ISSN 0819-2642
ISBN 0 7340 2618 8

THE UNIVERSITY OF MELBOURNE

DEPARTMENT OF ECONOMICS

RESEARCH PAPER NUMBER 961

FEBRUARY 2006

CONTINUOUS STATE DYNAMIC PROGRAMMING VIA NONEXPANSIVE APPROXIMATION

by

John Stachurski

Department of Economics
The University of Melbourne
Melbourne Victoria 3010
Australia.

CONTINUOUS STATE DYNAMIC PROGRAMMING VIA NONEXPANSIVE APPROXIMATION

JOHN STACHURSKI

Abstract. This paper studies fitted value iteration for continuous state dynamic programming using nonexpansive function approximators. A number of nonexpansive approximation schemes are discussed. The main contribution is to provide error bounds for approximate optimal policies generated by the value iteration algorithm.

1. Introduction

Most infinite horizon dynamic programming problems are solved using some version of Bellman's principle of optimality, which allows optimal policies to be computed from a two-period program defined using the value function v^*. Bellman's principle of optimality is central to economic modeling not only because it can describe the decision problems of individual agents under rational expectations, but also because many decentralized market equilibria can be obtained as the solution to a corresponding dynamic program.[1]

When no simple analytical representation of v^* is available, a standard algorithm for solving the programming problem is value iteration. Value iteration involves computing an approximate value function by

Date: February 22, 2006.

The author is grateful for financial support from Australian Research Council Grant DP0557625, and for many helpful discussions with Rabee Tourkey.

[1] The set of potential references is far too large to attempt a serious bibliography. Influential applications of dynamic programming to economic problems include McCall (1970), Samuelson (1971), Lucas and Prescott (1971), Brock and Mirman (1972), Hall (1978), Lucas (1978), Kydland and Prescott (1982) and Mehra and Prescott (1985).

iteration of the Bellman operator T on some initial function v.[2] Under mild assumptions T is sup-norm contracting, and the resulting sequence $(T^n v)_{n=1}^{\infty}$ converges geometrically to v^*. The contractiveness of T also yields bounds for the error associated with calculating an approximate optimal policy using $T^n v$ in place of the true value function v^*.

If the state space is infinite, one cannot in general implement the functions $v, Tv, \ldots, T^n v$ on a computer. One solution is to replace the state space with a finite grid, and the original model with a "similar" model which evolves on this reduced state space. A second is fitted value iteration, a typical algorithm for which is

```
initialize $v$
**repeat**
    sample $Tv(x_i)$ at finite set of grid points $\{x_i\}$
    use samples to construct approximation $w \in \mathcal{F}$ of $Tv$
    set $v = w$
**until** a suitable stopping rule is satisfied
```

Here $\mathcal{F}$ is a class of functions with finite parametric representation. The map from v to w is in effect an approximate Bellman operator $\hat{T}$, and fitted value iteration is equivalent to iteration with $\hat{T}$ in place of T.

Approximation maps sending $Tv \mapsto w \in \mathcal{F}$ are typically chosen to minimize some distance measure:

$$w \in \operatorname{argmin}_{\omega \in \mathcal{F}} \|Tv - \omega\|, \quad \| \cdot \| \text{ a suitable norm.} \tag{1}$$

A number of approximation schemes have good performance over the class of functions typically encountered in applied economic modeling. Popular choices include Chebychev polynomials, cubic splines and neural nets.[3]

[2] The excellent survey of Rust (1996) contains an extensive discussion of value iteration, along with other numerical methods for dynamic programming, such as policy iteration and Euler equation methods.

[3] Efficient approximation makes use not only of the observations of Tv on the grid points, but also of any smoothness, convexity and other such properties of Tv, which give information about the function *between* the grid points. Ideally, relatively few grid points are able to convey a large amount of information about Tv,

Note, however, that the ultimate objective is to minimize not (1) but some measure of distance between the optimal policy and the approximate optimal policy computed from $\hat{T}^n v$. In this connection, attention must be paid to whether or not the approximation scheme interacts well with the *iteration* scheme needed to compute the fixed point v^*: a scheme which represents the function Tv well in the sense of (1) at each iteration may still lead to poor dynamic properties for the sequence $(\hat{T}^n v)$. As approximation errors are compounded at each iteration, $\lim_{n\to\infty} \hat{T}^n v$ may deviate substantially from $\lim_{n\to\infty} T^n v = v^*$; in fact the sequence may fail to converge at all.[4]

The key problem here is a lack of compatibility between the sup-norm contraction property of T—which drives convergence of $(T^n v)_{n=1}^{\infty}$ to v^*—and the potentially expansive properties of the approximation. To clarify this point, let us decompose $\hat{T}$ into the action of two operators L and T: First T is applied to v—in practice Tv is evaluated only at finitely many points—and then an approximation operator L sends the result into $w \in \mathcal{F}$. The contractiveness of $\hat{T} = L \circ T$ depends on the contractiveness of L, and L is not generally contracting.[5]

The present paper proceed as follows. Following a suggestion of Gordon (1995), we restrict attention to approximation architectures such that L is nonexpansive with respect to the sup-norm; from which it follows that the composition $\hat{T} := L \circ T$ is a contraction mapping. We exploit the contractiveness of $\hat{T}$ to obtain a set of errors bounds for approximate optimal policies which applies to any nonexpansive approximation architecture.

thereby reducing the number of computations needed to update with the Bellman operator.

[4] See, for example, Tsitsiklis and Van Roy (1996, Section 4), which gives an example of divergence under least-squares approximation.

[5] We should remark that many approximation operators are naturally nonexpansive—particularly those which involve orthogonal projection onto a closed, convex set. However, this nonexpansiveness is with respect to the norm in (1), rather than the sup-norm. A standard choice for the norm in (1) is some version of the L_2 norm (under which the function space in question is a Hilbert space, and the orthogonal projection map is well-defined).

An additional contribution of this paper is to investigate the expansiveness of shape-preserving function approximators. Previously, Judd and Solnick (1994) highlighted the computational advantages of such approximators, where the "shapes" of greatest interest are monotonicity and convexity (concavity). We show that a certain class of shape-preserving quasi-interpolants popular in computer aided design are in fact nonexpansive.

We also observe that when the map L corresponds to a simple nearest neighbor approximation rule—a kind of nonexpansive interpolant—iteration with $L \circ T$ provides an algorithm that can be identified with discretization of the dynamic program. The algorithm is simple to program, admits the use of adaptive grids, and error bounds constructed in the paper all apply. In contrast, the common procedure of replacing a continuous state model with a "similar" discrete model and solving the discrete version permits no adaptation of the grid between iterations, and is relatively difficult to analyze in terms of approximation error.

A brief summary of existing research is as follows. Within the artificial intelligence literature, Gordon (1995) proposed the idea of constructing a general theory of nonexpansive approximations applied to dynamic programming. Drummond (1996) investigated adding penalties to the derivatives of function approximators in order to prevent sup-norm expansiveness (overshooting). Guestrin et al. (2001) study nonexpansive approximations in factored Markov Decision Processes. We add to this literature by establishing error bounds for policies computed using value iteration based on a general nonexpansive approximation operator.[6] Our focus is on structures suitable for economic applications.

Within the economic literature, various studies have been made of approximation architectures which turn out to be nonexpansive. Judd and Solnick (1994) observed that a class of spline interpolants preserve the contraction property of $\hat{T}$, and exploited this fact in their discussion of errors. Santos and Vigo-Aguiar (1998) considered a finite element method using piecewise affine functions. They also observed

[6] See also the important results of Tsitsiklis and Van Roy (1996), who provided error bounds for optimal policies when the state and action spaces are finite.

that their approximation scheme preserve the contraction property of $\hat{T}$. Rust (1997) studies a random discretized Bellman operator which is a probability one contraction.

The paper proceeds as follows. Section 2 formulates the dynamic programming problem. Section 3 discusses nonexpansive approximation schemes. Section 4 considers the measurement of approximation error, and provides some justification for the measure used in this paper. Section 5 states results, and Section 6 gives proofs.

2. Formulation of the Problem

If (U, d) is a metric space, then $\mathcal{B}(U)$ denotes the Borel subsets of U, $C(U)$ is the continuous functions from U to $\mathbb{R}$, $b\mathcal{B}(U)$ is the bounded Borel measurable functions from U to $\mathbb{R}$, and $bC(U) = b\mathcal{B}(U) \cap C(U)$. In what follows, measurability refers to Borel measurability unless otherwise stated. For $f \in b\mathcal{B}(U)$ we let $\|f\|_\infty$ be defined by $\sup_{x \in U} |f(x)|$. Further, d_∞ denotes the metric on $b\mathcal{B}(U)$ associated with this norm.[7] A map $M \colon U \to U$ is called *nonexpansive* if it satisfies the condition

$$d(Mw, Mw') \leq d(w, w'), \quad \forall w, w' \in U; \tag{2}$$

and a *contraction of modulus* ϱ if there exists a $\varrho \in [0, 1)$ with

$$d(Mw, Mw') \leq \varrho d(w, w'), \quad \forall w, w' \in U. \tag{3}$$

Let M_1 and M_2 be two maps from the space U to itself. It is trivial to show that if M_1 is a contraction of modulus ϱ and M_2 is nonexpansive, then $M_2 \circ M_1$ is a contraction of modulus ϱ.

Consider the following abstract infinite horizon stochastic dynamic programming problem, defined by a tuple $(S, A, \Gamma, r, \varrho, \mathbf{M})$. Here S is a state space, A is an action space, and Γ is a nonempty correspondence mapping S into $\mathcal{B}(A)$, with $\Gamma(x)$ interpreted as the set of feasible actions when the current state is x. Both S and A are Borel subsets of finite-dimensional Euclidean space.

Given S, A and Γ, define

$$K := \{(x, u) \in S \times A \,:\, u \in \Gamma(x)\}.$$

[7] Both $(bC(U), d_\infty)$ and $(b\mathcal{B}(U), d_\infty)$ are complete metric spaces.

This collection of points is called the set of all feasible state/action pairs. The map $r\colon K \to \mathbb{R}$ is a measurable "reward" function, while $\varrho \in (0,1)$ is a discount factor, and $\mathbf{M}(x,u;dy)$ is a distribution over S for each feasible state/action pair $(x,u) \in K$. Here $\mathbf{M}(x,u;B)$ should be be interpreted as the conditional probability that next period state $X_{t+1} \in B$ when the current state $X_t = x$ and the current action $U_t = u$.[8] For example, if the future state is determined according to

$$X_{t+1} = F(X_t, U_t, W_{t+1}), \tag{4}$$

where $(W_t)_{t=1}^{\infty}$ is a sequence of independent shocks with distribution φ, then

$$\mathbf{M}(x,u;B) = \int \mathbb{1}_B[F(x,u,z)]\varphi(dz). \tag{5}$$

The system evolves as follows. At the start of time, the agent observes $X_0 = x_0 \in S$, where x_0 is some fixed initial condition, and then chooses action $U_0 \in \Gamma(X_0) \subset A$. After choosing U_0, the agent receives a reward $r(X_0, U_0)$. The next state X_1 is now drawn according to distribution $\mathbf{M}(X_0, U_0; dy)$ and the process repeats, with the agent choosing U_1, receiving reward $r(X_1, U_1)$, and so on.

Let Π denote the set of all measurable functions $\pi\colon S \to A$ with $\pi(x) \in \Gamma(x)$ for all $x \in S$. We refer to Π as the set of *feasible policies*. Each fixed policy $\pi \in \Pi$ and initial condition $x_0 \in S$ defines a Markov chain (X_t), where X_0 is set equal to x_0, and then, recursively, X_{t+1} is drawn from $\mathbf{M}(X_t, \pi(X_t); dy)$. We let $\mathbf{P}_\pi^{x_0}$ denote the joint distribution on the sequence space $(S^\infty, \otimes_{n=1}^{\infty}\mathcal{B}(S))$ associated with this chain, while $\mathbf{E}_\pi^{x_0}$ denotes the expectation operator corresponding to $\mathbf{P}_\pi^{x_0}$.

To set up the problem, we define a function from $\Pi \times S$ into $\mathbb{R}$ by

$$v_\pi(x_0) := \mathbf{E}_\pi^{x_0}\left[\sum_{t=0}^{\infty} \varrho^t r(X_t, \pi(X_t))\right]. \tag{6}$$

Thus $v_\pi(x_0)$ is the value of following the policy π when starting at initial condition x_0. The optimization problem is then given by $\max_{\pi\in\Pi} v_\pi(x_0)$,

[8] Formally, by a distribution on S is meant a probability measure on $(S, \mathcal{B}(S))$. In addition, $(x,u) \mapsto \mathbf{M}(x,u;B)$ is required to be measurable, $\forall B \in \mathcal{B}(S)$.

where x_0 is regarded as fixed. The *value function* $v^*\colon S \to \mathbb{R}$ is defined as

$$v^*(x_0) = \sup_{\pi \in \Pi} v_\pi(x_0), \quad x_0 \in S. \tag{7}$$

A policy $\pi^* \in \Pi$ is called *optimal* if it attains the supremum in (7) for every $x_0 \in S$. In other words, $\pi^* \in \Pi$ is optimal if and only if v_{π^*} and v^* are the same function.

Assumption 2.1. The map r is continuous and bounded on K, while Γ is continuous and compact valued. Further,

$$(x, u) \mapsto \int w(y)\mathbf{M}(x, u; dy)$$

is continuous as a map from K to $\mathbb{R}$ whenever $w \in bC(S)$.[9]

The following theorem is a standard optimality result.[10]

Theorem 2.1. *Under Assumption 2.1, the value function v^* is the unique function in $b\mathcal{B}(S)$ which satisfies*

$$v^*(x) = \sup_{u \in \Gamma(x)} \left\{ r(x, u) + \varrho \int v^*(y)\mathbf{M}(x, u; dy) \right\}, \quad \forall\, x \in S. \tag{8}$$

In fact v^* is continuous, and we can replace sup *with* max *in* (8). *If $\pi^* \in \Pi$ and*

$$v^*(x) = r(x, \pi^*(x)) + \varrho \int v^*(y)\mathbf{M}(x, \pi^*(x); dy), \quad \forall\, x \in S, \tag{9}$$

then π^* is optimal. At least one such optimal policy $\pi^* \in \Pi$ exists. Conversely, if π^* is an optimal policy then it satisfies (9).

Two kinds of contraction mappings are used to study the optimality results. First, let $T_\pi\colon b\mathcal{B}(S) \to b\mathcal{B}(S)$ be defined for all $\pi \in \Pi$ by

$$T_\pi w(x) = r(x, \pi(x)) + \varrho \int w(y)\mathbf{M}(x, \pi(x); dy). \tag{10}$$

Further, let $T\colon b\mathcal{B}(S) \to b\mathcal{B}(S)$ be defined by

$$Tw(x) = \sup_{u \in \Gamma(x)} \left\{ r(x, u) + \varrho \int w(y)\mathbf{M}(x, u; dy) \right\}. \tag{11}$$

[9] This last assumption is a version of the so-called Feller property. See, for example, Stokey, Lucas and Prescott (1989, Chapter 8).

[10] See, for example, Hernández-Lerma and Lasserre (1999, § 8.5).

The second operator T is usually called the *Bellman operator*. Using the Bellman operator we can restate the first part of Theorem 2.1 as: v^* is the unique fixed point of T in $b\mathcal{B}(S)$.

It is well-known that for every $\pi \in \Pi$, the operator T_π is a contraction on $(b\mathcal{B}(S), d_\infty)$ of modulus ϱ. The unique fixed point of T_π in $b\mathcal{B}(S)$ is v_π, where the definition of v_π is given in (6). In addition, T_π is monotone on $b\mathcal{B}(S)$, in the sense that if $w, w' \in b\mathcal{B}(S)$ and $w \leq w'$, then $T_\pi w \leq T_\pi w'$.[11] Similarly, the Bellman operator is also a contraction of modulus ϱ; and monotone on $b\mathcal{B}(S)$.[12]

3. The Approximation Operator

To carry out fitted value iteration we use an approximation operator L which maps $b\mathcal{B}(S)$ into a collection of functions $\mathcal{F} \subset b\mathcal{B}(S)$. In general, L constructs an approximation $Lv \in \mathcal{F}$ to $v \in b\mathcal{B}(S)$ according to a sample $\{v(x_i)\}_{i=1}^k$ of evaluations of v on grid points $\{x_i\}_{i=1}^k$. As discussed in the introduction, we focus on approximation architectures with the property that L is nonexpansive with respect to d_∞:

$$\|Lv - Lw\|_\infty \leq \|v - w\|_\infty, \quad \forall v, w \in b\mathcal{B}(S). \tag{12}$$

We assume further that L is a projection, in the sense that $L \circ L = L$ on $b\mathcal{B}(S)$. In particular, if $v \in \mathcal{F}$, then v is a fixed point of L.

Example 3.1. (**Nearest neighbors**) An elementary class of nonexpansive maps is provided by k-nearest neighbors approximation, the simplest version of which is when $k = 1$. For this specification $Lv(x)$ is set to $v(x_i)$, where $i = \operatorname{argmin}_j \|x - x_j\|$.[13] Thus Lv takes only finitely many values. Moreover, it is clear that

$$\|Lw - Lv\|_\infty \leq \sup_{1 \leq i \leq k} |w(x_i) - v(x_i)|, \quad \forall w, v \in b\mathcal{B}(S). \tag{13}$$

In particular, L is nonexpansive on $b\mathcal{B}(S)$ with respect to the sup norm.

Interestingly, iteration with $\hat{T} = L \circ T$ provides an implementation of discretization for dynamic programs: Let $\hat{v}_n := \hat{T}^n v$, so that $\hat{v}_n$ takes

[11] Here inequalities such as $w \leq w'$ are pointwise inequalities on S.

[12] These results are standard. See, for example, Puterman (1994), Stokey, Lucas and Prescott (1989) or Hernández-Lerma and Lasserre (1999).

[13] Here $\|\cdot\|$ is the Euclidean norm on S.

finitely many values $\hat{v}_n(x_i)$, $1 \leq i \leq k$. Let B_i be the subset of the state S on which $\hat{v}_n$ takes the value $\hat{v}_n(x_i)$. We can now obtain $T\hat{v}_n$ at the grid point x_j via

$$\begin{aligned} T\hat{v}_n(x_j) &= \sup_{u \in \Gamma(x_j)} \left\{ r(x_j, u) + \varrho \int \hat{v}_n(y) \mathbf{M}(x_j, u; dy) \right\} \\ &= \sup_{u \in \Gamma(x_j)} \left\{ r(x_j, u) + \varrho \sum_{i=1}^{k} \hat{v}_n(x_i) \mathbf{M}(x_j, u; B_i) \right\}. \end{aligned}$$

These values $T\hat{v}_n(x_1), \ldots, T\hat{v}_n(x_k)$ define $LT\hat{v}_n = \hat{T}^{n+1}v = \hat{v}_{n+1}$, and the iteration proceeds.[14]

There are several advantages to this form of discretization. First, since L is nonexpansive the error bounds developed below all apply. Second, the reward function r is never discretized, and nor need it be—presumably the primitive r can be implemented without discretization. Finally, it is possible to adjust the location and size of the grid at each iteration.[15]

Example 3.2. (**Kernel averagers**) Kernel-based approximation methods provide a class of smooth approximation architectures which have attracted much attention in recent years, partly because they are simple to implement in high-dimensional state spaces. One of these methods is the so-called kernel averages, which typically can be represented by an expression of the form

$$Lv(x) = \frac{\sum_{i=1}^{k} K_h(x_i - x) v(x_i)}{\sum_{i=1}^{k} K_h(x_i - x)}. \tag{14}$$

Here the kernel K_h is a nonnegative mapping from $S \to \mathbb{R}$ such as the radial basis function $e^{-\|\cdot\|/h}$. The value of the kernel decays to zero as x diverges from x_i. Thus, $Lv(x)$ is a convex combination of the observations $v(x_1), \ldots, v(x_k)$ with larger weight being given to those observations $v(x_i)$ for which x_i is close to x. The smoothing parameter h controls the weight assigned to more distant observations.

[14] In high dimensions it may be more efficient to evaluate the terms $\mathbf{M}(x_j, u; B_i)$ by Monte Carlo rather than numerical integration. See Rust (1997) for more discussion of Monte Carlo methods in high-dimensional problems.

[15] A number of algorithms use variable grids for discretized dynamic programming. See, for example, Rust (1997).

The following lemma (Gordon, 1995) is elementary but useful. It shows that the approximation operators associated with kernel averagers are nonexpansive with respect to d_∞. It also provides an upper bound for the d_∞-distance between Lw and Lv which can be computed exactly.

Lemma 3.1. *The operator L in* (14) *satisfies* (13)*. In particular, L is nonexpansive with respect to the sup norm.*

Proof. Pick any $x \in S$, and let $\lambda(x,i) := K_h(x_i - x) / \sum_{j=1}^k K_h(x_j - x)$. Using $\sum_{i=1}^k \lambda(x,i) = 1$, we have

$$\begin{aligned} |Lw(x) - Lv(x)| &= \left| \sum_{i=1}^k \lambda(x,i)(w(x_i) - v(x_i)) \right| \\ &\leq \sum_{i=1}^k \lambda(x,i) |w(x_i) - v(x_i)| \leq \sup_{1 \leq i \leq k} |w(x_i) - v(x_i)|. \end{aligned}$$

Since x is arbitrary the claim in the lemma holds. □

Example 3.3. (**Continuous piecewise linear interpolation**) A common form of approximation in dynamic programming is piecewise linear (piecewise affine) spline interpolation.[16] To describe a general set up, let $\{x_i\}_{i=1}^k$ be a finite subset of $S \subset \mathbb{R}^d$ with the property $\text{c.hull}\{x_i\}_{i=1}^k = S$, and let $\mathcal{T}$ be a triangularization of S relative to the nodes $\{x_i\}_{i=1}^k$.[17] In other words, $\mathcal{T}$ is a partition of S into a finite collection of non-overlapping, non-degenerate simplexes, where, for each $\Delta \in \mathcal{T}$, the set of vertices $\{\zeta_i\}_{i=1}^{d+1} \subset \{x_i\}_{i=1}^k$.[18]

Each $x \in \Delta$ can be represented uniquely by its barycentric coordinates relative to Δ:

$$x = \sum_{i=1}^{d+1} \lambda(x,i)\zeta_i, \quad \text{where } \lambda(x,i) \geq 0, \ \sum_{i=1}^{d+1} \lambda(x,i) = 1.$$

For $v \in b\mathcal{B}(S)$ we define the interpolation operator L by

$$Lv(x) = \sum_{i=1}^{d+1} \lambda(x,i) v(\zeta_i).$$

[16] See, for example, Santos and Vigo-Aguiar (1998) and Munos and Moore (1999).

[17] Here $\text{c.hull}\{x_i\}_{i=1}^k$ is the convex hull of $\{x_i\}_{i=1}^k$.

[18] A simplex is called non-degenerate if it has positive measure in $\mathbb{R}^d$.

An argument similar to the proof of Lemma 3.1 shows that if $v, w \in b\mathcal{B}(S)$, then at x we have

$$|Lw(x) - Lv(x)| \leq \sup_{1 \leq i \leq d+1} |w(\zeta_i) - v(\zeta_i)| \leq \|w - v\|_\infty.$$

Since x is arbitrary, L is clearly nonexpansive.

Example 3.4. (**Schoenberg's variation diminishing operator**) In a well-known study, Judd and Solnick (1994) emphasize the advantages of fitted value iteration with shape-preserving approximators; here the shapes of greatest interest are monotonicity and convexity, and approximators which preserve them not only incorporate known structure from the target function in the approximating function, they also allow monotonicity and convexity to be exploited in the optimization step of the value iteration algorithm.[19]

Judd and Solnick discuss several univariate shape-preserving architectures, including (nonsmooth) univariate piecewise linear interpolants and (smooth) Schumaker splines. Here we describe a further class of smooth, shape-preserving approximators known as Schoenberg variation diminishing splines. Variation diminishing splines are extremely popular in applications such Computer Aided Geometric Design both for their shape preserving properties and for their simplicity—which in turn gives fast evaluation. An easy argument shows that the approximation operator associated with variation diminishing splines is not only smooth and shape-preserving, but also nonexpansive.

To construct the operator we set $S = [a, b] \subset \mathbb{R}$, and in place of a standard grid we use for each $d \in \mathbb{N}$ a $d+1$-regular knot sequence $(t_i)_{i=1}^{k+d+1}$, which satisfies

$$a = t_1 = \cdots = t_{d+1} < t_{d+2} < \cdots < t_{k+1} = \cdots = t_{k+d+1} = b.$$

Here d is the order of the spline, so that, for example, $d = 3$ corresponds to a cubic spline. The Schoenberg splines are built using k

[19] Monotonicity is exploited as follows: In monotone programs the optimal action is often increasing in the state, in which case one need not search for optimal actions in that subset of the action space which is dominated by the optimal action at a lower state. The importance of convexity in optimization needs no illustration here.

basis functions which are known as B-splines. The latter are defined recursively by

$$B_{i,0} := \mathbb{1}_{[t_i, t_{i+1})}, \quad i = 1, \ldots, k,$$

and then, $i = 1, \ldots, k$,

$$B_{i,d}(x) := \frac{x - t_i}{t_{i+d} - t_i} B_{i,d-1}(x) + \frac{t_{i+d+1} - x}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(x),$$

where in the definition we are using the convention that $0/0 = 0$. For fixed d the basis functions $B_{1,d}, \ldots, B_{k,d}$ are linearly independent and satisfy

$$\sum_{i=1}^{k} B_{i,d} = \mathbb{1}_S.$$

Their span is often denoted by $\mathbb{S}_d$:

$$\mathbb{S}_d := \left\{ \sum_{i=1}^{k} \alpha_i B_{i,d} \, : \, (\alpha_1, \ldots, \alpha_k) \in \mathbb{R}^k \right\}.$$

Clearly $\mathbb{S}_d \subset b\mathcal{B}(S)$. Schoenberg's variation diminishing operator is now given by

$$L \colon b\mathcal{B}(S) \ni v \mapsto \sum_{i=1}^{k} v(t_i^*) B_{i,d} \in \mathbb{S}_d,$$

where $t_i^* := (t_{i+1} + \cdots + t_{i+d})/d$.

It is well-known that L preserves monotonicity and convexity (concavity) in v.[20] It is easy to see that L is also nonexpansive:

Lemma 3.2. *Schoenberg's variation diminishing operator is nonexpansive as a map from $(b\mathcal{B}(S), d_\infty)$ to itself.*

The proof is very similar in spirit to that of Lemma 3.1 and is omitted.

[20] See, for example, Lyche and Mørken (2002, Chapter 5).

4. A Digression on Measurement of Error

Any analysis of approximation methods requires a measurement of error. One algorithm is determined to be better than another when it produces an approximate solution with smaller error than the other for a given amount of computational effort. Conversely, one cannot rank two algorithms or approximation methods unless error measurement is specified in advance. In this section we discuss appropriate measurements of error from the perspective of economic modeling, arguing in favor of an approach which measures "behavioral" rather than geometric error.

To fix ideas, consider again the dynamic programming problem formulated in Section 2. Let π^* be an optimal policy, and let $\hat{\pi}$ be an approximation. The error associated with $\hat{\pi}$ is often measured as either

$$e_1(\hat{\pi}) = \sup_{x \in S} |\pi^*(x) - \hat{\pi}(x)|$$

or

$$e_2(\hat{\pi}) = \left(\int (\pi^*(x) - \hat{\pi}(x))^2 dx \right)^{1/2}.$$

The first measures the least upper bound of the pointwise deviation between $\hat{\pi}$ and the target π^*, while the second is the so-called L_2 distance. The former is often preferred because it is easy to interpret. On the other hand, e_1 is very sensitive to local deviations—even those on sets of measure zero which in simulations have no influence on time series generated by the model. For this reason some authors prefer the L_2 distance, which ignores deviation on sets of zero measure.[21]

The issue is further complicated by the existence of other viable error measures. For example, one might also favor the L_1 distance $\int |\hat{\pi} - \pi^*|$, or a measure such as $\sup_{x \in S} (\pi^*(x) - \hat{\pi}(x))^2$, which gives more than proportional penalty to large deviations. In choosing between these error measures, the problem we are facing is that in a function space such as Π there is no universal measure of "closeness." To determine

[21] See, for example, Munos (2005). Reiter (2001) makes a similar point. The argument for the L_2 norm is more compelling if the norm is weighted by the stationary distribution of the state variables under the approximate optimal policy.

when one approximation is better than another one must take a stand on how closeness (and hence errors) should be determined.

In doing so, the economic modeler requires a loss function over the set of policies Π which indicates the cost (to the modeler) of deviating from the optimal policy as a result of approximation error. From a scientific perspective, good approximations should lead to effective tests for whether the model is correct. In other words, good approximations must accurately reflect the testable implications of the model—in which case a suitable rule for the loss function is that the modeler prefers approximations which correspond well to the predictions of the model over those which correspond poorly. The optimal policy itself corresponds exactly to the predictions of the model, and hence incurs no loss.

Consider, for example, the Euler residual techniques studied by Judd (1992, 1998), Den Haan and Marcet (1994), Santos (2000), Reiter (2001) and others. Errors are assessed by inserting the time series generated by approximate optimal policies into the corresponding Euler equations. For example, a well-known optimal growth model due to Brock and Mirman (1972) has an Euler equation of the form

$$u'(c_t) = \varrho \mathbf{E}_t u'(c_{t+1}) f'(k_{t+1}, z_{t+1}), \tag{15}$$

where u is utility, c is consumption, f is a production function, k is capital and z is a shock. The argument is that if a given policy produces consumption paths which fit (15) poorly then we are unlikely to observe such behavior by agents, as a violation of (15) indicates there are incentives for the agent to transfer consumption across time periods until equality holds. The size of the error in (15) corresponds to the degree of incentive to modify behavior.

While Euler residual methods are not always applicable—in that they require smooth primitives and interiority of optimal choices—here we adopt the essential principle: Approximation $\hat{\pi}_1$ is preferred to approximation $\hat{\pi}_2$ if $\hat{\pi}_1$ is more compatible with the economic incentives of agents in the model. The rationale is that if agents respond to incentives, then $\hat{\pi}_1$ is a more plausible description of the behavioral implications of the model than $\hat{\pi}_2$. This is true regardless of whether

$e(\hat{\pi}_1) > e(\hat{\pi}_2)$ or vice versa, where e is one of the error measures such as e_1 or e_2 discussed above.

One can always test the incentive compatibility of approximation $\hat{\pi}$ in a straightforward way by evaluating the value-loss

$$\mathcal{E}(\hat{\pi}) := v^*(x_0) - v_{\hat{\pi}}(x_0), \quad \hat{\pi} \in \Pi.$$

For example, consider a single monopolist whose objective is to maximize the present discounted value of a stream of net profits. Let x_0 be the initial condition for the state variable, let $v_\pi(x_0)$ be the value of following policy π as in (6), and let v^* be the value function, so $v^*(x_0)$ is the maximum (net present value of) profit from x_0. In this case the agent's incentives are *by definition* dictated by the profit stream, and approximation $\hat{\pi}_1$ is preferred to $\hat{\pi}_2$ if and only if $v_{\hat{\pi}_1}(x_0) > v_{\hat{\pi}_2}(x_0)$; equivalently, $\mathcal{E}(\hat{\pi}_1) < \mathcal{E}(\hat{\pi}_2)$.

Notice that in making this argument one need not take a position on the cognitive processes that underpin rational behavior. The actors in the economy represented by the monopolist agent can be viewed as solving optimization problems, or responding to price signals such as market valuation, or they can be seen as the product of an evolutionary process where poorly managed firms do not survive. In either case, if $v_{\hat{\pi}_1}(x_0) < v_{\hat{\pi}_2}(x_0)$, then the higher profit stream generated by $\hat{\pi}_1$ implies that this behavior corresponds more closely to the predictions of the model than does that implied by $\hat{\pi}_2$.

In most fields the value-loss measure $\mathcal{E}$ is the de facto standard for measuring approximation error for policies, and we have argued that the same should be true of economics.[22] Below, all our error bounds are stated in terms of $\mathcal{E}$-error. Before presenting them we conclude this section with a second example of the suitability of the measure $\mathcal{E}$ which concerns a decentralized market involving many agents. The model is Samuelson's (1971) famous theory of price equilibrium in a commodity market with speculative investment.

[22] See, for example, Puterman (1994, Theorem 6.3.1) or Hernández-Lerma and Lasserre (1999, Proposition 8.4.2). Santos (2000) and Reiter (2001) both link Euler equation residuals to value-loss.

In brief, the model describes intertemporal equilibrium in a single commodity market with two sources of demand: final consumption demand c_t determined by inverse demand function P ($p_t = P(c_t)$), and speculative demand q_t. In equilibrium these demands sum to the total supply at t, denoted by y_t. This supply y_t consists of the "harvest" H_t plus aq_{t-1}, where $a < 1$ is a "shrinkage" parameter and q_{t-1} is carryover from the last period. The harvest process (H_t) is independent and identically distributed.

For fixed interest rate r, the system of prices and path for carryover and consumption must satisfy the arbitrage conditions

$$(1+r)^{-1} a \mathbf{E}_t p_{t+1} - p_t \leq 0, \quad t \geq 0, \tag{16}$$

$$q_t\{(1+r)^{-1} a \mathbf{E}_t p_{t+1} - p_t\} = 0, \quad t \geq 0. \tag{17}$$

As Samuelson points out, one can construct an equilibrium path for prices, consumption and carryover by setting out the problem of a fictitious social planner with discount factor $(1+r)^{-1}$ and period utility function $U(c) = \int_0^c P(x)dx$. The resulting dynamic program

$$\max \mathbf{E} \left[\sum_{t=0}^{\infty} (1+r)^{-t} U(c_t) \right] \tag{18}$$

$$\text{s.t. } c_t + q_t = y_t, \quad y_{t+1} = aq_t + H_t, \quad y_0 \text{ given}$$

has Karush–Khun–Tucker first order optimality conditions given by

$$(1+r)^{-1} a \mathbf{E}_t U'(c_{t+1}) - U'(c_t) \leq 0, \quad t \geq 0, \tag{19}$$

$$q_t\{(1+r)^{-1} a \mathbf{E}_t U'(c_{t+1}) - U'(c_t)\} = 0, \quad t \geq 0, \tag{20}$$

and setting $p_t = U'(c_t)$ produces an equilibrium system satisfying the arbitrage conditions (16) and (17), along with $p_t = P(c_t)$. The system is fully defined by an optimal carryover function π^* which solves (18).

Observe that the process for prices, consumption, carryover and stock generated by carryover policy $\hat{\pi}_1$ accords better with the incentives of agents in the market than those generated by carryover policy $\hat{\pi}_2$ precisely when $\mathcal{E}(\hat{\pi}_1) < \mathcal{E}(\hat{\pi}_2)$, or, equivalently, $v_{\hat{\pi}_1}(y_0) > v_{\hat{\pi}_2}(y_0)$. Lower loss (higher value) equates to greater consumer surplus, which in turn

```
initialize $v \in b\mathcal{B}(S)$ and tolerance $\delta$
repeat
    sample $Tv$ at a finite set of grid points
    compute $\hat{T}v = L \circ Tv$ from the samples
    set $e = \|\hat{T}v - v\|_\infty$ and then $v = \hat{T}v$
until $e \leq \delta$
solve for a policy $\pi \in \Pi$ with
$\pi(x) \in \operatorname{argmax}_{u \in \Gamma(x)} \left\{ r(x,u) + \varrho \int v(y) \mathbf{M}(x,u;dy) \right\}$
```

Figure 1. Approximate Value Iteration Algorithm

means that $\hat{\pi}_1$ realizes more of the potential gains from trade. Approximation $\hat{\pi}_2$, on the other hand, generates a lower consumer surplus, and unexploited gains from trade mean larger violation of the arbitrage conditions, putting the associated price process under greater pressure. Policy $\hat{\pi}_1$ therefore accords better with the predictions of the model than does $\hat{\pi}_2$.

5. Results

In all of what follows, $L\colon b\mathcal{B}(S) \to \mathcal{F}$ is a nonexpansive approximation operator (Section 3). The construction of that operator in turn depends on a fixed set of grid points $\{x_i\}_{i=1}^k \subset S$ as discussed above. The map $\hat{T} := L \circ T$ is the approximate Bellman operator, $\hat{T}^n$ is n compositions of $\hat{T}$ with itself, and $\hat{v}$ is the unique fixed point of $\hat{T}$ in $b\mathcal{B}(S)$.

Consider the approximate value iteration algorithm in Figure 1. We wish to bound the deviation $v^*(x_0) - v_\pi(x_0)$, where $x_0 \in S$ is the initial condition for the dynamic programming problem. Here $v^*(x_0)$ is of course the value of the optimal policy, and $v_\pi(x_0)$ is the value of the policy π produced in the final step—the function $v_\pi \in b\mathcal{B}(S)$ is defined by (6).

If the fixed point $\hat{v}$ of $\hat{T}$ is equal to the fixed point v^* of T, and if in addition we can take the limit of $\hat{T}^n v$ and so compute $\hat{v}$ exactly, then there is no error, and the policy π chosen using $\hat{v}$ is optimal. Actually neither of these two conditions hold in practice, and in fact they provide a natural decomposition of errors into two components: The first is the deviation of $\hat{v}$ from v^*, which results from imperfect approximation under L. The second error is the deviation of $\hat{T}^n v$ from $\hat{v}$, and should be stated in terms of the distance between $\hat{T}^n v$ and $\hat{T}^{n+1} v$ or some other observable, as the deviation of $\hat{T}^n v$ from $\hat{v}$ cannot be computed directly.

We use this error decomposition to give an error bound for the value of the approximate optimal policy. Combining ideas found in Puterman (1994), Judd and Solnick (1994), Gordon (1995), Rust (1996) and Santos and Vigo-Aguiar (1998), the value of the approximate optimal policy is shown to deviate from that of the optimal policy by less than a bound determined by $\sup_{w \in \mathcal{V}} \|w - Lw\|_\infty$ and $\|\hat{T}^{n+1} v - \hat{T}^n v\|_\infty$. Here $\|\cdot\|_\infty$ is the sup-norm, and $\mathcal{V}$ is a class of functions containing v^*. The first term $\sup_{w \in \mathcal{V}} \|w - Lw\|_\infty$ indicates the performance of the approximation map L. The second term $\|\hat{T}^{n+1} v - \hat{T}^n v\|_\infty$ is an observable error which can be used to test a stopping rule in the iteration algorithm.

In the first theorem below, we require that, as well as v^*, the sequence $(T\hat{T}^n v)_{n=1}^\infty$ also lies in $\mathcal{V}$.

Theorem 5.1. *Let L be nonexpansive, let δ is the tolerance for the stopping rule and let $\mathcal{V}$ be a class of functions in $b\mathcal{B}(S)$ containing v^* and the sequence $(T\hat{T}^n v)_{n=1}^\infty$. If π is the policy generated by the approximate value iteration algorithm, then for all initial conditions $x_0 \in S$ we have*

$$v^*(x_0) - v_\pi(x_0) \leq \frac{2}{1-\varrho}\left(\varrho\delta + \sup_{w \in \mathcal{V}} \|w - Lw\|_\infty\right).$$

Remark 5.1. The bound in Theorem 5.1 should be compared to the bound $v^*(x_0) - v_\pi(x_0) \leq 2\varrho\delta/(1-\varrho)$ given by Puterman (1994, Theorem 6.3.1) for the finite state case, where no approximation is used and value iteration can be carried out exactly. In the present case, if there is no approximation error and $\sup_{w \in \mathcal{V}} \|w - Lw\|_\infty = 0$, then the bound in Theorem 5.1 reduces to Puterman's bound. This suggests that our bound is relatively tight.

Remark 5.2. It may seem that the error $e = \|\hat{T}v - v\|_\infty$ in the value iteration algorithm will be difficult to evaluate accurately. However, since both v and $\hat{T}v$ lie in the simple parametric class $\mathcal{F}$, evaluation of the error is in practice usually straightforward.

Now we turn to the second theorem of the paper. Here, our objective is the weaken the assumptions of Theorem 5.1. In particular, the assumption that $T\hat{T}^n v$ lies in a simple class $\mathcal{V}$ for each n may be too strict. In contrast, one often has a considerable amount of information about v^* which can be used to assess the approximation error $\|v^* - Lv^*\|_\infty$. The next bound uses only this information, but at the cost of a larger constant term:

Theorem 5.2. *Let L and δ be as in Theorem 5.1. If π is the policy generated by the approximate value iteration algorithm, then for all initial conditions $x_0 \in S$ we have*

$$v^*(x_0) - v_\pi(x_0) \leq \frac{2}{(1-\varrho)^2} \left(\varrho\delta + \|v^* - Lv^*\|_\infty\right).$$

The proofs of Theorems 5.1 and 5.2 are given below.

6. Proofs

Let us now address the proof of Theorem 5.1. Since the initial condition x will vary according to the problem, we construct a bound on the deviation $v^*(x) - v_\pi(x)$ which is uniform over $x \in S$. In practice, this is done by bounding the sup-norm error $\|v^* - v_\pi\|_\infty$. Using the triangle inequality, the sup-norm error is broken down as

$$\|v^* - v_\pi\|_\infty \leq \|v^* - \hat{T}^{n+1}v\|_\infty + \|\hat{T}^{n+1}v - v_\pi\|_\infty, \tag{21}$$

where $v \in b\mathcal{B}(S)$ is the initial condition in the value iteration algorithm. The next lemma bounds the first of these two errors on the right hand side of (21) in terms of the stopping rule error $\|\hat{T}^{n+1}v - \hat{T}^n v\|_\infty$ and the approximation error $\|v^* - Lv^*\|_\infty$.

Lemma 6.1. *For every $n \in \mathbb{N}$ we have*

$$(1-\varrho)\|v^* - \hat{T}^{n+1}v\|_\infty \leq \|v^* - Lv^*\|_\infty + \varrho\|\hat{T}^{n+1}v - \hat{T}^n v\|_\infty.$$

Proof. Fix $n \in \mathbb{N}$. By the triangle inequality,

$$\|v^* - \hat{T}^{n+1}v\|_\infty \leq \|v^* - \hat{v}\|_\infty + \|\hat{v} - \hat{T}^{n+1}v\|_\infty. \tag{22}$$

Regarding the first term in the sum (22), we have

$$\begin{aligned}\|v^* - \hat{v}\|_\infty &\leq \|v^* - \hat{T}v^*\|_\infty + \|\hat{T}v^* - \hat{v}\|_\infty \\ &= \|v^* - Lv^*\|_\infty + \|\hat{T}v^* - \hat{T}\hat{v}\|_\infty \\ &\leq \|v^* - Lv^*\|_\infty + \varrho\|v^* - \hat{v}\|_\infty.\end{aligned}$$

$$\therefore \quad (1-\varrho)\|v^* - \hat{v}\|_\infty \leq \|v^* - Lv^*\|_\infty. \tag{23}$$

Regarding the second term in the sum (22), we have

$$\begin{aligned}\|\hat{v} - \hat{T}^{n+1}v\|_\infty &\leq \|\hat{v} - \hat{T}^{n+2}v\|_\infty + \|\hat{T}^{n+2}v - \hat{T}^{n+1}v\|_\infty \\ &\leq \varrho\|\hat{v} - \hat{T}^{n+1}v\|_\infty + \varrho\|\hat{T}^{n+1}v - \hat{T}^{n}v\|_\infty.\end{aligned}$$

$$\therefore \quad (1-\varrho)\|\hat{v} - \hat{T}^{n+1}v\|_\infty \leq \varrho\|\hat{T}^{n+1}v - \hat{T}^{n}v\|_\infty. \tag{24}$$

Combining (22), (23) and (24) gives the bound we are seeking. □

Next consider the second term in (21). The following bound holds.

Lemma 6.2. *If the approximate value iteration algorithm terminates after $n+1$ iterations, so that, for all $x \in S$,*

$$\pi(x) \in \operatorname{argmax}_{u \in \Gamma(x)} \left\{ r(x,u) + \varrho \int \hat{T}^{n+1}v(y)\mathbf{M}(x,u;dy) \right\}, \tag{25}$$

then for this n we have the error bound

$$(1-\varrho)\|\hat{T}^{n+1}v - v_\pi\|_\infty \leq \|T\hat{T}^{n+1}v - LT\hat{T}^{n+1}v\|_\infty + \varrho\|\hat{T}^{n+1}v - \hat{T}^{n}v\|_\infty.$$

Proof. Fix $n \in \mathbb{N}$. By the triangle inequality,

$$\|\hat{T}^{n+1}v - v_\pi\|_\infty \leq \|\hat{T}^{n+1}v - T\hat{T}^{n+1}v\|_\infty + \|T\hat{T}^{n+1}v - v_\pi\|_\infty. \tag{26}$$

Consider the second term in the right hand side of (26). From the definition of T_π in (10) and the fact that π solves (25), it is clear that $T\hat{T}^{n+1}v$ and $T_\pi\hat{T}^{n+1}v$ are equal. Moreover, we know that T_π is a contraction of modulus ϱ, and v_π is the unique fixed point. Hence

$$\|T\hat{T}^{n+1}v - v_\pi\|_\infty = \|T_\pi\hat{T}^{n+1}v - T_\pi v_\pi\|_\infty \leq \varrho\|\hat{T}^{n+1}v - v_\pi\|_\infty.$$

Substituting this into (26) we get

$$\|\hat{T}^{n+1}v - v_\pi\|_\infty \leq \|\hat{T}^{n+1}v - T\hat{T}^{n+1}v\|_\infty + \varrho\|\hat{T}^{n+1}v - v_\pi\|_\infty.$$

$$\therefore \quad (1-\varrho)\|\hat{T}^{n+1}v - v_\pi\|_\infty \leq \|\hat{T}^{n+1}v - T\hat{T}^{n+1}v\|_\infty. \tag{27}$$

The right hand side of (27) is a slightly awkward bound to work with in applications, so we split it up as follows:

$$\begin{aligned}\|\hat{T}^{n+1}v - T\hat{T}^{n+1}v\|_\infty &\leq \|\hat{T}^{n+1}v - \hat{T}^{n+2}v\|_\infty + \|\hat{T}^{n+2}v - T\hat{T}^{n+1}v\|_\infty \\ &\leq \varrho\|\hat{T}^{n}v - \hat{T}^{n+1}v\|_\infty + \|LT\hat{T}^{n+1}v - T\hat{T}^{n+1}v\|_\infty.\end{aligned}$$

Substituting this into (27) gives the bound that we are seeking. □

Proof of Theorem 5.1. Pick any $x \in S$. Suppose that the approximate value iteration algorithm terminates after $n+1$ iterations. Substituting the bounds in Lemmas 6.1 and 6.2 into (21) yields

$$\begin{aligned}(1-\varrho)\|v^* - v_\pi\|_\infty \leq \|v^* - Lv^*\|_\infty& \\ + \|T\hat{T}^{n+1}v - LT\hat{T}^{n+1}v\|_\infty + 2\varrho\|\hat{T}^{n+1}v - \hat{T}^{n}v\|_\infty&.\end{aligned}$$

Since $v^* \in \mathcal{V}$ and $T\hat{T}^{n+1}v \in \mathcal{V}$, this reduces to

$$(1-\varrho)\|v^* - v_\pi\|_\infty \leq 2\sup_{w\in\mathcal{V}}\|w - Lw\|_\infty + 2\varrho\|\hat{T}^{n+1}v - \hat{T}^{n}v\|_\infty.$$

By the definition of n and δ we have $\|\hat{T}^{n+1}v - \hat{T}^{n}v\|_\infty \leq \delta$.

$$\therefore \quad (1-\varrho)\|v^* - v_\pi\|_\infty \leq 2\sup_{w\in\mathcal{V}}\|w - Lw\|_\infty + 2\varrho\delta.$$

$$\therefore \quad (1-\varrho)(v^*(x) - v_\pi(x)) \leq 2\sup_{w\in\mathcal{V}}\|w - Lw\|_\infty + 2\varrho\delta.$$

Dividing through by $(1-\varrho)$ gives the bound we are seeking. □

Next we turn to the proof of Theorem 5.2. The proof is based on the following lemma:

Lemma 6.3. *If the approximate value iteration algorithm terminates after $n+1$ iterations, so that, for all $x \in S$,*

$$\pi(x) \in \operatorname{argmax}_{u\in\Gamma(x)}\left\{r(x,u) + \varrho\int \hat{T}^{n+1}v(y)\mathbf{M}(x,u;dy)\right\}, \tag{28}$$

then for this n we have

$$(1-\varrho)\|v^* - v_\pi\|_\infty \leq 2\|\hat{T}^{n+1}v - v^*\|_\infty. \tag{29}$$

Proof. We have

$$\|v^* - v_\pi\|_\infty \leq \|v^* - \hat{T}^{n+1}v\|_\infty + \|\hat{T}^{n+1}v - v_\pi\|_\infty. \tag{30}$$

But

$$\|\hat{T}^{n+1}v - v_\pi\|_\infty \leq \|\hat{T}^{n+1}v - T\hat{T}^{n+1}v\|_\infty + \|T\hat{T}^{n+1}v - v_\pi\|_\infty. \tag{31}$$

Consider the first term on the right hand side of (31). Observe that for any $w \in b\mathcal{B}(S)$ we have

$$\begin{aligned}\|w - Tw\|_\infty &\leq \|w - v^*\|_\infty + \|v^* - Tw\|_\infty \\ &\leq \|w - v^*\|_\infty + \varrho\|v^* - w\|_\infty = (1+\varrho)\|w - v^*\|_\infty.\end{aligned}$$

Substituting in $\hat{T}^{n+1}v$ for w, we obtain

$$\|\hat{T}^{n+1}v - T\hat{T}^{n+1}v\|_\infty \leq (1+\varrho)\|\hat{T}^{n+1}v - v^*\|_\infty. \tag{32}$$

Now consider the second term on the right hand side of (31). It has already been observed that for this particular policy π we have $T\hat{T}^{n+1}v = T_\pi\hat{T}^{n+1}v$, so

$$\begin{aligned}\|T\hat{T}^{n+1}v - v_\pi\|_\infty &= \|T_\pi\hat{T}^{n+1}v - v_\pi\|_\infty \\ &= \|T_\pi\hat{T}^{n+1}v - T_\pi v_\pi\|_\infty \leq \varrho\|\hat{T}^{n+1}v - v_\pi\|_\infty.\end{aligned}$$

Substituting this bound and (32) into (31), we obtain

$$\|\hat{T}^{n+1}v - v_\pi\|_\infty \leq (1+\varrho)\|\hat{T}^{n+1}v - v^*\|_\infty + \varrho\|\hat{T}^{n+1}v - v_\pi\|_\infty.$$

$$\therefore \quad \|\hat{T}^{n+1}v - v_\pi\|_\infty \leq \frac{1+\varrho}{1-\varrho}\|\hat{T}^{n+1}v - v^*\|_\infty.$$

This inequality and (30) together give

$$\|v^* - v_\pi\|_\infty \leq \|v^* - \hat{T}^{n+1}v\|_\infty + \frac{1+\varrho}{1-\varrho}\|\hat{T}^{n+1}v - v^*\|_\infty.$$

Simple algebra now gives (29). □

Proof of Theorem 5.2. Pick any $x \in S$, and suppose that the value iteration algorithm terminates after $n+1$ steps. By Lemma 6.3 we have

$$v^*(x) - v_\pi(x) \leq \frac{2}{1-\varrho}\|\hat{T}^{n+1}v - v^*\|_\infty.$$

Applying Lemma 6.1, this becomes

$$v^*(x) - v_\pi(x) \leq \frac{2}{(1-\varrho)^2}(\varrho\|\hat{T}^{n+1}v - \hat{T}^n v\|_\infty + \|v^* - Lv^*\|_\infty).$$

The claim in Theorem 5.2 now follows from the definition of δ. □

References

[1] Brock, W. A. and L. Mirman (1972): "Optimal Economic Growth and Uncertainty: The Discounted Case," *Journal of Economic Theory,* 4, 479–513.

[2] Den Haan, W.J. and Marcet, A. (1994): "Accuracy in Simulations," *Review of Economic Studies,* 61 (1), 3–17.

[3] Drummond, C (1996): "Preventing Overshoot of Splines with Application to reinforcement Learning," Computer Science Dept. Ottawa TR-96-05.

[4] Gordon, G.J. (1995): "Stable Function Approximation in Dynamic Programming," Proceedings of the 12th International Conference on Machine Learning.

[5] Guestrin, C., D. Koller and R. Parr (2001): "Max-Norm Projections for Factored MDPs," International Joint Conference on Artificial Intelligence, Vol. 1, 673–680.

[6] Hall, R.E. (1978): "Stochastic Implications of the Life Cycle-Permanent Income Hypothesis: Theory and Evidence," *Journal of Political Economy,* 86, 971–987.

[7] Hernández-Lerma, O., and J.B. Lasserre (1999): *Further Topics in Discrete-Time Markov Control Processes,* Springer-Verlag, New York.

[8] Judd, K.L. (1992): "Projection Methods for Solving Aggregate Growth Models," *Journal of Economic Theory,* 58 (2), 410–452.

[9] Judd, K.L. (1998): *Numerical Methods in Economics,* MIT Press, Cambridge, Massachusetts.

[10] Judd, K.L. and A. Solnick (1994): "Numerical Dynamic Programming with Shape-Preserving Splines," unpublished manuscript.

[11] Kydland, F. and E.C. Prescott (1982): "Time to Build and Aggregate Fluctuations," *Econometrica,* 50, 1345–1371.

[12] Lucas, R.E., Jr. and E.C. Prescott (1971): "Investment under Uncertainty," *Econometrica,* 39, 659–681.

[13] Lucas, R.E., Jr. (1978): "Asset Prices in an Exchange Economy," *Econometrica,* 46 (6), 1429–1445.

[14] Lyche, T and K. Mørken (2002): *Spline Methods,* draft manuscript, University of Oslo.

[15] McCall, J.J. (1970): "Economics of Information and Job Search," *The Quarterly Journal of Economics,* 84 (1), 113–26.

[16] Mehra, R. and E.C. Prescott (1985): "The Equity Premium: A Puzzle," *Journal of Monetary Economics,* 15, 145–161.

[17] Munos, R. and A. Moore (1999): "Variable Resolution Discretization in Optimal Control," *Machine Learning,* 1, 1–24.

[18] Munos, R. (2005): "Error Bounds for Approximate Value Iteration," American Conference on Artificial Intelligence.

[19] Puterman, M. (1994): *Markov Decision Processes: Discrete Stochastic Dynamic Programming,* John Wiley & Sons, New York.

[20] Reiter, M. (2001): "Estimating the Accuracy of Numerical Solutions to Dynamic Optimization Problems," manuscript, Universitat Pompeu Fabra.

[21] Rust, J. (1996): "Numerical Dynamic Programming in Economics," in H. Amman, D. Kendrick and J. Rust (eds.) *Handbook of Computational Economics,* Elsevier, North Holland.

[22] Rust, J. (1997): "Using Randomization to Break the Curse of Dimensionality," *Econometrica,* 65 (3), 487–516.

[23] Samuelson, P.A. (1971): "Stochastic Speculative Price," *Proceedings of the National Academy of Science,* 68 (2), 335–337.

[24] Santos, M.S. and J. Vigo-Aguiar (1998): "Analysis of a Numerical Dynamic Programming Algorithm Applied to Economic Models," *Econometrica,* 66(2), 409–426.

[25] Santos, M.S. (2000): "Accuracy of Numerical Solutions Using the Euler Equation Residuals," *Econometrica,* 68 (6), 1377–1402.

[26] Stokey, N. L., R. E. Lucas and E. C. Prescott (1989): *Recursive Methods in Economic Dynamics,* Harvard University Press, Massachusetts.

[27] Tsitsiklis, J.N. and B. Van Roy (1996): "Feature-Based Methods for Large Scale Dynamic Programming," *Machine Learning,* 22, 59–94.

Department of Economics, University of Melbourne, VIC 3010, Australia, *j.stachurski@econ.unimelb.edu.au*