover, concave and piecewise linear functions in  $\mathbb{R}^\ell$  are absolutely continuous. Hence, we can apply the integral form of the mean-value theorem to the left-hand side of (5.12) so as to obtain

$$(5.13) \quad - \int_0^1 D_3\psi(\Phi_1(\lambda)) d\lambda \geq \frac{\eta}{2} \|\hat{k}_1 - k_1^h\|^2$$

for  $\Phi_1(\lambda) = (k_0^j, z_0^i, k_1^h + \lambda(\hat{k}_1 - k_1^h), u^h; \hat{k}_1 - k_1^h)$ .

<sup>3</sup>The added difficulty in the proof is that concavity in  $k_1$  is assumed only at point  $u = u^h$ ; i.e., see property (ii) above.

Also, by definition,  $\psi(k_0^j, z_0^i, \widehat{k}_1, \widehat{u}) - \psi(k_0^j, z_0^i, k_1^h, \widehat{u}) \geq 0$ . Hence,

$$(5.14) \quad \int_0^1 D_3\psi(\Phi_2(\lambda))d\lambda \geq 0$$

for  $\Phi_2(\lambda) = (k_0^j, z_0^i, k_1^h + \lambda(\widehat{k}_1 - k_1^h), \widehat{u}; \widehat{k}_1 - k_1^h)$ . Adding up inequalities (5.13)–(5.14), we get

$$(5.15) \quad - \int_0^1 [D_3\psi(\Phi_1(\lambda)) - D_3\psi(\Phi_2(\lambda))]d\lambda \geq \frac{\eta}{2} \|\widehat{k}_1 - k_1^h\|^2.$$

By (iii) above,

$$(5.16) \quad \int_0^1 |D_3\psi(\Phi_1(\lambda)) - D_3\psi(\Phi_2(\lambda))|d\lambda \leq H\|\widehat{u} - u^h\| \|\widehat{k}_1 - k_1^h\|.$$

Therefore, (5.15)–(5.16) implies

$$(5.17) \quad \|\widehat{k}_1 - k_1^h\| \leq \frac{2H}{\eta} \|\widehat{u} - u^h\|.$$

Finally, a straightforward application of Lemma 5.1 establishes Proposition 5.8 for  $\widehat{L} = \frac{\kappa}{h} \frac{2H}{\eta}$ .  $\square$

**5.2.2. Superlinear convergence.** In this part we lift the convexity and concavity conditions. Hence, we just assume that  $\Omega$  is the graph of a continuous correspondence  $\Gamma : K \times Z \rightarrow K$  defined on compact set  $K \times Z$  and the reward function  $v$  is a continuous mapping on  $\Omega$ .

Even though we are not able to guarantee quadratic convergence for the iteration scheme, we shall establish the intermediate property of superlinear convergence. This is a faster rate of convergence than that of the method of successive approximations, where by the contraction property of the dynamic programming operator one easily shows that

$$(5.18) \quad \|V^h - V_{n+1}^h\| \leq \beta \|V^h - V_n^h\|$$

for all  $n \geq 1$ , for every sequence  $\{V_n^h\}_{n \geq 1}$  generated by (4.13).

**PROPOSITION 5.11.** *Let  $\{V_n^h\}_{n \geq 1}$  be a sequence of functions generated by (4.4)–(4.5) that converge to the fixed point  $V^h$ . Then, for every  $\epsilon > 0$  there is  $\widehat{n}$  such that for each  $P_{g_n^h}$  with  $n \geq \widehat{n}$  there exists some  $P_{g^h}$  with the property that  $\|P_{g_n^h} - P_{g^h}\| \leq \epsilon$ .*

What this result states is that the correspondence of maximizers is upper-semicontinuous. Hence, for every  $P_{g_n^h}$  one can find an arbitrarily close  $P_{g^h}$ , provided that  $n$  is sufficiently large. With Proposition 5.11 at hand, from the proof of Theorem 5.7 one can easily obtain the following result.

**THEOREM 5.12.** *Assume that the conditions of Proposition 5.11 are satisfied. Then,*

$$(5.19) \quad \limsup_{n \rightarrow \infty} \frac{\|V^h - V_{n+1}^h\|}{\|V^h - V_n^h\|} = 0.$$

**6. Numerical experiments.** In this section we report some simple numerical experiments which are intended to evaluate the performance of policy iteration and the method of successive approximations. The analysis centers on a one-sector deterministic growth model with leisure, and the main purpose is to evaluate the computing cost and approximation errors of the value and policy functions under each computational method.

Formally, our one-sector growth model is described by the following optimization problem:

$$\begin{aligned}
 (6.1) \quad V(k_0) = & \max_{\{c_t, l_t, i_t\}} \sum_{t=0}^{\infty} \beta^t [\lambda \log c_t + (1 - \lambda) \log l_t] \\
 \text{s.t. } & c_t + i_t = Ak_t^\alpha (1 - l_t)^{1-\alpha}, \\
 & k_{t+1} = i_t + (1 - \delta)k_t, \\
 & k_t, c_t \geq 0, 0 \leq l_t \leq 1, k_0 \text{ given}, t = 0, 1, 2, \dots, \\
 & 0 < \beta < 1, 0 < \lambda < 1, A > 0, \\
 & 0 < \alpha < 1, 0 < \delta \leq 1.
 \end{aligned}$$

This is a familiar optimization problem in which  $c$  represents consumption,  $k$  represents the stock of physical capital, and  $l$  the fraction of time devoted to leisure activities. It is well known that for  $\delta = 1$  the value function  $V(k_0)$  takes the simple form  $V(k_0) = B + C \log k_0$ , where  $B$  is a certain constant and  $C = \frac{\lambda\alpha}{(1-\alpha\beta)}$ . Also, the policy function  $k_{t+1} = \alpha\beta Ak_t^\alpha (1 - l_t)^{1-\alpha}$  with  $l_t = \frac{(1-\lambda)(1-\alpha\beta)}{\lambda(1-\alpha) + (1-\lambda)(1-\alpha\beta)}$ . Under these simple functional forms, there is a unique steady state  $k^* = g(k^*)$ , which is globally stable.

The existence of one state variable,  $k$ , and two controls,  $l$  and  $c$ , suggests that all numerical maximizations may be efficiently carried out with a unique decision variable. Let us then write the model in a more suitable form for our computations. Note that at time  $t = 0$  the first-order conditions for our two control variables  $c$  and  $l$  are given by

$$\frac{\lambda}{c_0} = \mu \quad \text{and} \quad \frac{(1-\lambda)(1-l_0)^\alpha}{l_0 A k_0^\alpha (1-\alpha)} = \mu,$$

where  $\mu$  is a Lagrange multiplier. After some simple rearrangements, we obtain

$$c_0 = \frac{\lambda l_0 A k_0^\alpha (1-\alpha)}{(1-\lambda)(1-l_0)^\alpha}.$$

Hence,

$$k_1 = A k_0^\alpha (1-l_0)^{-\alpha} \left[ (1-l_0) - \frac{\lambda l_0 (1-\alpha)}{1-\lambda} \right].$$

The iterative process  $V_{n+1}^h = T^h(V_n^h)$  in (4.13) is then effected as follows:

$$\begin{aligned}
 (6.2) \quad V_{n+1}^h(k_0) = & \max_{l_0} \lambda \log \left( \frac{\lambda A k_0^\alpha l_0 (1-\alpha)}{(1-\lambda)(1-l_0)^\alpha} \right) + (1-\lambda) \log l_0 \\
 & + \beta V_n^h \left( A k_0^\alpha (1-l_0)^{-\alpha} \left[ (1-l_0) - \frac{\lambda l_0 (1-\alpha)}{1-\lambda} \right] \right).
 \end{aligned}$$

Although expression (6.2) may appear rather cumbersome, this form will prove appropriate for our computations as it involves maximization only in one single variable.

Our numerical exercises will focus on the following parameterization:

$$\beta = 0.95, \quad \lambda = \frac{1}{3}, \quad A = 10, \quad \alpha = 0.34, \quad \delta = 1.$$

For such values the stationary state is  $k^* = 1.9696$ .

We consider a uniform grid of points  $k^j$  with step size  $h$ . For the purposes of this exercise, the domain of possible capitals is restricted to the interval  $[h, 10]$ . We should remark that in this simple univariate case our interpolations will yield concave value functions  $\{V_n^h\}_{n \geq 1}$  for the method of successive approximations, but the sequence of functions  $\{V_n^h\}_{n \geq 1}$  may not be concave under policy iteration, since each  $V_n^h$  must solve the equation system (4.5) for a given  $g_{n-1}^h$ .

Our numerical computations were coded in standard  $C$  and run on a Silicon Graphics Octane 2 (with a dual processor, each component rated at 600 MHz), which in a double precision floating-point arithmetic allows for a 16-digit accuracy. All required numerical maximizations were effected by Brent's algorithm (cf. Press et al. (1992)) with an accuracy of  $10^{-12}$ . Such a high precision should allow us to trace out the errors derived from other discretizations embedded in our algorithms. Also, for both policy iteration and the method of successive approximations the iteration process will stop once two consecutive value functions  $V_n^h$  and  $V_{n+1}^h$  satisfy the tolerance bound

$$(6.3) \quad \|V_n^h - V_{n+1}^h\| \leq \frac{1}{5}h^2.$$

The adequacy of this stopping rule for the method of successive approximations is discussed in Santos (1999). Roughly, constant  $\frac{1}{5}h^2$  is selected so as to balance the approximation error from the use of a finite grid of points and the truncation error from stopping the iteration process in finite time.

For the method of successive approximations as specified in (6.2), we start each numerical exercise with a given grid size  $h$  and initial condition  $V_0 \equiv 0$ . The program then stops once condition (6.3) is satisfied. For each  $h$ , Table 6.1 reports the number of iterations, computing time, and the maximum observed errors in the last iteration for the value and policy functions.<sup>4</sup> For policy iteration, we follow the procedure specified in (4.4)–(4.5); for each  $h$  the iteration process starts with an initial condition  $V_0 \equiv 0$ , and it stops once condition (6.3) is satisfied. These findings are displayed in Table 6.2.

From the calculations reflected in Tables 6.1 and 6.2, we now discuss the computational cost and speed of convergence of these two algorithms.

(a) *Complexity.* As one can determine from Table 6.1, for  $h = 10^{-1}$  the average time cost per iteration is roughly 0.017 seconds, and for  $h = 10^{-2}$  the average time cost per iteration is roughly 0.17 seconds. Hence, for the method of successive approximations the average time cost per iteration grows linearly with the number of grid points. (This regular pattern is also observed for pairwise comparisons of other grid sizes.) To a certain extent, this result is to be expected since the major computational cost in each iteration is the number of maximizations, which grows linearly with the number of grid points. (Incidentally, this exercise shows that the cost of each maximization remains roughly invariant to the grid size.) In contrast, for policy iteration

<sup>4</sup>These values are defined, respectively, by  $\|V - V_n^h\|$  and  $\|g - g_n^h\|$ , where  $V$  and  $g$  are the closed-form solutions for (6.1), and  $V_n^h$  and  $g_n^h$  are the computed value and policy functions corresponding to the last iteration,  $n$ , under a grid size  $h$ .

TABLE 6.1

Computational method: The method of successive approximations with linear interpolation.<sup>a</sup>

No. of vertex points	Mesh size $h$	Iterations	CPU time	Max. error in $V$	Max. error in $g$
100	$1.00 \times 10^{-1}$	91	1.54	$3.84 \times 10^{-2}$	$5.44 \times 10^{-2}$
300	$3.33 \times 10^{-2}$	128	5.42	$3.44 \times 10^{-3}$	$1.56 \times 10^{-2}$
1000	$1.00 \times 10^{-2}$	181	30.17	$3.68 \times 10^{-4}$	$5.83 \times 10^{-3}$
3000	$3.33 \times 10^{-3}$	223	94.58	$3.42 \times 10^{-5}$	$1.68 \times 10^{-3}$
10000	$1.00 \times 10^{-3}$	271	354.27	$3.36 \times 10^{-6}$	$5.84 \times 10^{-4}$

<sup>a</sup>Parameter values:  $\beta = 0.95$ ,  $\lambda = \frac{1}{3}$ ,  $A = 10$ ,  $\alpha = 0.34$ , and  $\delta = 1$ .

TABLE 6.2

Computational method: Policy iteration with linear interpolation.<sup>b</sup>

No. of vertex points	Mesh size $h$	Iterations	CPU time	Constant $\widehat{\mathcal{L}}^h$	Max. error in $V$	Max. error in $g$
100	$1.00 \times 10^{-1}$	4	0.11	27.35	$4.36 \times 10^{-2}$	$6.136 \times 10^{-2}$
300	$3.33 \times 10^{-2}$	5	2.54	1629.63	$8.47 \times 10^{-4}$	$2.264 \times 10^{-2}$
1000	$1.00 \times 10^{-2}$	7	215.93	19816.63	$8.51 \times 10^{-6}$	$7.160 \times 10^{-3}$
3000	$3.33 \times 10^{-3}$	10	16868.33	78308.85	$1.35 \times 10^{-6}$	$2.400 \times 10^{-3}$

<sup>b</sup>Parameter values:  $\beta = 0.95$ ,  $\lambda = \frac{1}{3}$ ,  $A = 10$ ,  $\alpha = 0.34$ , and  $\delta = 1$ .

we can see from Table 6.2 that for  $h = 10^{-1}$  the average time cost per iteration is roughly 0.03 seconds, whereas for  $h = 10^{-2}$  the average time cost per iteration goes up to about 31 seconds. Hence, for policy iteration an increase in the number of grid points by a factor of 10 leads to an increase in the average time cost by over a factor of  $10^3$ . Again, this result is to be expected since the most complicated step in policy iteration is (4.5), which involves a matrix inversion. This simple complexity analysis illustrates that policy iteration is faster for small grids, but it becomes relatively more costly for fine grids, unless further operational procedures are introduced for the matrix inversion required in (4.5). Under our present methods, it is extremely costly to go beyond grids of 3000 points for policy iteration, whereas we can carry out the method of successive approximations over grids of about 50000 points.

(b) *Convergence.* It is well known that the dynamic programming algorithm approaches the fixed-point solution at a linear rate, and this has been observed in many applications. In order to evaluate the quadratic convergence of policy iteration, we have computed the corresponding constant

$$\widehat{\mathcal{L}}_n^h = \frac{\|V_{\widehat{n}}^h - V_{n+1}^h\|}{\|V_{\widehat{n}}^h - V_n^h\|^2},$$

where  $V_{\widehat{n}}^h$  is the value function of an arbitrarily high iteration,  $\widehat{n}$ , so that  $V_{\widehat{n}}^h$  is a good approximation of the fixed point  $V^h$  (cf. Theorem 5.9). For each  $h$ , in Table 6.2 we report the max value,  $\widehat{\mathcal{L}}^h = \max_n \widehat{\mathcal{L}}_n^h$ . This constant takes on relatively high values, and it seems to grow as predicted by our analysis. Indeed, from the previous section (cf. Remark 5.10) we may conclude that our worst-case theoretical bounding constant  $\widehat{\mathcal{L}} = \frac{\beta \widehat{\mathcal{L}}}{1-\beta}$  is at least  $O(\frac{1}{h^2})$ , which appears to be in line with the observed estimates. The quadratic convergence near the fixed-point solution was further confirmed by a

TABLE 6.3

Computational method: The method of successive approximations with linear interpolation.<sup>c</sup>

No. of vertex points	Mesh size $h$	Iterations	CPU time	Max. error in $V$	Max. error in $g$
100	$1.00 \times 10^{-1}$	460	6.20	$2.09 \times 10^{-1}$	$4.88 \times 10^{-2}$
300	$3.33 \times 10^{-2}$	694	27.75	$1.98 \times 10^{-2}$	$1.63 \times 10^{-2}$
1000	$1.00 \times 10^{-2}$	920	126.02	$1.95 \times 10^{-3}$	$5.58 \times 10^{-3}$
3000	$3.33 \times 10^{-3}$	1126	470.65	$1.95 \times 10^{-4}$	$1.87 \times 10^{-3}$
10000	$1.00 \times 10^{-3}$	1379	1748.29	$1.93 \times 10^{-5}$	$5.99 \times 10^{-4}$

<sup>c</sup>Parameter values:  $\beta = 0.99$ ,  $\lambda = \frac{1}{3}$ ,  $A = 10$ ,  $\alpha = 0.34$ , and  $\delta = 1$ .

detailed analysis of the evolution of the errors  $\|V_n^h - V_n^h\|$ . For the sake of brevity, these results are not reported here.

It may seem paradoxical that the number of required iterations in Table 6.2 does not vary greatly with the grid size. But one could argue that stopping rule (6.3) is suitable for the method of successive approximations, which features a linear rate of convergence, but such a stopping rule is not so sensitive for algorithms displaying faster rates of convergence. The insensitivity on the number of policy steps was also observed as we varied the discount factor. Variations in  $\beta$  were reflected in changes in the above constant  $\hat{\mathcal{L}}^h$ , but the number of required policy iterations was always below 20. For the method of successive approximations, however, the required number of iterations changes substantially with variations in the discount factor. For instance, in Table 6.3 the discount factor is increased from  $\beta = 0.95$  to  $\beta = 0.99$ . Then, as compared to Table 6.1, the number of required iterations and the corresponding computational cost go up by a factor of 5.

As one can see, the bounding constant  $\hat{\mathcal{L}} = \frac{\beta \hat{\mathcal{L}}}{1-\beta}$  of Theorem 5.9 varies inversely with the discount factor  $\beta$ . A related type of dependence under (6.3) can be established for the bounding constant in the method of successive approximations (see (5.18)). But convergence is linear for the method of successive approximations, and hence changes in the bounding constant should necessarily be reflected in changes of the same magnitude in the number of iteration. Policy iteration, however, displays a faster convergence rate. Thus, for changes in the discount factor and corresponding bounding constant, the extra required iteration steps should be of a smaller order of magnitude. Therefore, quadratic convergence seems to be driving the insensitivity of the required number of policy iteration steps under this algorithm for changes in the grid size of the discretization and the discount factor. This does not mean that it is possible to bound the required number of policy iteration steps regardless of the discount factor or the mesh level of the discretization. Indeed, this paper shows that the theoretical constants involved in the orders of convergence get unbounded as the grid size of the discretization converges to zero.

**7. Concluding remarks.** This paper provides new convergence results on the policy iteration algorithm. Our work is motivated by the fact that this algorithm usually converges in a small number of steps (typically fewer than 20) even though the number of feasible policies that the policy iteration algorithm could evaluate is huge and increases exponentially fast with the number of states and possible actions. Unfortunately, in section 2 we presented an example of an MDP problem in which the number of iteration steps grows equally with the number of states, thus dispelling

the hope of proving a general result that only a small number of policy iteration steps will be necessary to solve an MDP problem.

We then focused on the observation that policy iteration is equivalent to Newton's method and thus ought to have quadratic rates of convergence. This equivalence holds even for MDP problems where the state and action spaces are no longer finite sets. This very rapid rate of convergence suggests that policy iteration could dominate the method of successive approximations, at least for discount factors  $\beta$  close to 1. A standard sufficient condition used to establish quadratic convergence is that the derivative of the nonlinear operator defining the system of equations to be solved by Newton's method (which in the case of policy iteration equals the identity minus the derivative of the Bellman operator) satisfies a Lipschitz condition. In the original work of Puterman and Brumelle (1979) this Lipschitz condition was assumed to hold globally, but they did not provide easily verifiable primitive conditions under which this Lipschitz bound could be satisfied. As a result, an open question remains: Under what conditions and for which classes of MDP problems might we obtain quadratic convergence rates for policy iteration?

This paper attempts to address this question by considering a relatively narrow class of dynamic models arising in economic growth theory. This class involves continuous state and control variables, and thus some sort of discretization procedure must be used to implement policy iteration in practical situations. The key Lipschitz condition that guarantees quadratic convergence is shown to hold for piecewise linear interpolation under a concavity assumption at the optimal interior solution, and it may be extended to other approximation schemes. But we also demonstrate that a weaker general bound does hold that involves the *square root* of the maximum absolute difference in the value function from the fixed-point solution. To establish that this weaker bound holds globally, we have imposed a concavity condition on both the return function and the fixed-point solution which are defined on a convex domain of state and control variables. Under this concavity assumption, we were able to prove a global result for the policy iteration algorithm of superlinear convergence with an exponent equal to 1.5. Furthermore, the best bounding constant on the errors of our algorithm is inversely proportional to the grid size  $h$  of the discretization of the state space. These results suggest the following two pessimistic conclusions about policy iteration: (1) As  $h$  goes to zero, not only will the number of states in the approximate MDP problem tend to infinity, but the theoretical convergence bounds (and possibly the number of policy iteration steps) will also tend to infinity as well. (2) Policy iteration will not converge globally at a rapid quadratic rate; quadratic convergence was validated under the concavity assumption and piecewise interpolation, but the general rate of convergence is expected to be superlinear.

Section 6 reported the results of some numerical experiments in which these theoretical results were verified. We corroborated the quadratic convergence of the algorithm, and that the bounding constants evolved as predicted by our analysis. As a result of the computational cost involved in finding the solution in each policy iteration step, our numerical experiments illustrated that for very fine grids policy iteration is substantially slower than simple successive approximations. Thus, our results provide a rather pessimistic perspective on the usefulness of policy iteration for solving large-scale MDP problems. Fine discretizations with large numbers of states are required to approximate the solutions accurately in most applications in economics and engineering. Other studies (see Benitez-Silva et al. (2001)) have suggested that an alternative type of policy iteration known as *parametric policy iteration* can be more effective for solving such large continuous-state MDP problems. Relatively little is

known, however, about the convergence properties of this latter algorithm. In fact, the results of this paper suggest that there is still much to be learned about the convergence properties of the standard policy iteration algorithm.

## REFERENCES

- R. BELLMAN (1955), *Functional equations in the theory of dynamic programming. V. Positivity and quasi-linearity*, Proc. Nat. Acad. Sci. USA, 41, pp. 743–746.
- R. BELLMAN (1957), *Dynamic Programming*, Princeton University Press, Princeton, NJ.
- H. BENITEZ-SILVA, G. HALL, G. HITCH, G. PAULETTO, AND J. RUST (2001), *A Comparison of Discrete and Parametric Approximation Methods for Continuous-state Dynamic Programming Problems*, manuscript, Yale University, New Haven, CT.
- D. BLACKWELL (1965), *Discounted dynamic programming*, Ann. Math. Statist., 36, pp. 226–235.
- G. DAHLQUIST AND A. BJORCK (1974), *Numerical Methods*, MIT Press, Cambridge, MA.
- W. H. FLEMING AND R. W. RISHEL (1975), *Deterministic and Stochastic Optimal Control*, Springer-Verlag, New York.
- R. HOWARD (1960), *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA.
- K. L. JUDD (1998), *Numerical Methods in Economics*, MIT Press, Cambridge, MA.
- V. KLEE AND G. J. MINTY (1972), *How good is the simplex algorithm?*, in Inequalities III, O. Shisha, ed., Academic Press, New York, pp. 159–175.
- L. MONTRUCCHIO (1987), *Lipschitz continuous policy functions for strongly concave optimization problems*, J. Math. Econom., 16, pp. 259–273.
- W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, AND B. P. FLANNERY (1992), *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK.
- M. L. PUTERMAN AND S. L. BRUMELLE (1979), *On the convergence of policy iteration in stationary dynamic programming*, Math. Oper. Res., 4, pp. 60–69.
- R. T. ROCKAFELLAR (1970), *Convex Analysis*, Princeton University Press, Princeton, NJ.
- J. RUST (1987), *Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher*, Econometrica, 55, pp. 999–1033.
- M. S. SANTOS (1999), *Numerical solution of dynamic economic models*, in Handbook of Macroeconomics, J. B. Taylor and M. Woodford, eds., Elsevier Science, Amsterdam, The Netherlands, pp. 311–386.
- M. S. SANTOS (2000), *Accuracy of numerical solutions using the Euler equation residuals*, Econometrica, 68, pp. 1377–1402.
- N. L. STOKEY, R. E. LUCAS WITH E. C. PRESCOTT (1989), *Recursive Methods in Economic Dynamics*, Harvard University Press, Cambridge, MA.
- J. N. TSITSIKLIS (2000), *private communication*, MIT, Cambridge, MA.
- J. F. TRAUB AND H. WOŹNIAKOWSKI (1979), *Convergence and complexity of Newton iteration for operator equations*, J. Assoc. Comput. Mach., 26, pp. 250–258.