Fitted Value Function Iteration with Probability One Contractions[☆]

Jenő Pál

Department of Economics, Central European University, 1051 Budapest, Nádor u. 9., Hungary

John Stachurski

Research School of Economics, Australian National University, Canberra 0200, Australia

Abstract

This paper studies a value function iteration algorithm based on nonexpansive function approximation and Monte Carlo integration that can be applied to almost all stationary dynamic programming problems. The method can be represented using a randomized fitted Bellman operator and a corresponding algorithm that is shown to be globally convergent with probability one. When additional restrictions are imposed, an $O_P(n^{-1/2})$ rate of convergence for Monte Carlo error is obtained.

Keywords: Dynamic programming, value function iteration, Monte Carlo

1. Introduction

Many economic models contain stochastic dynamic programs (SDPs), either as representations of competitive equilibria, or, more commonly, as sub-problems defining the behavior of firms, households, or other individual agents. When solving these SDPs, computational constraints remain a major bottleneck. The difficulty is particularly acute in settings where the SDP must be solved at a large

[☆]The authors are indebted to Toni Braun, Chiaki Hara, Hidehiko Ichimura, Timothy Kam, Takashi Kamihigashi, Pedro Gomis-Porqueras, Kevin Reffett, Orion Sanders, Juan Pablo Rincón-Zapatero, Yiannis Vailakis, two anonymous referees, editor Paul Klein, and seminar participants at the 10th SAET conference in Singapore and the 17th international CEF conference in San Francisco for many helpful comments. We also thank Corvinus University of Budapest and the Institute of Economic Research at Kyoto University for their contributions to our education and research. This project was supported in part by the Japan Society for the Promotion of Science and Australian Research Council Grant DP120100321.

Email addresses: pal_jeno@ceu-budapest.edu (Jenő Pál), john.stachurski@anu.edu.au (John Stachurski)

number of different parameterizations, either to compute equilibria (as in Bewley models and dynamic games), or to estimate structural econometric models with unknown parameters in the primitives of the SDP.

In recent years, many specialist algorithms have been proposed. These specialist algorithms take advantages of certain features of a given application in order to obtain fast convergence rates. In most of these studies, global (or even local) convergence of the algorithm to the optimum is not proved. Instead, the authors test the algorithm against a special case possessing analytical solutions, or compare their convergence rates against competing algorithms in a specific example. Needless to say, this provides no guarantee of convergence for problems other than the examples treated in the study.

There are good reasons to be concerned about whether common dynamic programming routines do in fact converge to optimal value functions and policies. This is particularly the case in continuous state space settings, where iterative techniques involve some form of function approximation. The interplay between function approximation and dynamic programming routines is known to be relatively delicate. For example, in value function iteration, many function approximation techniques fail to preserve the global contraction properties of the Bellman operator, and several authors have already demonstrated how adding standard function approximation steps can lead to cycles and failure of convergence (Boyan and Moore, 1995; Baird, 1995; Tsitsiklis and Van Roy, 1996).

While specialist algorithms known to converge quickly in particular settings are certainly of value, in this paper our aim is to study a simulation-based value iteration algorithm that has guaranteed convergence properties across a very wide variety of applications. Our set up includes a function approximation step, admits continuous state-action spaces, and makes no use of densities. We provide general conditions under which the fixed point of our random fitted Bellman operator converges uniformly to the value function with probability one. Under additional regularity conditions, we show that the supremum norm deviation is $O_P(n^{-1/2})$.

Our techniques provide a natural alternative to discretized value function iteration, a method which also has very broad applicability, and remains a popular benchmark in economic applications. In discretized value function iteration, a continuous state/action problem is replaced by a "nearby" discrete problem. Relative to the method we study here, discretization has several disadvantages. First, while the discretized algorithm always locates the solution to the *discrete* problem, the deviation between this discrete solution and the solution to the original problem is not easily obtained. To the best of our knowledge, no global con-

vergence results are available in a setting as general as the one that we treat here.[1] Second, in terms of finite time properties, discrete representation of continuous curves is costly relative to continuous parametric representations and inherently subject to the curse of dimensionality.[2]

1.1. Methodology

Successful fitted value function iteration in a continuous state setting requires careful choice of both function approximation scheme and of numerical integration method. In the numerical dynamic programming literature, most authors use standard approximation theory to guide their choices. For example, Chebyshev polynomials are popular for function approximation because classical approximation theory shows they have desirable properties in terms of fitting a certain class of functions. However, the final objective with numerical dynamic program is to produce an estimate of the value or policy function, and to this end the criteria for "good" approximation methods should include not only their approximation properties at each step, but also their interplay with the overall dynamic programming algorithm.

As discussed above, failure to consider this interplay may result in iterative techniques that break the global contraction property of the theoretical Bellman operator, causing instability. Without contractiveness, small errors can be compounded at each iteration, and analysis of the algorithm is highly problematic. For this reason, we focus on approximation methods that preserve contractiveness. On the function approximation side, we use nonexpansive approximation operators, as pioneered by Judd and Solnick (1994) and Gordon (1995).[3] On the numerical integration side we use Monte Carlo. We prove below that this combination preserves contractiveness with probability one. (The "probability one" statement means that the approximate Bellman operator is a contraction for every

[1] One issue is that discretization errors for continuous curves tend to be bounded in terms of derivatives, which fail to exist or cannot be bounded in many economic settings. Optimal growth models often have unbounded derivatives as a result of Inada conditions. Derivatives can fail to exist when models include discrete choices, binding constraints, non-convexities and so on.

[2] The number of data points needed to represent function in $\mathbb{R}^d$ parametrically may be polynomial in d, while discrete representations are always exponential. The intractability of discrete representations in moderate to high dimensions has led practitioners in fields such as engineering and computer science to *reverse* the discretization process, replacing inherently discrete dynamic programs with continuous ones. This idea dates back to Bellman (Bellman et al., 1963).

[3] The focus in Judd and Solnick (1994) is on shape-preserving approximations. It turns out that some of these approximation methods are nonexpansive (see, for example, the proof of Theorem 4). Gordon (1995) is concerned exclusively with nonexpansive approximation. Other closely related studies in terms of approximation methods include Drummond (1996), Guestrin et al. (2001), Santos and Vigo-Aguiar (1998) and Stachurski (2008).

realization of the Monte Carlo sample.) The contractiveness of the approximate Bellman operator is central to all of our error analysis.

Regarding the use of Monte Carlo for numerical integration, a potential disadvantage is that Monte Carlo integration is often less efficient that deterministic routines in low dimensional settings with smooth functions. On the other hand, Monte Carlo methods tend to perform well in higher dimensional settings, or when the target function is less smooth. In the present setting, there is an additional efficiency: The Monte Carlo sample is drawn once-off to form the approximate Bellman operator, and re-used many times to evaluate expectations.

Regarding the function approximation step, the idea behind nonexpansive approximation operators is straightforward: Let A be an operator, mapping real-valued function w into another real-valued function Aw on the same domain. The image Aw is interpreted as an approximation of w. The operator A is called nonexpansive if, for any two functions v and w, the images Av and Aw are no further apart than the preimages v and w. (Here distance is measured in the supremum norm.)

Examples of nonexpansive approximation schemes include continuous piecewise linear interpolation, k-nearest neighbors, Schoenberg variation diminishing splines and the class of kernel smoothers. With kernel smoothers, the approximation Aw of w has the form

$$(Aw)(x) = \sum_{i=1}^{m} w(x_i)\psi\left\{\frac{\|x - x_i\|}{h}\right\}\eta(x). \tag{1}$$

The scalar h is a smoothing parameter, $\{x_i\}_{i=1}^m$ is a set of grid points, $\psi\colon \mathbb{R}_+ \to \mathbb{R}_+$ is continuous and monotone decreasing, and $\eta(x)$ is a normalization term defined as $\eta(x) := 1/[\sum_{j=1}^m \psi(\|x - x_j\|/h)]$. Essentially, $(Aw)(x)$ is a weighted average of the function values $\{w(x_i)\}$ on the grid, with higher weight for grid points close to x. A common choice for ψ, particularly in higher dimensions, is $\psi(t) = \exp(-t^2)$. In this case A is called a (Gaussian) radial basis function kernel smoother. Nonexpansiveness of A is shown in Gordon (1995, Theorem 3.2).

An illustration of nonexpansiveness using a radial basis kernel is given in Figure 1. The figure shows two functions w and v as dashed lines, and their approximations Aw and Av as unbroken lines. (The approximation is deliberately coarse, so that the difference between the functions and their approximations is clearly visible.) It can be seen from the figure that the sup norm (i.e., maximal) deviation between the approximations is no larger than the maximal deviation between the original functions. For comparison, see Figure 2, where the same functions are approximated with Chebyshev polynomials. In this case the approximating polynomial overshoots the higher function in the middle of the interval, and the

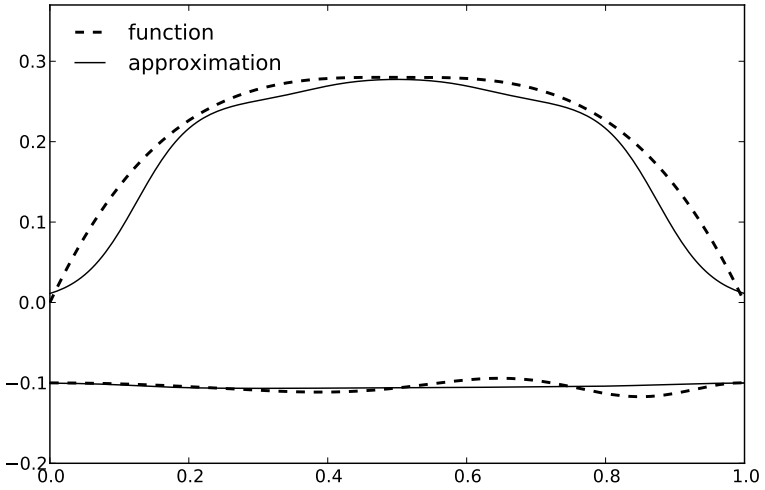

Figure 1: Nonexpansive approximation with a radial basis kernel smoother

maximal deviation (occurring near 0.5 on the x-axis) is increased by approximation.[4]

Figure 3 shows an example of instability in fitted value function iteration, when Chebyshev polynomials are used for approximation in the standard optimal growth model with log utility and Cobb-Douglas production $f(k) = k^\alpha$. (See Example 6.1 for a description of the model.) Here the discount factor is 0.95, $\alpha = 0.33$ and the state space is from 10^{-5} to 1. The shock z is multiplicative and lognormal, with parameters $\mu = 0$ and $\sigma = 0.25$. The initial condition for the iteration is the utility function. The Chebyshev polynomials are of order 10 with 150 nodes, and integration is Gaussian quadrature of order 5. In the plot, the true value function is the dashed line. The iterates of the approximate Bellman operator diverge downwards (successive iterates are plotted in darker grey) and away from the true value function.[5]

[4]Both approximations use 5 grid points. The kernel smoother uses the radial kernel $\psi(t) = \exp(-t^2)$ with $h = 8$. The Chebyshev polynomial approximation uses a third order polynomial.

[5]The growth model is the one presented in Section 6.3. For the distance between iterates see the last column of Table 1 (Section 7). In the exercise, computation of Chebyshev polynomials was standard (see, e.g., Judd, 1998, Chapter 6). We repeated the experiment with the extended Chebyshev array method and obtained similar results. In Section 7 below, we re-run the same experiment, but this time using various approximation methods that satisfy our conditions. In all cases, the sequence of iterates converge (all but last column of Table 1). The code for all our experiments can be found at `https://sites.google.com/site/fviprobone/`.

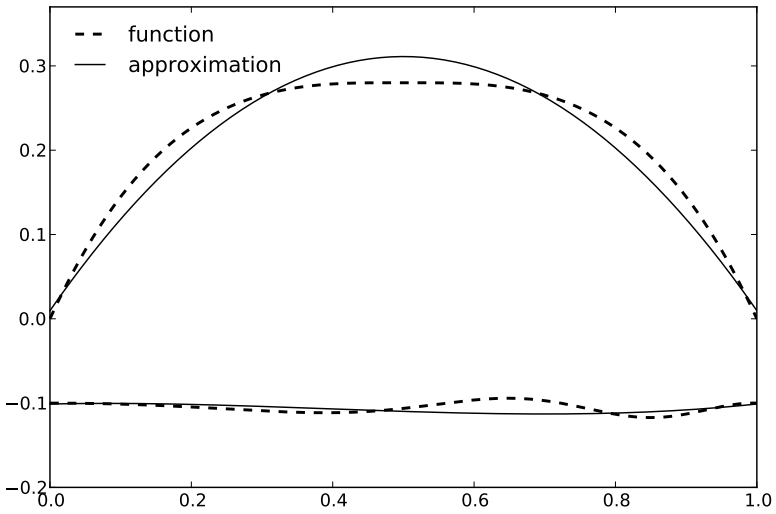

Figure 2: Expansive approximation with Chebyshev polynomials

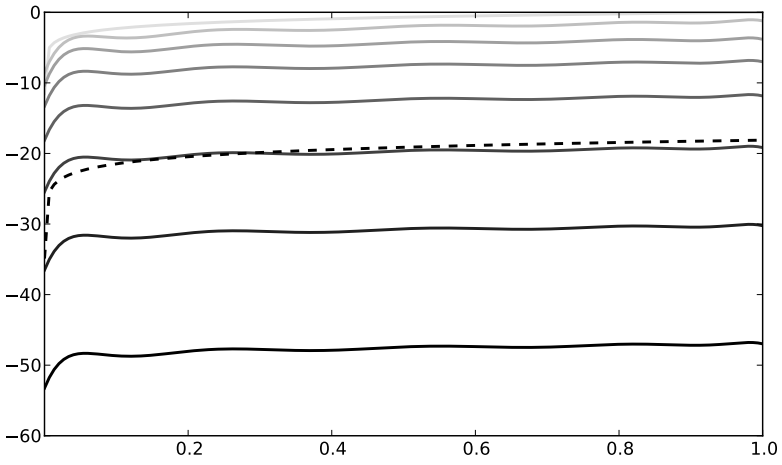

Figure 3: Instability in fitted VFI with Chebyshev polynomials

In this example, non-convergence is relatively extreme. (After all, failure of convergence does not necessarily entail divergence. For example, initial convergence followed by cycling is a less extreme failure of convergence that may be more likely in practice.) However, the results are informative because they suggest that convergence results for this particular class of polynomials cannot be obtained without restrictions that may not hold even for very basic economic models. (Judd and Solnick (1994) also mention finding examples where "wild divergence" occurs while conducting fitted value iteration with polynomials, although details are not given in the paper.)

Finally, it should also be added that the probability one global stability that is guaranteed in our algorithm is not necessarily important in and of itself, since a stable algorithm can in principle converge to an object that has no relationship to the object one wishes to compute. Rather, stability is a first step that allows us to pursue a general consistency result.

1.2. Related Literature

The emphasis in the paper is on global convergence in a general setting. Numerical methods for more specific models with additional structure can be found in many papers. The literature is too large too survey here. Useful introductions can be found in Marimon and Scott (2001) or Aruoba et al. (2006). Both of these references include discussion of methods for solving smooth SDPs where optimal policies satisfy Euler equations. (We assume no such smoothness here.) An iterative method for concave SDPs is analyzed in the recent paper of Fukushima and Waki (2011). Some well-known algorithms based on Monte Carlo include those found in Keane and Wolpin (1994), Rust (1997), Pakes and McGuire (2001) and Longstaff and Schwartz (2001).

Several authors have published studies on finite-time bounds for fitted value function iteration. For example, Rust (1997) proposed an ingenious function approximation step that can be implemented when the one-step transition probabilities for the dynamic programming problem are absolutely continuous with respect to Lebesgue measure (i.e., the distribution for the next period state given current state and action can be represented by a density). He proved that for decision problems satisfying certain Lipschitz conditions, his algorithm breaks the curse of dimensionality, in the sense that worst-case computational complexity is polynomial in the dimension of the state space. Further important developments for models satisfying similar restrictions were reported in Munos and Szepesvári (2008). Finally, Stachurski (2008) provided finite time bounds for fitted value iteration, although the problem of numerical integration was not considered.

The contribution of this paper is different. The generality of our setting precludes us from developing tight finite-time bounds. Instead, our focus is on

obtaining consistency under very weak assumptions. For example, in our consistency result, we make no smoothness or differentiability assumptions. This allows us to include models with discrete choices, occasionally binding constraints, non-convexities and so on. In addition, we do not assume that the one-step transition probabilities are absolutely continuous, an assumption that was central to Rust's (1997) algorithm. This additional generality is important in economics, since many applications have one-step transition probabilities that fail to be absolutely continuous. To give an example, consider a benchmark macroeconomic model, where next period capital stock is given by

$$k_{t+1} = (1 - \delta)k_t + f(k_t, z_t) - c_t.$$

Here δ is a depreciation rate, f is a production function, c_t is consumption and $\{z_t\}$ is an exogenous shock process, typically Markovian. Observe that as soon as the current state (k_t, z_t) and the current action c_t is given, next period capital is deterministic. As a result, the one-step transition probability fails to be absolutely continuous, and cannot be represented by a density.

In this example, the problem is caused by stochastic rank deficiency—the shock space has lower dimension than the state space. While the example is simplistic, it is also representative of the growth and macroeconomic literature—see for example the standard formulation of Stokey and Lucas (1989, p. 290)—and illustrates the fact that many models in these fields cannot be treated with density-based approaches unless modifications are imposed.

In addition to stochastic rank deficiency, failure of absolute continuity can be caused by discrete shocks (e.g., labor productivity shocks following discrete Markov chains), constraints and other common features. Representative dynamic programming problems where the transition probability fails to be absolutely continuous include those found in Kydland and Prescott (1982, p. 1354), Aiyagari (1994), Huggett (1997) and Clementi and Hopenhayn (2006).

1.3. Outline

Section 2 of the paper provides background concepts and notation. Section 3 defines the model, and Section 4 introduces the algorithm. Section 5 provides convergence results, Section 6 discusses rates of convergence, Section 7 gives applications, and Section 8 concludes. Proofs can be found in Section 9.

2. Preliminaries

We begin by introducing notation. For topological space $\mathbb{T}$, the symbol $\mathscr{C}(\mathbb{T})$ denotes the collection of continuous, bounded, real-valued functions on $\mathbb{T}$, while

$\|\cdot\|$ is the supremum norm on $\mathscr{C}(\mathbb{T})$. Operator $S\colon \mathscr{C}(\mathbb{T}) \to \mathscr{C}(\mathbb{T})$ is called a contraction of modulus ρ if $0 \leq \rho < 1$ and

$$\|Sv - Sw\| \leq \rho\|v - w\| \text{ for all pairs } v, w \in \mathscr{C}(\mathbb{T}). \tag{2}$$

S is called nonexpansive if (2) holds with $\rho = 1$. By Banach's contraction mapping theorem, every contraction S of modulus ρ on $\mathscr{C}(\mathbb{T})$ has a unique fixed point $W \in \mathscr{C}(\mathbb{T})$, and, moreover, $\|S^n w - W\| = O(\rho^n)$ for each $w \in \mathscr{C}(\mathbb{T})$.

Lemma 2.1. *Let S and S' be operators from $\mathscr{C}(\mathbb{T})$ to itself.*

1. *If S is nonexpansive and S' is a contraction of modulus ρ, then the composition $S \circ S'$ is a contraction of modulus ρ.*
2. *If S and S' are both contractions of modulus ρ with fixed points W and W' respectively, then $\|W - W'\| \leq (1-\rho)^{-1}\|SW' - W'\|$.*

Part 1 is trivial. For a proof of part 2, see, for example, Rust (1997, Lemma 2.1).

In what follows, all random variables are defined on a common probability space $(\Omega, \mathscr{F}, \mathbf{P})$, and $\mathbf{E}$ is the expectation with respect to $\mathbf{P}$. If X is a map from Ω into $\mathbb{R}$ that is not necessarily measurable, then the outer expectation of X is $\mathbf{E}^* X := \inf_Y \mathbf{E}Y$, where the infimum is over all real random variables Y such that $X \leq Y$ and $\mathbf{E}Y$ exists (cf., e.g., van der Vaart, 1998, p. 258). For a sequence of possibly nonmeasurable maps $\{U_n\}$ from Ω into a metric space $(\mathbb{T}, d)$ and a $\mathbb{T}$-valued random variable U, we say that $U_n \to U$ holds $\mathbf{P}^*$-almost surely if there exists a measurable real-valued sequence Δ_n with $d(U_n, U) \leq \Delta_n$ and $\mathbf{P}\{\Delta_n \to 0\} = 1$. We say that U_n converges in distribution to U if $\mathbf{E}^* g(U_n) \to \mathbf{E}g(U)$ for every $g \in \mathscr{C}(\mathbb{T})$. For the former convergence we write $U_n \overset{a.s.^*}{\to} U$, while for the latter we write $U_n \overset{d^*}{\to} U$.

The continuous mapping theorem continues to hold in this setting:

Lemma 2.2. *If $\mathbb{T}'$ is another metric space and $g\colon \mathbb{T} \to \mathbb{T}'$ is continuous, then*

$$U_n \overset{d^*}{\to} U \implies g(U_n) \overset{d^*}{\to} g(U).$$

Let $\{X_n\}_{n\geq 1}$ be a sequence of (not necessarily measurable) maps from Ω into $\mathbb{R}$. We write $X_n = O_{P^*}(n^{-1/2})$ if there exists a sequence of real-valued random variables $\{\Delta_n\}_{n\geq 1}$ such that $|X_n| \leq \Delta_n$ for all n and $\Delta_n = O_P(n^{-1/2})$.

3. Set Up

In this section we introduce a general stochastic dynamic programming problem and describe the value function iteration algorithm.

3.1. The Model

Consider an SDP of the following form. A controller observes the state $x \in \mathbb{X}$ of a given system, and responds with an action a from a feasible set $\Gamma(x) \subset \mathbb{A}$. Given this state-action pair (x, a), the controller receives current reward $r(x, a)$, and the new state is determined as $x' = F(x, a, U)$, where U is a draw from a fixed distribution ϕ. The process now repeats. The controller's objective is to maximize the sum of expected discounted rewards given discount factor β.

The sets $\mathbb{X}$ and $\mathbb{A}$ are referred to as the state and action spaces respectively, and Γ is called the feasible correspondence. We let

$$\mathbb{G} := \operatorname{graph} \Gamma := \{(x, a) \in \mathbb{X} \times \mathbb{A} : a \in \Gamma(x)\}.$$

The set $\mathbb{G}$ is called the set of feasible state-action pairs.

A feasible policy is a Borel measurable map $\sigma \colon \mathbb{X} \to \mathbb{A}$ such that $\sigma(x) \in \Gamma(x)$ for all $x \in \mathbb{X}$. Let Σ be the set of all such policies. The controller's problem is

$$\max_{\sigma \in \Sigma} \left\{ \mathbf{E} \sum_{t=0}^{\infty} \beta^t r(X_t, \sigma(X_t)) \right\}$$

subject to

$$X_{t+1} = F(X_t, \sigma(X_t), U_{t+1}) \quad \text{with } x_0 \text{ given.} \tag{3}$$

Almost any stationary infinite horizon dynamic program with additively separable preferences can be formulated in this way. We assume throughout the paper that

1. $\mathbb{X}$ and $\mathbb{A}$ are compact metric spaces.
2. Γ is continuous and compact-valued.
3. The shocks $\{U_t\}_{t \geq 1}$ are IID with common distribution ϕ.[6]
4. ϕ is a Borel probability measure over metric space $\mathbb{U}$.
5. The reward function $r \colon \mathbb{G} \to \mathbb{R}$ is continuous.

[6]The assumption of IID shocks is not restrictive. For example, consider the following macroeconomic model with exogenous Markov shock sequence: The state space is a product space $K \times Z \subset \mathbb{R}^m \times \mathbb{R}^n$, where $k \in K$ is a vector of endogenous variables and $z \in Z$ is a vector of exogenous variables. Technology is summarized by a feasible set $\Theta \subset K \times Z \times K$. The exogenous process $\{z_t\}_{t \geq 0}$ evolves according to $z_{t+1} = g(z_t, \epsilon_{t+1})$, where $\{\epsilon_t\}_{t \geq 1}$ is IID. Instantaneous rewards are given by $v(k, z, k')$. This formulation is a special case of our SDP. To see this, for the state take $x := (k, z) \in K \times Z$, and for the action take $a := k' \in K$. The feasible correspondence is $\Gamma(x) := \Gamma(k, z) := \{k' \in K : (k, z, k') \in \Theta\}$. The shock is $u := \epsilon$, and the transition function is $F(x, a, u) := F(k, z, k', \epsilon) := (k', g(z, \epsilon)) \in K \times Z$. The reward function is $r(x, a) := r(k, z, k') := v(k, z, k')$.

6. The function $\mathbb{G} \ni (x, a) \mapsto F(x, a, u) \in \mathbb{X}$ is continuous for all $u \in \mathbb{U}$.

For $\{X_t\}$ as given by (3), let $V_\sigma(x_0) = \mathbf{E} \sum_{t=0}^{\infty} \beta^t r(X_t, \sigma(X_t))$. Let $T\colon \mathscr{C}(\mathbb{X}) \to \mathscr{C}(\mathbb{X})$ be the Bellman operator, defined at $v \in \mathscr{C}(\mathbb{X})$ by

$$Tv(x) := \max_{a \in \Gamma(x)} \left\{ r(x, a) + \beta \int v[F(x, a, u)] \phi(du) \right\} \qquad (x \in \mathbb{X}). \tag{4}$$

For $v \in \mathscr{C}(\mathbb{X})$, a policy $\sigma \in \Sigma$ is called v-greedy if $\sigma(x)$ is a maximizer of the right-hand size of (4) for all $x \in \mathbb{X}$. The value function V_T is defined pointwise on $\mathbb{X}$ by $V_T(x) = \sup_{\sigma \in \Sigma} V_\sigma(x)$. A policy $\sigma \in \Sigma$ is called optimal if $V_\sigma = V_T$.

3.2. Value Function Iteration

The following result is standard:

Theorem 3.1. *Under Assumptions 1–6 above,*

1. *T is a contraction of modulus β on $\mathscr{C}(\mathbb{X})$, and V_T is the unique fixed point;*
2. *a policy $\sigma \in \Sigma$ is optimal if and only if it is V_T-greedy; and*
3. *at least one such policy exists.*

In principle, V_T can be calculated by value function iteration (VFI), which involves fixing an initial $v \in \mathscr{C}(\mathbb{X})$ and iterating with T. From Theorem 3.1 we have $\|T^k v - V_T\| = O(\beta^k)$. Using this fact and optimality of V_T-greedy policies, one can show that a $T^k v$-greedy policy is approximately optimal when k is sufficiently large.[7]

4. Random Fitted VFI

Evaluation of the expression $r(x, a) + \beta \int v[F(x, a, u)] \phi(du)$ on the right-hand side of (4) requires approximation of the integral. To do so we use Monte Carlo, which allows us to preserve the contractiveness of the Bellman operator, as discussed below. Another advantage of using Monte Carlo in our set up is that we will be able to evaluate every integral by drawing a single sample

$$U_1, \ldots, U_n \overset{\text{IID}}{\sim} \phi \tag{5}$$

once off. Given this sample, we then iterate with the random Bellman operator R_n defined by

$$R_n v(x) := \max_{a \in \Gamma(x)} \left\{ r(x, a) + \beta \frac{1}{n} \sum_{i=1}^{n} v[F(x, a, U_i)] \right\} \qquad (x \in \mathbb{X}). \tag{6}$$

[7] See, e.g., Puterman (1994, Theorem 6.3.1). An appropriate k is usually chosen according to some stopping criterion that depends on the deviation between successive iterates of T.

A realization of $\omega \in \Omega$ determines a particular realization $\{U_i(\omega)\}_{i=1}^n$ of the sample (5), which in turn defines a realization $R_n(\omega)$ of R_n. Each realization $R_n(\omega)$ is an operator from $\mathscr{C}(\mathbb{X})$ to itself.

A second numerical issue is as follows: If $\mathbb{X}$ is infinite, then, for arbitrary given $w \in \mathscr{C}(\mathbb{X})$, one cannot evaluate either $Tw(x)$ or $R_n w(x)$ at each $x \in \mathbb{X}$ in finite time (or store the functions in a look-up table). Hence, we approximate $R_n w$ using a finite parametric representation. To do so, we introduce an approximation operator $A\colon \mathscr{C}(\mathbb{X}) \to \mathscr{A}(\mathbb{X}) \subset \mathscr{C}(\mathbb{X})$, where, given function $w \in \mathscr{C}(\mathbb{X})$, Aw is an approximation of w, and $\mathscr{A}(\mathbb{X})$ is a class of functions such that each element can be represented by a finite number of parameters. In addition, we assume that Aw can be computed on the basis of a finite number of observations (i.e., by observing the value of $w(x)$ at a finite number of $x \in \mathbb{X}$). For example, the mapping $w \mapsto Aw$ might proceed by evaluating w on a fixed and finite grid of points $\{x_i\}_{i=1}^m$, and then constructing Aw based on these "interpolation" points. Finally, we assume throughout that A is nonexpansive.

Example 4.1. Continuous piecewise linear interpolation in $\mathbb{R}^d$ is a nonexpansive approximation scheme.[8]

Other nonexpansive schemes include kernel smoothers (see Section 1.1), k-nearest neighbors, shape-preserving Schumaker splines, and the variation diminishing splines of Schoenberg.

The complete procedure for random fitted value function iteration is given in Algorithm 1. In step 3, $(AR_n)^k$ is the k-th composition of the operator $AR_n := A \circ R_n$ with itself. In practice, when applying the operator AR_n to a given function w, first $R_n w$ is evaluated on a finite grid of points $\{x_i\}_{i=1}^m$ by solving the maximization problem in (6) at each x_i. A is then applied to produce the fitted function $AR_n w$.

5. Analysis

We begin our analysis with the following lemma, the proof of which can be found in Section 9.

[8]To describe it formally, let $\mathbb{X}$ be a convex subset of $\mathbb{R}^d$, let $\mathbb{V}$ be a finite subset of $\mathbb{X}$ such that the convex hull of $\mathbb{V}$ is $\mathbb{X}$, and let T be a $\mathbb{V}$-triangularization of $\mathbb{X}$. (That is, T is a finite collection of non-degenerate simplexes such that the vertices of each simplex lie in $\mathbb{V}$ and any two simplexes intersect on a common face or not at all.) Given a simplex $\Delta \in T$ with vertices $\zeta_1, \ldots, \zeta_{d+1}$, each $x \in \Delta$ can be represented uniquely as $\sum_{i=1}^{d+1} \lambda(x,i)\zeta_i$, where $\lambda(x,i)$ is its i-th barycentric coordinate relative to Δ. (By definition, $\lambda(x,i) \geq 0$ and $\sum_{i=1}^{d+1} \lambda(x,i) = 1$.) For $v \in \mathscr{C}(\mathbb{X})$, we define A by $Av(x) = \sum_{i=1}^{d+1} \lambda(x,i)v(\zeta_i)$. The operator A is nonexpansive (see, e.g., Stachurski, 2008).

Algorithm 1: Random Fitted VFI

generate the sample $\{U_1, \ldots, U_n\} \overset{\text{IID}}{\sim} \phi$ in (5) ;
fix $v \in \mathscr{C}(\mathbb{X})$;
compute $v_k := (AR_n)^k v$ by starting at v and iterating with AR_n k times;
compute a v_k-greedy policy σ ;

Lemma 5.1. *The operator AT is a contraction on $\mathscr{C}(\mathbb{X})$ of modulus β. The operator $AR_n(\omega)$ is also a contraction on $\mathscr{C}(\mathbb{X})$ of modulus β for all $n \in \mathbb{N}$ and all $\omega \in \Omega$.*

As a consequence, there exists

1. a unique fixed point $V_{AT} \in \mathscr{C}(\mathbb{X})$ of AT
2. a unique fixed point $V_{AR_n(\omega)} \in \mathscr{C}(\mathbb{X})$ of $AR_n(\omega)$ for each $\omega \in \Omega$

The operator AT is the fitted Bellman operator where function approximation is included, but the integral is computed exactly. Its fixed point V_{AT} is deterministic. On the other hand, V_{AR_n} is random. In what follows, we refer to V_{AR_n} as a random function, although $\omega \mapsto V_{AR_n(\omega)}$ may not be Borel measurable as a mapping from Ω to $\mathscr{C}(\mathbb{X})$.

Our primary goal is to study the convergence of V_{AR_n} to the value function V_T.[9] By the triangle inequality, the error can be decomposed as

$$\|V_T - V_{AR_n}\| \leq \|V_T - V_{AT}\| + \|V_{AT} - V_{AR_n}\| \qquad \forall\, n \in \mathbb{N}. \tag{7}$$

Let us consider the two terms on the right-hand side of (7). The first term is the function approximation error, caused by replacing T with AT. The second is the integral approximation error, caused by replacing AT with AR_n. We now consider these two errors in turn.

Consider first the function approximation error $\|V_T - V_{AT}\|$. Given a suitable approximation scheme, establishing convergence of the error to zero is relatively straightforward. Rates of convergence will depend on the particular function approximation scheme used in a given implementation, but sufficiently "fine" approximations will make the error arbitrarily small, provided that the range space of A becomes sufficiently rich. To justify this claim, consider the kernel smoother A in (1). The next result shows that the function approximation error can be made arbitrarily small for an operator A in this class without additional assumptions.

[9]The relative optimality of the $(AR_n)^k v$-greedy policy σ computed by Algorithm 1 depends on the deviation between $(AR_n)^k v$ and V_T. Using the triangle inequality, we can bound the latter by $\|(AR_n)^k v - V_{AR_n}\| + \|V_{AR_n} - V_T\|$. By Lemma 5.1, the first term is $O(\beta^k)$ in k. Convergence of V_{AR_n} to V_T is less clear, and hence we focus on this term.

Lemma 5.2. *For any $\epsilon > 0$, there exists a choice of $\{x_i\}_{i=1}^m$, ψ and h such that the corresponding operator A in* (1) *satisfies $\|V_T - V_{AT}\| < \epsilon$.*

Now we turn our attention to the integral approximation error, which is the second term on the right-hand side of (7). Our first major result for the paper shows probability one convergence without any additional assumptions.[10]

Theorem 5.1. $\|V_{AT} - V_{AR_n}\| \overset{a.s.^*}{\to} 0$ *as* $n \to \infty$.

Regarding the proof, recall that a class $\mathscr{H}$ of bounded measurable functions mapping $\mathbb{U}$ into $\mathbb{R}$ is called ϕ-Glivenko-Cantelli if the strong law of large numbers holds uniformly over $\mathscr{H}$, in the sense that if $\{U_i\}$ is an IID sample from ϕ, then

$$\lim_{n\to\infty} \sup_{h\in\mathscr{H}} \left| \frac{1}{n}\sum_{i=1}^{n} h(U_i) - \int h \, d\phi \right| = 0 \quad \mathbf{P}^*\text{-almost surely} \tag{8}$$

(cf., e.g., van der Vaart, 1998). In the proof of Theorem 5.1, a sequence of relatively straightforward manipulations shows that if we set

$$\mathscr{H} := \{\mathbb{U} \ni u \mapsto V_{AT}[F(x,a,u)] \in \mathbb{R} : (x,a) \in \mathbb{G}\}, \tag{9}$$

then the error $\|V_{AT} - V_{AR_n}\|$ is bounded above by a constant times the supremum on the left-hand side of (8). That $\mathscr{H}$ has the Glivenko-Cantelli property is then verified using a well-known sufficient condition. Details are in Section 9.

6. Rates of Convergence

The result in Theorem 5.1 gives no indication as to the rate of convergence. To obtain a rate, we need to give a rate for the convergence in (8). The ϕ-Glivenko-Cantelli property used in the proof of Theorem 5.1 is not sufficient for rates, so further restrictions on the class $\mathscr{H}$ are required.

6.1. Donsker Classes

Let $\mathscr{G}$ be a class of uniformly bounded, measurable functions from $\mathbb{U}$ into $\mathbb{R}$, and let $(b\mathscr{G}, \|\cdot\|)$ be the Banach space of bounded, real-valued functionals on $\mathscr{G}$ with the supremum norm. The class $\mathscr{G}$ is called ϕ-Donsker if

$$\nu_n(g) := \sqrt{n}\left\{ \frac{1}{n}\sum_{i=1}^{n} g(U_i) - \int g \, d\phi \right\}$$

[10]Since Borel measurability of $\omega \mapsto V_{AR_n(\omega)}$ is inherently problematic, the theorem uses the concept of $\mathbf{P}^*$-almost sure convergence.

converges in distribution to a tight Gaussian process ν in the space $b\mathscr{G}$ (cf., e.g., van der Vaart, 1998, p. 269). Here $\omega \mapsto \nu_n(\cdot)(\omega)$ and $\omega \mapsto \nu(\cdot)(\omega)$ are maps from Ω into $b\mathscr{G}$. The maps $\omega \mapsto \nu_n(\cdot)(\omega)$ are not necessarily measurable, and convergence in distribution is to be understood in the sense of $\nu_n \xrightarrow{d^*} \nu$.

Proposition 6.1. *If $\mathscr{H}$ defined in* (9) *is ϕ-Donsker, then $\|V_{AR_n} - V_{AT}\| = O_{P^*}(n^{-1/2})$.*

In essence, Proposition 6.1 tells us that the $\sqrt{n}$ rate can be obtained if the SDP has enough structure for the function class (9) to have the ϕ-Donsker property. There are several well-known sets of sufficient conditions for a function class to be ϕ-Donsker. Below, we use two of these sets to obtain rates of convergence in important but relatively specialized settings.

6.2. The Lipschitz Case

Our first result is based on a Lipschitz condition. To apply the method, we add the following assumptions:

(i) $\mathbb{G} \subset \mathbb{R}^d$.

(ii) Aw is Lipschitz continuous for every $w \in \mathscr{C}(\mathbb{X})$.[11]

(iii) There exists a measurable function $m_0 \colon \mathbb{U} \to \mathbb{R}$ with $\int m_0^2 \, d\phi < \infty$ and[12]

$$\|F(y,u) - F(y',u)\|_2 \leq m_0(u)\|y - y'\|_2 \qquad \forall\, y, y' \in \mathbb{G},\ u \in \mathbb{U}. \tag{10}$$

Notice that the assumptions concern only the transition function, not the reward function. Many dynamic macroeconomic models have Lipschitz transition rules. The consumer's problem in the incomplete markets models of Aiyagari (1994) and Huggett (1997) are obvious examples, and many recent variations have a similar structure (see, e.g., Pijoan-Mas, 2006, or Ábrahám and Cárceles-Poveda, 2010).

Proposition 6.2. *If* (i)–(ii) *hold, then $\|V_{AR_n} - V_{AT}\| = O_{P^*}(n^{-1/2})$.*

An important special case of our Lipschitz assumption is models with linear transition rules. The next lemma provides details.

Lemma 6.1. *If $\mathbb{U} \subset \mathbb{R}^k$, and F is linear, in the sense that*

$$F(x,a,u) = Ax + Ba + Cu \qquad (x \in \mathbb{X},\ a \in \Gamma(x),\ u \in \mathbb{U}) \tag{11}$$

for matrices A, B and C, then Assumption (ii) is satisfied.

[11] This condition depends on the approximation architecture used in the fitted VFI routine, and is satisfied by, for example, the piecewise linear interpolation operator in Example 4.1.

[12] Here $\|\cdot\|_2$ represents the Euclidean norm on $\mathbb{R}^d$.

6.3. The Monotone Case

Another way to establish the ϕ-Donsker property is via monotonicity. To this end, we drop the Lipschitz assumptions of Section 6.2 and replace them with the following:

(i) $\mathbb{X} \subset \mathbb{R}^d$ and $\mathbb{U} \subset \mathbb{R}$.

(ii) A maps $i\mathscr{C}(\mathbb{X})$ to itself, where $i\mathscr{C}(\mathbb{X})$ is the increasing functions in $\mathscr{C}(\mathbb{X})$.

(iii) For all $x, x' \in \mathbb{X}$ with $x \leq x'$, $\Gamma(x) \subset \Gamma(x')$; $r(x,a) \leq r(x',a)$ for all $a \in \Gamma(x)$; and $F(x,a,u) \leq F(x',a,u)$ for all $a \in \Gamma(x)$ and $u \in \mathbb{U}$.

(iv) For all $y \in \mathbb{G}$, $F(y,u) \leq F(y,u')$ whenever $u \leq u'$.

Assumption (ii) depends on the approximation architecture, and is satisfied by, for example, the linear interpolation operator in Example 4.1. The other assumptions are satisfied by a number of standard models.

Proposition 6.3. *If* (i)–(iv) *hold, then* $\|V_{AR_n} - V_{AT}\| = O_{P^*}(n^{-1/2})$.

Example 6.1. Consider the growth model

$$\max \mathbb{E} \sum_{t=0}^{\infty} \beta^t w(c_t) \quad \text{subject to}$$
$$c_t \geq 0, \quad k_{t+1} \geq 0, \quad c_t + k_{t+1} \leq z_t f(k_t).$$

Suppose that z_t is Markov, following transition rule $z_{t+1} = g(z_t, U_{t+1})$, where $(U_t)_{t \geq 1}$ is IID. The state is $(k,z) \in \mathbb{R}^2_+$. To write the model in our framework, we take $F(k,z,k',u) = (k', g(z,u))$, $r(k,z,k') = w(zf(k) - k')$ and $\Gamma(k,z) = [0, zf(k)]$. If f and g are both increasing, then Assumptions (iii) and (iv) above are satisfied.

7. Applications

In this section we consider two numerical applications. In the first application, we revisit the instability illustrated in Figure 3, and re-run the experiment using Algorithm 1 and a variety of approximation operators that conform to our assumptions. As in Figure 3, we use a well-known special case of the growth model in Example 6.1, with $w(c) = \ln c$ and $f(k) = k^\alpha$. The shock is IID and lognormal, with parameters $\mu = 0$ and $\sigma = 0.25$. The remaining parameters are $\beta = 0.95$ and $\alpha = 0.33$. The natural state space $(0, \infty)$ is truncated to the interval $\mathbb{X} = [10^{-5}, 1]$. In performing value function iteration, we use 150 grid points. All of these settings are identical to Figure 3.

Regarding the details of the algorithm, since the shock is IID, we can reduce the two-dimensional state space in Example 6.1 down to one dimension by setting $x := uf(k)$. Here u is the current shock, x represents output, and the model is mapped into our set up via $F(x, k, u) = uk^\alpha$, $r(x, k) = \ln(x - k)$ and $\Gamma(x) = [0, x]$. The random Bellman operator has the form

$$R_n v(x) := \max_{0 \le k \le x} \left\{ \ln(x - k) + \beta \frac{1}{n} \sum_{i=1}^{n} v(U_i k^\alpha) \right\} \qquad (x \in \mathbb{X}), \tag{12}$$

where the shocks $U_1, \ldots, U_n$ are IID draws from the lognormal density specified above. In all cases we take $n = 100$. For the approximation operator A, we use several nonexpansive operators: Continuous piecewise linear interpolation (`pwise-linear`), k-nearest neighbors with $k = 1$ (`nearest nbr`) and the kernel smoother in (1) with $h = 0.25$ (`ksmooth 1`), $h = 0.5$ (`ksmooth 2`) and $h = 0.75$ (`ksmooth 3`). Algorithm 1 is applied to these specifications, with initial condition $v = w$ (i.e., starting at the utility function).

The columns from `pwise-linear` to `ksmooth 3` in Table 1 show the sup-norm distance $\|(AR_n)^j w - (AR_n)^{j-1} w\|$ between successive iterates of AR_n, with each column corresponding to a different approximation method. In all cases the distance is monotonically decreasing, as implied by the theory. (For `pwise-linear`, the first 25 iterates $(AR_n)^k w$ are themselves plotted in Figure 4. The dashed line is the true value function.) For comparison purposes, the last column (`chebyshev`) of Table 1 gives distances between iterates for the Chebyshev–quadrature method with Chebyshev polynomials of order 10. As illustrated previously in Figure 3, the sequence of iterates diverges.

As a second numerical application, we consider the stochastic LQ problem, where

$$r(x, a) = -x'Rx - 2a'Hx - a'Qa, \qquad F(x, a, u) = Mx + Na + u \tag{13}$$

and a is unconstrained. Here x is a k-vector, a is a j-vector, R is $k \times k$ and positive definite, Q is $j \times j$ and positive definite, H is $j \times k$, M is $k \times k$ and N is $k \times j$. As in the previous application, the LQ problem has an analytical solution against which the results of our algorithm can be compared.[13] We wish to investigate whether our algorithm produces accurate output with a relatively small grid and low number of iterations. (In other words, we wish to investigate the small

[13] More correctly, the LQ problem can be solved by iterating on the Riccati equation, a solution method which combines analytical and numerical elements. However, the method is accurate and stable, and hence we refer to the output of the method as the "exact" or "analytical" solution.

iterate	pwise-linear	nearest nbr	ksmooth 1	ksmooth 2	ksmooth 3	chebyshev
1	4.320550	4.320550	7.429569	8.456384	8.921369	2.476053
2	1.994024	1.733695	1.122005	1.153222	1.200513	2.624954
3	1.158638	1.113960	0.975892	1.076373	1.062914	3.449942
4	0.875121	0.856361	0.885546	1.015897	1.000330	5.212505
5	0.760378	0.750628	0.827461	0.963558	0.949328	7.880053
6	0.700180	0.691972	0.781766	0.915041	0.901760	11.910691
7	0.658340	0.650938	0.741351	0.869216	0.856661	18.007254
8	0.623338	0.616740	0.703879	0.825739	0.813827	27.227489
9	0.591538	0.586429	0.668562	0.784449	0.773136	41.169071
10	0.561770	0.557302	0.635096	0.745225	0.734479	62.249318
11	0.533623	0.528925	0.603330	0.707964	0.697755	94.123510
12	0.506925	0.503980	0.573160	0.672566	0.662867	142.318592
13	0.481573	0.477532	0.544501	0.638937	0.629724	215.191524
14	0.457493	0.454445	0.517276	0.606991	0.598238	325.378374
15	0.434618	0.431318	0.491412	0.576641	0.568326	491.985391
16	0.412887	0.409204	0.466841	0.547809	0.539909	743.902007
17	0.392243	0.390251	0.443499	0.520419	0.512914	1124.810219
18	0.372630	0.369872	0.421324	0.494398	0.487268	1700.758995
19	0.353999	0.351188	0.400258	0.469678	0.462905	2571.617068
20	0.336299	0.334070	0.380245	0.446194	0.439760	3888.390044
21	0.319484	0.316834	0.361233	0.423884	0.417772	5879.404569
22	0.303510	0.301357	0.343171	0.402690	0.396883	8889.899853
23	0.288334	0.286551	0.326013	0.382555	0.377039	13441.891687
24	0.273918	0.272124	0.309712	0.363428	0.358187	20324.689266
25	0.260222	0.258491	0.294226	0.345256	0.340278	30731.760334
26	0.247211	0.245460	0.279515	0.327993	0.323264	46467.676868
27	0.234850	0.233864	0.265539	0.311594	0.307101	70261.025402
28	0.223108	0.221631	0.252262	0.296014	0.291746	106237.540229
29	0.211952	0.210856	0.239649	0.281213	0.277158	160635.500111
30	0.201355	0.201770	0.227667	0.267153	0.263300	242887.437345
31	0.191287	0.191375	0.216283	0.253795	0.250135	367255.726034
32	0.181723	0.181114	0.205469	0.241105	0.237629	555305.658369
33	0.172636	0.171922	0.195196	0.229050	0.225747	839644.836982
34	0.164005	0.163115	0.185436	0.217598	0.214460	1269577.288914
35	0.155804	0.154937	0.176164	0.206718	0.203737	1919652.716880
36	0.148014	0.147639	0.167356	0.196382	0.193550	2902593.316376
37	0.140613	0.140678	0.158988	0.186563	0.183872	4388839.651145
38	0.133583	0.133131	0.151039	0.177235	0.174679	6636104.815231
39	0.126904	0.126643	0.143487	0.168373	0.165945	10034061.533153
40	0.120558	0.120358	0.136313	0.159954	0.157648	15171910.880616

Table 1: Distance between successive iterates of fitted value iteration

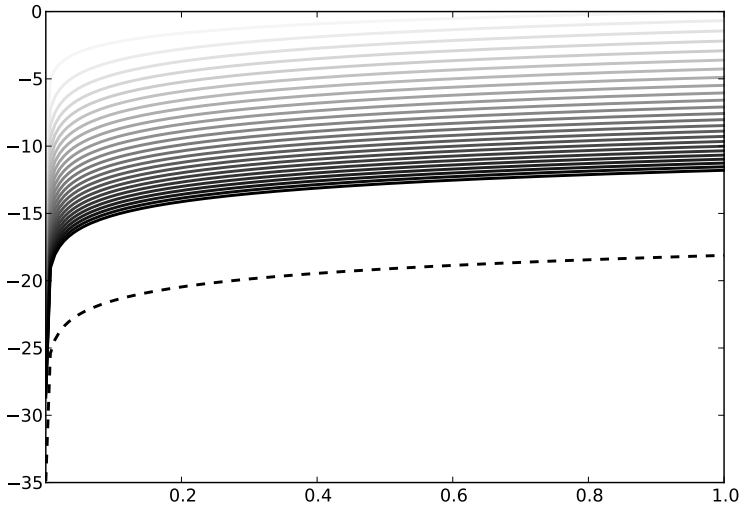

Figure 4: 25 elements of the sequence $(AR_n)^k w$

sample properties, given that our theory already covers the asymptotics.) To be specific, we focus on the quadratic problem

$$\max \mathbf{E} \sum_{t=0}^{\infty} \beta^t \left\{ -(c_t - b)^2 - \gamma i_t^2 \right\} \qquad \text{subject to}$$

$$\begin{aligned} w_{t+1} &= \delta w_t + i_t \\ c_t + i_t &= r w_t + y_t \\ y_{t+1} &= (1-\rho)\bar{y} + \rho y_t + \varepsilon_{t+1}. \end{aligned}$$

Here w denotes the stock of a commodity, y is exogenous supply of the commodity, c is consumption, i is investment, r is the net interest rate per unit of the commodity (which can be borrowed for short sales and other purposes), γ and b are preference parameters, δ parameterizes depreciation, ρ is the correlation coefficient of $\{y_t\}$, $\bar{y}$ is the unconditional mean, and $\{\epsilon_t\}_{t\geq 0}$ is an IID shock with distribution $N(0, \sigma^2)$. The problem maps into the LQ setting (13) with state $x = (w, y)$ and control $a = i$.

In our experiment, we assess the accuracy of the solution computed via Algorithm 1 by comparing it against the analytical solution. To provide a comparison with practical implications, we compare the second moments of the state variables under the approximate and analytical optimal policies.[14] The two (cen-

[14]Moments are calculated by simulating time series of length 5,000 from the approximate and

β		0.9	0.925	0.95	0.975
$\text{Var}(w_t)$	analytical	0.00294547	0.00294097	0.00293646	0.00293194
	approximate	0.00294443	0.00293989	0.00293534	0.00293078
$\text{Cov}(w_t, y_t)$	analytical	0.00230056	0.00229881	0.00229704	0.00229528
	approximate	0.00230019	0.00229842	0.00229664	0.00229486

Table 2: Approximate and analytical second moments after 5 iterations

tered) moments we consider are $\text{Var}(w_t)$ and $\text{Cov}(w_t, y_t)$. The approximation scheme is the radial basis kernel smoother (see Section 1.1) with $h = 0.00125$.

Table 2 shows the results of the simulation for 4 different values of the discount parameter β.[15] In all cases, the approximate policies are computed from 5 iterations of the fitted random Bellman operator AR_n (i.e., by setting $k = 5$ in Algorithm 1). The initial condition from which iteration begins is the constant zero function. The value of n is 100, and the number of grid points is 400 (the cartesian product of 20 grid points in each of the two dimensions). Despite the coarseness of the grid and the low number of iterations, the moments produced by the fitted value function iteration algorithm with the kernel smoother scheme are close approximations of the true values.[16]

8. Conclusion

We studied a Monte Carlo VFI algorithm with function approximation. We proved that the algorithm is consistent for a wide variety of models. This guaranteed convergence stands in contrast to many other numerical techniques proposed in the literature. Under additional restrictions, we established a parametric rate of convergence, independent of the dimension of the state, action and shock spaces.

Many avenues for future research exist. First, we identified only two cases where the ϕ-Donsker property is satisfied (the Lipschitz and monotonicity conditions of Sections 6.2 and 6.3). Additional research should illuminate other cases. In addition, we treated only stationary, additively separable, infinite horizon SDPs, leaving open the cases of nonstationary models, optimal stopping, and general recursive utility. All of these issues are left for future study.

analytical policies respectively, and taking sample averages over time.

[15]The values of the other parameters are $b = 30$, $\gamma = 1.1$, $r = 1/\beta - 1$, $\rho = 0.4$, $\bar{y} = 29$, $\delta = 0.9$ and $\sigma = 0.1$.

[16]Code for our experiments can be found at `https://sites.google.com/site/fviprobone/`.

9. Proofs

Proof of Lemma 5.1. The contractiveness of AT follows from Lemma 2.1. Next we consider contractiveness of R_n. Fix $n \in \mathbb{N}$ and $\omega \in \Omega$. Let $R := R_n(\omega)$. Fix $w, w' \in \mathscr{C}(\mathbb{X})$ and $x \in \mathbb{X}$. In view of (15), we have

$$|Rw(x) - Rw'(x)| \leq \beta \max_{a \in \Gamma(x)} \left| \frac{1}{n} \sum_{i=1}^{n} w[F(x, a, U_i(\omega))] - \frac{1}{n} \sum_{i=1}^{n} w'[F(x, a, U_i(\omega))] \right|.$$

Using the triangle inequality and the definition of $\|\cdot\|$, we obtain

$$|Rw(x) - Rw'(x)| \leq \beta \|w - w'\|.$$

Taking the supremum over $x \in \mathbb{X}$ yields the desired result.

Finally, contractiveness of AR_n now follows from Lemma 2.1. □

Proof of Lemma 5.2. Fix $\epsilon > 0$. By Lemma 2.1, we have

$$\|V_T - V_{AT}\| \leq (1 - \beta)^{-1} \|AV_T - V_T\|. \tag{14}$$

Since $\mathbb{X}$ is compact, V_T is uniformly continuous, and we select $\delta > 0$ with $|V_T(x) - V_T(y)| < (1 - \beta)\epsilon$ whenever $d(x, y) < \delta$. Using compactness again, we choose $\{x_i\}_{i=1}^m$ such that, given any $x \in \mathbb{X}$, there exists at least one x_i with $d(x, x_i) < \delta$. Finally, we choose ψ such that $\psi(u) = 0$ whenever u is greater than some constant M,[17] and h such that $Mh < \delta$.

Now fix any $x \in \mathbb{X}$. Letting $\lambda(x, i) := \psi[d(x, x_i)/h] / \sum_j \psi[d(x, x_j)/h]$, we can write

$$|AV_T(x) - V_T(x)| = \left| \sum_i \lambda(x, i) V_T(x_i) - V_T(x) \right| \leq \sum_i \lambda(x, i) |V_T(x_i) - V_T(x)|.$$

If $d(x, x_i) \geq \delta$, then $d(x, x_i)/h \geq M$, and hence $\psi[d(x, x_i)/h] = \lambda(x, i) = 0$. For the remaining terms in the sum we have $d(x, x_i) < \delta$, and hence $|V_T(x_i) - V_T(x)| < (1 - \beta)\epsilon$. Since x is arbitrary, we have $\|AV_T - V_T\| < (1 - \beta)\epsilon$. Combining this bound with (14) completes the proof of the lemma. □

Proof of Theorem 5.1. By Lemma 2.1 and the nonexpansiveness of A, we have

$$\begin{aligned} \|V_{AR_n} - V_{AT}\| &\leq \frac{1}{1 - \beta} \|AR_n V_{AT} - V_{AT}\| \\ &= \frac{1}{1 - \beta} \|AR_n V_{AT} - ATV_{AT}\| \\ &\leq \frac{1}{1 - \beta} \|R_n V_{AT} - TV_{AT}\|. \end{aligned}$$

[17] A typical example is the Epanechnikov kernel.

Hence, to prove Theorem 5.1, it is sufficient to prove that $\|R_n V_{AT} - TV_{AT}\|$ converges to zero with probability one. To bound this term, we make use of the following standard inequality: If $g, g' \in \mathscr{C}(\mathbb{Y})$ for compact set $\mathbb{Y}$, then

$$|\max g - \max g'| \leq \max |g - g'| =: \|g - g'\|. \tag{15}$$

Using (15), we obtain

$$|R_n V_{AT}(x) - TV_{AT}(x)| \\ \leq \beta \max_{a \in \Gamma(x)} \left| \frac{1}{n} \sum_{i=1}^{n} V_{AT}[F(x, a, U_i)] - \int V_{AT}[F(x, a, u)] \phi(du) \right|,$$

where $x \in \mathbb{X}$ is arbitrary. Taking the supremum over all $x \in \mathbb{X}$, we now have

$$\|R_n V_{AT} - TV_{AT}\| \\ \leq \beta \max_{(x,a) \in \mathbb{G}} \left| \frac{1}{n} \sum_{i=1}^{n} V_{AT}[F(x, a, U_i)] - \int V_{AT}[F(x, a, u)] \phi(du) \right|. \tag{16}$$

Let $y = (x, a)$ denote a typical element of $\mathbb{G}$, and let

$$h_y(u) :=: h_{(x,a)}(u) := V_{AT}[F(x, a, u)] :=: V_{AT}[F(y, u)]. \tag{17}$$

Also, for $h \colon \mathbb{U} \to \mathbb{R}$, let $\phi_n(h) := \frac{1}{n} \sum_{i=1}^{n} h(U_i)$ and $\phi(h) := \int h d\phi$. Using this notation, (16) becomes

$$\|R_n V_{AT} - TV_{AT}\| \leq \beta \max_{y \in \mathbb{G}} |\phi_n(h_y) - \phi(h_y)|. \tag{18}$$

A class $\mathscr{H}$ of bounded measurable functions mapping $\mathbb{U}$ into $\mathbb{R}$ is called ϕ-Glivenko-Cantelli if $\sup_{h \in \mathscr{H}} |\phi_n(h) - \phi(h)| \to 0$ $\mathbf{P}^*$-almost surely as $n \to \infty$. A sufficient condition for this property[18] is that $\mathscr{H}$ consists of functions $h_\alpha \colon \mathbb{U} \to \mathbb{R}$ with index α in metric space Λ, and, moreover:

1. Λ is compact;
2. $\Lambda \ni \alpha \mapsto h_\alpha(u) \in \mathbb{R}$ is continuous for every $u \in \mathbb{U}$; and
3. there exists a measurable function $H \colon \mathbb{U} \to \mathbb{R}$ such that $\int H d\phi < \infty$ and $|h_\alpha| \leq H$ for every $\alpha \in \Lambda$.

[18] See, for example, van der Vaart, 1998, p. 272.

In our case, the relevant class of functions is $\{h_y\}_{y\in\mathbb{G}}$, where h_y is defined in (17). This family of functions satisfies the sufficient conditions in 1–3 above. First, $\mathbb{G}$ is a compact metric space, due to our assumptions on $\mathbb{X}$, $\mathbb{A}$ and Γ. Second, $\mathbb{G} \ni y \mapsto h_y(u) := V_{AT}[F(y,u)] \in \mathbb{R}$ is continuous for every $u \in \mathbb{U}$, due to continuity of V_{AT} and F. Third, $|h_y(u)|$ is bounded above by the finite constant $\|V_{AT}\|$ for all $y \in \mathbb{G}$ and $u \in \mathbb{U}$. Hence, $\{h_y\}_{y\in\mathbb{G}}$ is ϕ-Glivenko-Cantelli. This concludes the proof. □

Proof of Proposition 6.1. We need some preliminary results and additional notation. Throughout the proof, $\mathscr{H} = \{h_y\}_{y\in\mathbb{G}}$ is the function class defined in both (9) and (17). In addition, let

$$G_n(y) := \nu_n(h_y) := \sqrt{n}(\phi_n(h_y) - \phi(h_y)) \qquad (n \in \mathbb{N},\, y \in \mathbb{G}).$$

G_n can be understood as a real-valued stochastic process indexed by $y \in \mathbb{G}$:

$$G_n(y)(\omega) = \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} h_y(U_i(\omega)) - \int h_y(u)\phi(du)\right) \in \mathbb{R}.$$

Regarding measurability, we have the following result, which is proved immediately below the current proof:

Lemma 9.1. *For each $n \in \mathbb{N}$, the following measurability results hold:*

1. *$\omega \mapsto G_n(\cdot)(\omega)$ is a $\mathscr{C}(\mathbb{G})$-valued random variable, and*
2. *$\omega \mapsto \|G_n(\cdot)(\omega)\| = \sup_{y\in\mathbb{G}} |G_n(y)(\omega)|$ is a real-valued random variable.*

In view of (18), we have

$$\|V_{AR_n} - V_{AT}\| \le \frac{\beta}{1-\beta} n^{-1/2} \sup_{y\in\mathbb{G}} |G_n(y)|.$$

Since $\mathscr{H} := \{h_y\}_{y\in\mathbb{G}}$ is ϕ-Donsker, we have $\nu_n \xrightarrow{d^*} \nu$, where ν is a Gaussian process on $\mathscr{H}$. By Lemma 2.2 and continuity of the norm $\|\cdot\|$ on $b\mathscr{H}$, we then have $\|\nu_n\| \xrightarrow{d^*} \|\nu\|$ in $\mathbb{R}$. Observe that

$$\|\nu_n\| = \sup_{h\in\mathscr{H}} |\nu_n(h)| = \sup_{y\in\mathbb{G}} |\nu_n(h_y)| = \sup_{y\in\mathbb{G}} |G_n(y)|,$$

and hence $\sup_{y\in\mathbb{G}} |G_n(y)| \xrightarrow{d^*} \|\nu\|$. By part 2 of Lemma 9.1, this is convergence in distribution in the regular sense, and, as a consequence, we have $\sup_{y\in\mathbb{G}} |G_n(y)| = O_P(1)$. Therefore

$$\|V_{AR_n} - V_{AT}\| \le \frac{\beta}{1-\beta} n^{-1/2} O_P(1) = O_P(n^{-1/2}).$$

This concludes the proof of Proposition 6.1. □

Proof of Lemma 9.1. We begin by proving measurability of $\omega \mapsto H(\cdot)(\omega)$, where

$$H(y)(\omega) = h_y(U(\omega)) = V_{AT}[F(y, U(\omega))].$$

Since $\mathbb{G}$ is compact in the product topology, the Stone–Weierstrass theorem implies that $\mathscr{C}(\mathbb{G})$ is separable. Hence, by the Pettis measurability theorem, we need only show that $\omega \mapsto \ell(H(\cdot)(\omega))$ is measurable for each ℓ in the dual space $\mathscr{C}(\mathbb{G})^*$ of $\mathscr{C}(\mathbb{G})$. By the Riesz representation theorem, $\mathscr{C}(\mathbb{G})^*$ can be identified with $\mathscr{M}(\mathbb{G})$, the space of finite signed Borel measures on $\mathbb{G}$. Thus, it remains to show that

$$\Omega \ni \omega \mapsto \int H(y)(\omega)\, \gamma(dy) \in \mathbb{R} \text{ is measurable} \quad \forall\, \gamma \in \mathscr{M}(\mathbb{G})$$

To this end it is sufficient to show that $H(y)(\omega) = V_{AT}[F(y, U(\omega)]$ is measurable with respect to the product σ-algebra $\mathscr{B}_{\mathbb{G}} \otimes \mathscr{F}$, where $\mathscr{B}_{\mathbb{G}}$ is the Borel σ-algebra on $\mathbb{G}$. Since H is continuous with respect to y and measurable with respect to ω, H is a Carathéodory function (Aliprantis and Border, 2006, Definition 4.50). As $\mathbb{G}$ is separable, measurability with respect to $\mathscr{B}_{\mathbb{G}} \otimes \mathscr{F}$ is established (Aliprantis and Border, 2006, Lemma 4.51).

Given measurability of $\omega \mapsto H(\cdot)(\omega)$, measurability of $\omega \mapsto G_n(\cdot)(\omega)$ follows from the fact that linear combinations of measurable random elements of a separable Banach space are themselves measurable.

Regarding the second claim in the lemma, measurability of $\omega \mapsto \|G_n(\cdot)(\omega)\|$ follows from measurability of $\omega \mapsto G_n(\cdot)(\omega)$, continuity of the norm as a map from $\mathscr{C}(\mathbb{G})$ to $\mathbb{R}$, and the fact that continuous transformations of measurable mappings are measurable. □

Proof of Proposition 6.2. By Proposition 6.1, it suffices to show that the class $\{h_y\}_{y\in\mathbb{G}}$ is ϕ-Donsker when (i)–(iii) hold. A sufficient condition for $\{h_y\}_{y\in\mathbb{G}}$ to be ϕ-Donsker is the existence of a measurable function $m\colon \mathbb{U} \to \mathbb{R}$ such that $\int m^2 d\phi < \infty$ and

$$|h_y(u) - h_{y'}(u)| \le m(u)\|y - y'\|_2 \qquad \forall\, y, y' \in \mathbb{G},\ u \in \mathbb{U} \tag{19}$$

(see, e.g., van der Vaart, 1998, p. 271). To find such an m, observe that V_{AT} is Lipschitz, as follows from (ii) and the relation $V_{AT} = ATV_{AT}$. As a consequence, there exists a $K < \infty$ such that, for any $y, y' \in \mathbb{G}$ and $u \in \mathbb{U}$, we have

$$\begin{aligned} |h_y(u) - h_{y'}(u)| &:= |V_{AT}[F(y,u)] - V_{AT}[F(y',u)]| \\ &\le K\|F(y,u) - F(y',u)\|_2 \le K m_0(u)\|y - y'\|_2, \end{aligned}$$

where m_0 is the function in (iii). Letting $m := Km_0$, we see that $\int m^2 d\phi = K^2 \int m_0^2 d\phi < \infty$. All the conditions are now verified, and hence $\{h_y\}_{y\in\mathbb{G}}$ is ϕ-Donsker. □

Proof of Lemma 6.1. To see this, observe that for any $y = (x, a) \in \mathbb{G}$, $y' = (x', a') \in \mathbb{G}$, and $u \in \mathbb{U}$,

$$\begin{aligned}\|Ax + Ba + Cu - Ax' - Ba' - Cu\|_2 \\ = \|A(x - x') + B(a - a')\|_2 \leq \gamma(\|x - x'\|_2 + \|a - a'\|_2),\end{aligned}$$

where γ is the maximum of the operator norms of A and B. Since $y = (x, a) \mapsto \|x\|_2 + \|a\|_2 \in \mathbb{R}$ defines a norm on $\mathbb{R}^d$, and since all norms on $\mathbb{R}^d$ are equivalent, we obtain

$$\|F(y, u) - F(y', u)\|_2 \leq M\gamma\|y - y'\|_2 \qquad \forall\, y, y' \in \mathbb{G},\ u \in \mathbb{U}$$

for some $M < \infty$. This verifies (ii). □

Proof of Proposition 6.3. From van der Vaart (1998, p. 273), it suffices to show that the class $\{h_y\}_{y \in \mathbb{G}}$ is uniformly bounded on $\mathbb{U}$, and that each element h_y is monotone increasing on $\mathbb{U}$. Since $h_y(u) = V_{AT}[F(y, u)]$, uniform boundedness will hold if V_{AT} is bounded on $\mathbb{X}$. That this is the case follows from the fact that $\mathbb{X}$ is compact and $V_{AT} \in \mathscr{C}(\mathbb{X})$.

Regarding monotonicity, we begin by showing that V_{AT} is monotone increasing. To see that this is the case, observe that V_{AT} is the fixed point of AT in $\mathscr{C}(\mathbb{X})$. Since $i\mathscr{C}(\mathbb{X})$ is a closed subset of $\mathscr{C}(\mathbb{X})$, we need only show that AT maps $i\mathscr{C}(\mathbb{X})$ into itself. Since $A\colon i\mathscr{C}(\mathbb{X}) \to i\mathscr{C}(\mathbb{X})$ by assumption, it remains to verify that T also has this property. For a proof of this fact, see Stachurski (2009, Theorem 12.1.2). As a result, V_{AT} is increasing, and the claim in the proposition now follows from Assumption (iv) above. □

References

[1] Ábrahám, A., Cárceles-Poveda, E., 2010. Endogenous Trading Constraints with Incomplete Asset Markets, Journal of Economic Theory, 145, 974–1004.

[2] Aiyagari, R. S., 1994. Uninsured Idiosyncratic Risk and Aggregate Saving, The Quarterly Journal of Economics, Vol. 109(3), 659-84.

[3] Aliprantis, C. D., Border, K. C., 2006. Infinite Dimensional Analysis, 3rd Edition, Springer-Verlag, New York.

[4] Arouba, S. B., Fernandez-Villaverde, J., Rubio-Ramirez, J.F., 2006. Comparing Solution. Methods for Dynamic Equilibrium Economies, Journal of Economic Dynamics and Control, 30, 2477–2508.

[5] Baird, L.C., 1995. Residual Algorithms: Reinforcement Learning with Function Approximation, Machine Learning: Proceedings of the Twelfth International Conference, Morgan Kaufman Publishers, San Francisco, CA, 9–12.

[6] Bellman, R. E., Kalaba, R., Kotkin, B., 1963. Polynomial Approximation: A New Computational Technique in Dynamic Programming, Mathematics of Computation, 17 (82), 155–161.

[7] Boyan, J., Moore, A., 1995. Generalization in Reinforcement Learning: Safely Approximating the Value Function, in G. Tesauro, D.S. Touretzky, and T.K. Lee (eds), Neural Information Processing Systems 7, MIT Press, Cambridge, MA, 369–376.

[8] Clementi, G. L., Hopenhayn, H., 2006. A Theory of Financing Constraints and Firm Dynamics, Quarterly Journal of Economics, 121 (1), 229–265.

[9] Drummond, C, 1996. Preventing Overshoot of Splines with Application to Reinforcement Learning, Computer Science Dept. Ottawa TR-96-05.

[10] Fukushima, K., Waki, Y., 2011. A Polyhedral Approximation Approach to Concave Numerical Dynamic Programming. Mimeo, University of Wisconsin.

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

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

[13] Huggett, M., 1997. The One-Sector Growth Model with Idiosyncratic Shocks: Steady States and Dynamics, Journal of Monetary Economics, 39 (3), 385–403.

[14] Judd, K.L., 1998. Numerical Methods in Economics, MIT Press.

[15] Judd, K.L., Solnick, A., 1994. Numerical Dynamic Programming with Shape-Preserving Splines. Unpublished manuscript.

[16] Keane, M. P., Wolpin, K. I., 1994. The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation. The Review of Economics and Statistics, 76 (4), 648–672.

[17] Kydland, F. E., Prescott, E. C., 1982. Time to Build and Aggregate Fluctuations, Econometrica, 50 (6), 1345-1370.

[18] Longstaff, F. A., Schwartz, E. S., 2001. Valuing American Options by Simulation: A Simple Least-squares Approach. Review of Financial Studies, 14, 113–147.

[19] Marimon, R., Scott, A. A. (eds.), 2001. Computational Methods for the Study of Dynamic Economies. Oxford University Press, USA.

[20] Martins-da-Rocha, F. V., Vailakis, Y., 2010. Existence and Uniqueness of a Fixed Point for Local Contractions. Econometrica, 78 (3), 1127–1141.

[21] Munos, R, Szepesvári, C., 2008. Finite Time Bounds for Fitted Value Iteration. Journal of Machine Learning Research, 1, 815–857.

[22] Pakes, A., McGuire, P., 2001. Stochastic Algorithms, Symmetric Markov Perfect Equilibrium, and the 'Curse' of Dimensionality. Econometrica, 69 (5), 1261–1281.

[23] Pijoan-Mas, J., 2006. Precautionary Savings or Working Longer Hours? Review of Economic Dynamics, 9 (2), 326–352.

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

[25] Rincon-Zapatero, J. P., Rodriguez-Palmero, C., 2003. Existence and Uniqueness of Solutions to the Bellman Equation in the Unbounded Case. Econometrica 71, 1519–55.

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

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

[28] Stachurski, J., 2008. Continuous State Dynamic Programming via Nonexpansive Approximation. Computational Economics, 31 (2), 141–160.

[29] Stachurski, J., 2009. Economic Dynamics: Theory and Computation. The MIT Press, Cambridge MA.

[30] Stokey, N. L., Lucas, R. E., Prescott, E. C., 1989. Recursive Methods in Economic Dynamics. Harvard University Press, Massachusetts.

[31] Tsitsiklis, J. N., Van Roy, B., 1996. Feature-Based Methods for Large Scale Dynamic Programming. Machine Learning, 22, 59–94.

[32] van der Vaart, A. W., 1998. Asymptotic Statistics, Cambridge series in statistical and probabilistic mathematics, Cambridge University Press, UK.