Approximate Dynamic Programming with Recursive Preferences[*]

Quentin Batista[†]

June, 2021

Abstract

This paper builds on the theory of Ren and Stachurski (2020) to study a value function iteration algorithm that uses isotone and convex/concave function approximation operators. Conditions that ensure the stability of the algorithm are developed and applied to multiple popular schemes used in economics such as piecewise continuous linear interpolation. An experimental comparison of various techniques suggests that variation diminishing spline approximation is a powerful and versatile tool for solving models with recursive preferences using fitted value function iteration. We also show that introducing momentum in the algorithm can significantly speed-up convergence.

1 Introduction

Economic and financial data contain important features that puzzle economists. To better account for these features, they have recently departed from standard additively separable preferences in favor of so-called recursive preferences. For instance, economists have had difficulty explaining the magnitude of the outperformance of stocks relative to Treasury bills. To justify this equity premium, Bansal and Yaron (2004) model consumption and dividend growth rates in conjunction with Epstein and Zin (1989) preferences. Besides, the Ellsberg paradox refers to the contradiction between predictions of

[*]I would like to thank Michihiro Kandori, Thomas J. Sargent, John Stachurski, Kyohei Okumura, and Daisuke Oyama for helpful comments and suggestions.

[†]University of Tokyo and University of Chicago. Email: qbatista@uchicago.edu

subjective expected utility theory and experimental results in settings of choice under uncertainty. To confront this issue, Klibanoff, Marinacci, and Mukerji (2005) propose a model of decision making that can explicitly reflect circumstances where a decision-maker faces ambiguity. Finally, policymakers, aware that their economic models are inherently flawed, need to confront significant uncertainty when making important decisions. Hansen and Sargent (2011) explain how this uncertainty can incorporated into economic modeling using multiplier preferences.

Solving models featuring recursive preferences has proved to be challenging. In many instances, it is not possible to characterize the solution of these models analytically, and therefore, researchers rely on numerical methods. Recent work has shown that some of the techniques employed to solve these models can be highly inaccurate. This is particularly problematic if the model is then evaluated by its ability to match the data. For example, long-run risk models in the tradition of Bansal and Yaron (2004) are typically solved using log-linearization techniques such as the ubiquitous Campbell and Shiller (1988) approximation. Unfortunately, when coupled with important nonlinear dynamics, these methods can lead to economically significant errors as recent studies such as Pohl, Schmedders, and Wilms (2018) illustrate. Besides, Lorenz, Schmedders, and Schumacher (2020) show that the quantitative results obtained by many models attempting to explain the "variance risk premium" (the premium that investors pay to hedge against fluctuations in volatility) is driven almost exclusively by inaccurate measures of conditional volatility. Aldrich and Kung (2020) show that the choice of solution method can, in some circumstances, have critical implications for asset prices and welfare costs in production-based asset pricing models. Finally, even with additively separable preferences, Dou et al. (2019) show that linearization methods can yield biased impulse responses in canonical macro-finance models with important non-linear dynamics. Such studies demonstrate a need for reliable, well-understood numerical methods.

This paper addresses this issue by studying conditions that preserve the geometric stability of fitted value function iteration algorithms; a class of global, non-linear solution methods that is well-known for models featuring additively separable preferences. We show that approximation operators that are isotone, convex/concave, and satisfy some boundary conditions preserve the stability of value function iteration by building on the abstract dynamic programming theory recently developed by

Ren and Stachurski (2018) and Ren and Stachurski (2020). While they show that, theoretically, the value function iteration algorithm converges to the optimal value function, in practice, solving dynamic programs often requires using approximation techniques. Objects such as continuous domains typically cannot be represented exactly on a computer, and as a result, functions, integrals, and the solution of optimization problems need to be approximated. The interaction between numerical error due to approximation and the value function algorithm is known to be delicate, and can lead to failure of the algorithm to converge, or even, divergence. We show that if one uses an averager function approximation scheme defined following Gordon (1995) or certain types of spline approximation, the stability of the Bellman operator is preserved.

With our theory in hand, we experimentally compare a selection of popular approximation schemes according to three criteria: (i) stability of the scheme, (ii) speed of convergence and, (iii) smoothness of the derivatives of the function approximation. Based on these criteria, variation diminishing spline approximation emerges as a tool of choice for solving dynamic programs with recursive preferences using a fitted value function iteration algorithm.

For cases with additively separable rewards, this problem has been studied by various authors. Tsitsiklis and Van Roy (1996) and Gordon (1995) provide examples where function approximation leads to instability. Stachurski (2008) shows how nonexpansive function approximation operators can preserve stability. The key feature of these operators is that they preserve the contractivity of the Bellman operator. Unfortunately, the contractivity condition fails to hold for many models with non-separable rewards, and as such, this line of reasoning cannot be applied. In contrast, Cai and Judd (2013) suggest using so-called shape-preserving techniques, that is, approximation techniques that preserve properties of the underlying function such as monotonicity or convexity. Pál and Stachurski (2013) study a randomized fitted value function iteration algorithm that features both function and integral approximation that has guaranteed convergence properties, and derive error bounds. To the best of our knowledge, little attention has been devoted to the impact of approximately solving optimization problems in the literature.

Section 2 provides preliminary notation. Section 3 outlines the framework introduced in Ren and Stachurski (2020). Section 4 develops conditions under which a value function iteration algorithm with function approximation is stable. Section 5 presents numerical experiments, and section 6 concludes.

2 Preliminaries

We adopt the notation and definitions from Ren and Stachurski (2020).

Let $\mathbb{R}^{\mathrm{X}}$ be all functions from some metric space X to $\mathbb{R}$. Let $b\mathrm{X}$ be the bounded Borel measurable functions in $\mathbb{R}^{\mathrm{X}}$ and let $bc\mathrm{X}$ be the continuous functions in $b\mathrm{X}$. Let $\|\cdot\|$ denote the supremum norm on $b\mathrm{X}$. For f and g in $\mathbb{R}^{X}$, the statement $f \leqslant g$ means $f(x) \leqslant g(x)$ for all $x \in X$ while $f \ll g$ means that $f \leqslant g - \epsilon$ for some positive constant ϵ.

Definition (Order Interval)**.** Given $a, b \in \mathcal{F} \subset b\mathrm{X}$, the order interval $I \equiv [a, b]$ is all f in $\mathcal{F}$ with $a \leqslant f \leqslant b$.

Let $I = [a, b]$ be an order interval in bcX.

Definition (Geometric Stability)**.** We call $S : I \to I$ *geometrically stable* on I if S has a unique fixed point $v^* \in I$, and we can find constants $\lambda \in (0, 1)$ and $K \in \mathbb{R}$ such that $\|S^n v - v^*\| \leqslant \lambda^n K$ for all $n \in \mathbb{N}$ and all $v \in I$.

Definition (Isotonicity)**.** S is called *isotone* on I if $Sv \leqslant Sv'$ whenever $v, v' \in \mathcal{X}$ with $v \leqslant v'$.

Definition (Convexity and Concavity)**.** S is called *convex* on I if $S(\lambda v + (1 - \lambda) v') \leqslant \lambda Sv + (1 - \lambda) Sv'$ whenever $v, v' \in I$ and $0 \leqslant \lambda \leqslant 1$. S is called concave if $-S$ is convex.

The following theorem by Du (1990) is taken for granted.

Theorem 1 (Du)**.** *Let $S : I \to I$ be isotone. If either (i) S is convex on I and $Sb \ll b$, or (ii) S is concave on I and $Sa \gg a$, then S is geometrically stable on I.*

3 Abstract MDP Model

Let X and A be metric spaces, called the state and action space respectively. Let Γ be a correspondence from X to A called the feasible correspondence and let $\mathbb{G} \equiv \{(x, a) \in X \times A : a \in \Gamma(x)\}$ be the feasible state-action pairs. A state-action aggregator H maps feasible state-action pairs (x, a) and functions $v \in bX$ into real values $H(x, a, v)$.

Fix w_1, w_2 in bcX and set $\mathcal{V} \equiv [w_1, w_2]$ in bX. Let $\mathcal{C}$ be the continuous functions in $\mathcal{V}$. We make some basic assumptions that will be assumed in every case:

(A1) The feasible correspondence Γ is nonempty, compact valued and continuous

(A2) The map $(x, a) \to H(x, a, v)$ is Borel measurable on $\mathbb{G}$ whenever $v \in \mathcal{V}$ and continuous on $\mathbb{G}$ whenever $v \in \mathcal{C}$

(A3) The state-action aggregator satisfies

$$v \leqslant v' \Longrightarrow H(x, a, v) \leqslant H(x, a, v') \text{ for all } (x, a) \in \mathbb{G}$$

(A4) The functions w_1 and w_2 satisfy

$$w_1(x) \leqslant H(x, a, v) \text{ and } H(x, a, v) \leqslant w_2(x) \text{ for all } (x, a) \in \mathbb{G}$$

Definition (Value Convexity and Value Concavity)**.** H is called value-convex if, for all $(x, a) \in \mathbb{G}$, $\lambda \in [0, 1]$ and $v, w \in \mathcal{V}$, we have:

$$H(x, a, \lambda v + (1 - \lambda) w) \leqslant \lambda H(x, a, v) + (1 - \lambda) H(x, a, w)$$

H is called value-concave if $-H$ is value-convex.

The following assumption will be used in the maximization case.

Assumption 1 (Convex Program)**.** *H is value-convex and there exists $\epsilon > 0$ such that $H(x, a, w_2) \leqslant w_2(x) - \epsilon$ for all $(x, a) \in \mathbb{G}$.*

The following assumption will be used in the minimization case.

Assumption 2 (Concave Program)**.** *H is value-concave and there exists $\epsilon > 0$ such that $H(x, a, w_2) \geqslant w_1(x) + \epsilon$ for all $(x, a) \in \mathbb{G}$.*

Let Σ be all maps from X to A such that each $\sigma \in \Sigma$ is Borel measurable and satisfies $\sigma(x) \in \Gamma(x)$ for all $x \in X$. For each $\sigma \in \Sigma$, we define the σ-value operator T_σ on $\mathcal{V}$ by

$$T_\sigma v(x) := H(x, \sigma(x), v) \quad (x \in \mathrm{X}, v \in \mathcal{V})$$

Ren and Stachurski (2020) show that T_σ is well-defined. A fixed point $v_\sigma \in \mathcal{V}$ of T_σ is called a σ-value function.

Maximization

With assumption 1 in force, a policy $\sigma^* \in \Sigma$ is called optimal if $v_{\sigma^*}(x) \geqslant v_\sigma(x)$ for all $\sigma \in \Sigma$ and all $x \in X$. The value function is defined at $x \in X$ by $v^*(x) = \sup_{\sigma \in \Sigma} v_\sigma(x)$.
A function $v \in \mathcal{V}$ is said to satisfy the Bellman equation if

$$v(x) = \max\nolimits_{a \in \Gamma(x)} H(x, a, v) \text{ for all } x \in \mathrm{X}$$

Given $v \in \mathcal{C}$, a policy $\sigma \in \Sigma$ is called v-greedy if $\sigma \in \operatorname{argmax}_{a \in \Gamma(x)} H(x, a, v)$ for all $x \in X$.
The Bellman operator T is a map sending $v \in \mathcal{C}$ into

$$Tv(x) = \max_{a \in \Gamma(x)} H(x, a, v)$$

Minimization

With assumption 2 in force, and, in the minimization setting, a policy $\sigma^* \in \Gamma$ is called optimal if $v_{\sigma^*}(x) \leqslant v_\sigma(x)$ for all $\sigma \in \Sigma$ and $x \in X$. The value function is defined by $v_{\sigma^*}(x) = \inf_{\sigma \in \Sigma} v_\sigma(x)$. A function $v \in \mathcal{V}$ is said to satisfy the Bellman equation if $v(x) = \min_{a \in \Gamma(x)} H(x, a, v)$ for all $x \in X$. A policy is called v-greedy if it satisfies $\sigma(x) \in \operatorname{argmin}_{a \in \Gamma(x)} H(x, a, v)$ for all $x \in X$. The Bellman operator is defined by $Tv(x) = \min_{a \in \Gamma(x)} H(x, a, v)$.

We are now ready to state part of the main result of Ren and Stachurski (2020).

Theorem 2 (Ren and Stachurski)**.** *If assumption 1 or assumption 2 holds, then*

(a) The Bellman equation has exactly one solution in $\mathcal{C}$ and that solution is v^*.

(b) The Bellman operator is geometrically stable on $\mathcal{C}$.

(c) A policy $\sigma \in \Sigma$ is optimal if and only if it is v^*-greedy.

(d) At least one optimal policy exists.

4 Value Function Iteration Algorithm with Function Approximation

Theorem 2 establishes the theoretical convergence of the value function iteration algorithm. In this section, we study value function iteration with function approximation. We proceed in three steps. First, we characterize the derivation of an approximate Bellman operator from an original Bellman operator. To do so, we introduce an approximation operator A, representing the function approximation technique, together with a composition operation such that the approximate Bellman operator $\hat{T}$ satisfies $\hat{T} = A \circ T$. Then, we show that the convergence properties of T can be preserved through this construction. Finally, we describe empirically relevant classes of stable approximation operators.

Throughout our discussion, an implicit assumption is that maximization problems can be solved exactly and that integrals can be computed exactly.[1] Our main tool is the following theorem.

Theorem 3. *Let A be an operator mapping $\mathcal{C}$ to itself. Let $\hat{T} = A \circ T$. Suppose that A is isotone. Maximization case: suppose that assumption 1 holds and that A is convex and satisfies the following:*

$$\forall f \in \mathcal{C}, f \ll w_2 \Longrightarrow Af \ll w_2$$

Minimization case: suppose that assumption 2 holds and that A is concave and satisfies the following:

$$\forall f \in \mathcal{C}, f \gg w_1 \Longrightarrow Af \gg w_1$$

Then, $\hat{T}$ is geometrically stable on $\mathcal{C}$.

Proof. Fix $v, v' \in \mathcal{C}$. Ren and Stachurski (2020) establish that T is isotone and convex under assumption 1 and concave under assumption 2 on $\mathcal{C}$. First, note that $\hat{T}$ is well-defined by Berge's theorem of the maximum, which implies that $Tv \in \mathcal{C}$ for all $v \in \mathcal{C}$. We establish that Du's theorem applies to $\hat{T}$.

Step 1: $\hat{T}$ is isotone.

Suppose that $v \leqslant v'$. Then, $Tv \leqslant Tv'$ holds by the isotonicity of T and $ATv \leqslant ATv'$ holds by the

[1] Approximation of integrals typically do not affect stability, but the choice of maximization routine can sometimes break it.

isotonicity of A. Therefore, $\hat{T}v \leqslant \hat{T}v'$.

Step 2.1 (Maximization case): $\hat{T}$ is convex.

Suppose that the maximization case assumption holds. Let $\lambda \in [0,1]$. We have:

$$\begin{aligned}
\hat{T}\left(\lambda v + (1-\lambda) v'\right) &= A\left(T\left(\lambda v + (1-\lambda) v'\right)\right) \\
&\leqslant A\left(\lambda T v + (1-\lambda) T v'\right) \\
&\leqslant \lambda A T v + (1-\lambda) A T v' \\
&= \lambda \hat{T} v + (1-\lambda) \hat{T} v'
\end{aligned}$$

The first and last equality hold by definition. The first inequality holds by the convexity of T and the isotonicity of A. The second inequality holds by the convexity of A.

Step 2.2 (Maximization case): $\hat{T}w_2 \ll w_2$

Suppose that the maximization case assumption holds. The proof of lemma 5.1 in Ren and Stachurski (2020) shows that $Tw_2 \ll w_2$. Therefore, $\hat{T}w_2 \ll w_2$ follows immediately by assumption.

Step 3.1 (Minimization case): $\hat{T}$ is concave.

Suppose that the minimization case assumption holds. The concavity of $\hat{T}$ can be obtained by making appropriate replacements in step 2.1.

Step 3.2 (Minimization case): $\hat{T}w_1 \gg w_1$

Suppose that the minimization case assumption holds. The proof of lemma 5.2 in Ren and Stachurski (2020) shows that $Tw_1 \gg w_1$. Therefore, $\hat{T}w_1 \gg w_1$ follows immediately by assumption.

□

Algorithm 1, which we call the fitted value function iteration algorithm, is a natural extension to value function iteration in the presence of function approximation. Theorem 3 guarantees the convergence of this algorithm.

We now proceed to describing classes of stable approximation techniques. Following Gordon (1995), we define a class of averager approximation schemes appropriate for our context and prove stability.

Algorithm 1 Fitted Value Function Iteration

Input: An approximation operator A

Initialize v_0, $\hat{T} = A \circ T$, $\Delta = 1$ and $i = 0$.
while $\Delta \geqslant$ TOL (a small positive number) **do**
    $v_i \leftarrow \hat{T} v_{i-1}$
    $\Delta \leftarrow \|v_i - v_{i-1}\|$
    $i \leftarrow i + 1$
Compute a v_{i-1}-greedy policy π.

Definition (Averager)**.** The operator $A : \mathcal{C} \to \mathcal{C}$ is called an averager if $Af(x) = \sum_{j=1}^{m} \beta_j(x) f(x_j)$ such that, for all $x \in \mathrm{X}$, $\sum_{j=1}^{m} \beta_j(x) = 1$ and $\beta_j(x) \geqslant 0$ for all $1 \leqslant j \leqslant m$.

For convenience, we assume below that w_1 and w_2 are constant functions. For given bounds $w_1, w_2 \in bcX$, constant bounds can be constructed by taking the supremum or infinimum of w_1 and w_2 respectively over X.

Theorem 4. *Suppose that A is an averager. If assumption 1 (maximization case) or assumption 2 (minimization case) holds, then $\hat{T} \equiv A \circ T$ is geometrically stable on $\mathcal{C}$.*

Proof. We show that theorem 3 is applicable to A. The isotonicity, convexity, and concavity of A follow from the linearity of $Af(x)$ in $\{f(x_j)\}_{j=1}^{m}$ for all $f \in \mathcal{C}$ and $x \in \mathrm{X}$.

We now show that for all $f \in \mathcal{C}$, $f \ll w_2$ implies $Af \ll w_2$. Fix any $f \in \mathcal{C}$ and $x \in \mathrm{X}$ such that $f \leqslant w_2 - \epsilon$ for some $\epsilon > 0$. Then,

$$\begin{aligned} Af(x) &= \sum_{j=1}^{m} \beta_j(x) f(x_j) \\ &\leqslant \max_i f(x_i) \sum_{j=1}^{m} \beta_j(x) \\ &\leqslant w_2\left(\arg\max_i f(x_i)\right) \sum_{j=1}^{m} \beta_j(x) - \epsilon \\ &= w_2(x) - \epsilon \end{aligned}$$

The first equality holds by definition. The first inequality holds because $\beta_j(x) \geqslant 0$ for all $0 \leqslant j \leqslant m$.

The second inequality holds by assumption. The second equality holds by the assumption that w_2 is a constant function.

By the arbitrariness of x, $Af \leqslant w_2 - \epsilon$ holds. A similar argument can be used to show that $\forall f \in \mathcal{C}, f \gg w_1 \Longrightarrow Af \gg w_1$ by making appropriate modifications.

□

Remark. *Ren and Stachurski (2018) use constant functions w_1 and w_2 for applications to risk sensitive, Epstein-Zin preferences, and the recursive smooth ambiguity model proposed by Ju and Miao (2012).*

We now provide an example of an approximation scheme that belongs to the class of averagers.

Example (Kernel averagers)**.** Kernel-based approximators can be represented by an expression of the form

$$Kv\left(x\right) = \frac{\sum_{i=1}^{k} K_h\left(x_i - x\right) v\left(x_i\right)}{\sum_{i=1}^{k} K_h\left(x_i - x\right)}$$

where $K_h : \mathrm{X} \to \mathbb{R}$ is a nonnegative mapping for a sequence of points $\{x_i\}_{i=1}^{k} \subset \mathrm{X}$. This representation clearly reveals that theorem 4 applies to this scheme.

We now develop conditions that apply to particular spline functions. We follow the treatment of Lyche and Morken (2018). For simplicity, we assume $\mathrm{X} = \mathbb{R}$.

Definition (B-spline)**.** Let d be a nonnegative integer and let $\boldsymbol{t} = \{t_j\}_{j=1}^{n+d+1}$ be a nondecreasing sequence of real numbers of length at least $d+2$. If $1 \leq j \leq n$, the j-th B-spline of degree d with knots $\boldsymbol{t}$ is defined by

$$B_{j,d,\boldsymbol{t}}\left(x\right) = \frac{x - t_j}{t_{j+d} - t_j} B_{j,d-1,\boldsymbol{t}}\left(x\right) + \frac{t_{j+1+d} - x}{t_{j+1+d} - t_{j+1}} B_{j+1,d-1,\boldsymbol{t}}\left(x\right)$$

for all real numbers x, starting with

$$B_{j,0,\boldsymbol{t}}(x) = \begin{cases} 1 & \text{if } t_j \leqslant x \leqslant t_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

Lemma 1 (Nonnegativity of B-Splines)**.** *Let d be a nonnegative polynomial degree and let $\boldsymbol{t} = \{t_j\}_{j=1}^{n+d+1}$ be a knot sequence. Then, $B_{j,d,\boldsymbol{t}}(x) \geqslant 0$ for all $x \in \mathrm{X}$.*

Proof. Lemma 2.3 in Lyche and Morken (2018) establishes that B-splines on $\boldsymbol{t}$ have the following properties:

1. Local support: $x \notin [t_j, t_{j+d+1}) \implies B_{j,d,\boldsymbol{t}}(x) = 0$

2. Positivity: $x \in (t_j, t_{j+d+1}) \implies B_{j,d,\boldsymbol{t}}(x) > 0$

It remains to show that $B_{j,d,\boldsymbol{t}}(t_j) \geqslant 0$. Using the local support property, we have $B_{j+1,d-1,\boldsymbol{t}}(t_j) = 0$. Therefore, the conclusion follows. $\square$

Definition (Spline functions)**.** Let $\boldsymbol{t} = \{t_j\}_{j=1}^{n+d+1}$ be a nondecreasing sequence of real numbers of length at least $d+2$. The linear space of all linear combinations of B-splines is the spline space $\mathbb{S}_{d,\boldsymbol{t}}$ defined by:

$$\mathbb{S}_{d,\boldsymbol{t}} = \left\{ \sum_{j=1}^{n} c_j B_{j,d,\boldsymbol{t}} : c_j \in \mathbb{R}, 1 \leqslant j \leqslant n \right\}$$

An element f in $\mathbb{S}_{d,\boldsymbol{t}}$ is called a spline function.

To develop our result for spline functions, we will make use of the following lemma:

Lemma 2. *For all y and y' in $\mathbb{R}^n$ and A in $\mathbb{R}^{n \times n}$, $y \geqslant y'$ implies $Ay \geqslant Ay'$ if and only if A is nonnegative.*

Proof. Fix A in $\mathbb{R}^{n \times n}$.

Step 1: if A is nonnegative, then, for all y and y' in $\mathbb{R}^n$, $y \geqslant y'$ implies $Ay \geqslant Ay'$

Suppose that A is nonnegative and fix y and y' in $\mathbb{R}^n$ such that $y \geqslant y'$. Then, $y - y' \geqslant 0$. Therefore, $A(y - y') \geqslant 0$ by the nonnegativity of A, implying $Ay \geqslant Ay'$.

Step 2: if, for all y and y' in $\mathbb{R}^n$, $y \geqslant y'$ implies $Ay \geqslant Ay'$, then A is nonnegative

We proceed by contraposition. Suppose that A is not nonnegative. Then, there exists $1 \leqslant i, j \leqslant n$

such that $A_{ij} < 0$. Let e_j denote a unit vector whose jth row is equal to 1. Then, $Ae_j \geqslant \mathbf{0}$ does not hold despite $e_j \geqslant \mathbf{0}$. $\square$

We now state a result applicable to a class of approximation techniques based on spline functions.

Theorem 5. *For some function f, let there be given data $\{x_i\}_{i=1}^m$ and let $y_i = f(x_i)$. For some matrix W, let S be an operator satisfying:*

$$
\begin{aligned}
(Sf)(x) &= \sum_{j=1}^{m} c_j B_{j,d,\boldsymbol{t}}(x) \\
\text{where} \qquad & \boldsymbol{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} = W \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = W\boldsymbol{y}
\end{aligned}
$$

such that $w_1(x) < c_j < w_2(x)$ for all $1 \leqslant j \leqslant m$ and $x \in \mathrm{X}$ if $w_1 \ll f \ll w_2$.

Then, $\hat{T} = S \circ T$ is geometrically stable on $\mathcal{C}$ if W is nonnegative and assumption 1 (maximization case) or 2 (minimization case) holds.

Proof. We show that the conditions of theorem 3 apply.

Step 1: S is isotone.

By lemma 1, $B_{j,d,\boldsymbol{t}}(x) \geqslant 0$ for all x. Suppose that $f \geqslant f'$ for some function f and f'. Then, we have $\boldsymbol{y} \geqslant \boldsymbol{y}'$ by definition. Let $\boldsymbol{c}$ and $\boldsymbol{c}'$ denote the vector of coefficients associated with Sf and Sf' respectively. Then, $Sf \geqslant Sf'$ if $\boldsymbol{c} \geqslant \boldsymbol{c}'$. By lemma 2, this holds if W is nonnegative.

Step 2: S is convex and concave.

This follows from the linearity of $\boldsymbol{c}$ in $\boldsymbol{y}$.

Step 3: $w_1 \ll f \ll w_2$ implies $w_1 \ll Sf \ll w_2$

From lemma 5.28 in Lyche and Morken (2018), we have $\min_j c_j \leqslant Sf \leqslant \max_j c_j$. Therefore, $w_1 \ll Sf \ll w_2$ follows by assumption as required. $\square$

Example (Continuous piecewise linear interpolation)**.** Let $\{x_i, y_i\}_{i=1}^m$ be a set of data points with

$x_i < x_{i+1}$ for $i = 1, \ldots, m-1$ and $\boldsymbol{t} = (x_1, x_1, x_2, x_3, \ldots, x_{m-1}, x_m, x_m)$. For a given function f, continuous piecewise linear interpolation satisfies:

$$(Lf)(x) = \sum_{i=1}^{m} y_i B_{i,1,\boldsymbol{t}}(x)$$

Notice that $W = I$, and therefore, it is a nonnegative matrix. The boundary conditions of theorem follow from assumption and lemma 5.28 in Lyche and Morken (2018). Therefore, $\hat{T} = L \circ T$ is geometrically stable on $\mathcal{C}$.

Example (Variation diminishing spline approximation)**.** Let f be a continuous function on the interval $[a, b]$, let d be a positive integer, and let $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{n+d+1})$ be a knot vector such that $\tau_{d+1} = a$ and $\tau_{n+1} = b$. The spline is given by

$$\begin{aligned} (Vf)(x) &= \sum_{j=1}^{n} f\left(\tau_j^*\right) B_{i,d,\boldsymbol{t}}(x) \\ \tau_j^* &= \frac{\tau_{j+1} + \cdots + \tau_{j+d}}{d} \end{aligned}$$

Once again, we have $W = I$. The boundary conditions of theorem hold by proposition 5.29 in Lyche and Morken (2018)[2]. Hence, $\hat{T} = V \circ T$ is geometrically stable on $\mathcal{C}$.

Remark. *These last two examples can also be shown to belong to the class of averagers.*

Another class of popular interpolation techniques rely on orthogonal polynomials. We show that Chebyshev polynomials fail to produce isotone approximation operators in general, which is one the key properties that our theory exploits to establish stability. We follow the exposition of Judd (1998).

Example (Chebyshev Polynomials)**.** Let $X = [-1, 1]$. For a function $f : X \to \mathbb{R}$, approximation with Chebyshev polynomials of order n is defined as follows:

$$C_n f(x) = \sum_{i=0}^{n} a_i T_i(x)$$

[2]The proposition guarantees that $\forall x \in \mathbb{R} : \min_y f(y) \leqslant (Vf)(x) \leqslant \max_y f(y)$. Therefore, we have $\forall x \in \mathbb{R} : w_1(x) \ll (Vf)(x) \ll w_2(x)$.

where:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x) \end{aligned}$$

Let $f(x) = |x|$ and $g(x) = x$. Then, $g \leqslant f$ holds on $[-1, 1]$ and:

$$\begin{aligned} C_1 f(x) &= \frac{2}{\pi} \\ C_1 g(x) &= x \end{aligned}$$

Therefore, $C_1 g(1) > C_1 f(1)$, implying that there exists n such that C_n is not isotone. We conjecture that this statement is true for all n.

5 Numerical Experiments

5.1 Comparison of approximation techniques

Which approximation technique should one use in practice? In this section, we carry out a numerical experiment to provide an answer to this question using three selection criteria: (i) stability of the scheme, (ii) speed of convergence and, (iii) smoothness of the derivatives of the approximation. We consider a neoclassical growth model modified to feature robust preferences. The model is described by the following set of equations:

$$
\begin{aligned}
v(x) &= \max_{0 \leqslant k \leqslant x} \left\{ \log(x-k) + \frac{\beta}{\theta} \log \mathbb{E}\left[\exp\left(\theta v\left(x'\right)\right)\right] \right\} \\
\text{subject to} \quad & x' = z k^{\alpha} \\
& \log z \sim \mathcal{N}(\mu, \sigma)
\end{aligned}
$$

We solve this model using algorithm 1 with seven different approximation schemes: Chebyshev polynomials, piecewise continuous linear interpolation, two kernel smoothers with Gaussian radial basis functions with smoothing parameters 0.25 and 0.75 respectively, variation diminishing spline approximation (VDSA), Akima interpolation, and monotonic cubic splines. Numerical integration is done using Gaussian quadrature, and Brent's method is used for numerical optimization. The initial condition for the guess of the value function is the constant zero function. We calibrate the parameters as follows: $\theta = 10$, $\beta = 0.95$, $\alpha = 0.33$, $\mu = 0$, and $\sigma = 0.25$. Table 1 shows the supremum norm of the guess of the value function at successive iterates of the algorithm. Stopping criteria for the algorithm are typically functions of this distance, and as such, we measure speed by the reciprocal of the number of iterations needed to achieve a fixed distance level. Under this definition of speed, piecewise linear interpolation, VDSA, Akima interpolation, and monotonic cubic spline approximation are approximately equally fast.

Out of these four approximation techniques, our theory predicts that two are always stable when only function approximation is used: piecewise linear interpolation and variation diminishing spline approximation. Monotone cubic approximation preserves monotonicity of the underlying function that Cai and Judd (2013) argue is important for preserving stability, but it is not isotone. It may be the case that shape-preservation is in fact a sufficient condition for stability, but our theory does not allow us to answer this question.

An advantage of VDSA over piecewise linear interpolation is that the smoothness of the approximate function depends on the choice of spline degree. Optimization routines that exploit derivatives tend to work better with smoother functions, and as such, VDSA appears to be the best technique among the ones that we have used for this experiment. Hence, in the absence of additional information to guide the selection, VSDA seems to be a good default technique.[3]

This example also illustrates that the instability issue with Chebyshev polynomials that is well-known in the case of additively separable preferences also afflicts the case of recursive preferences. Pál and Stachurski (2013) provide an example in a closely related model. In their case, the distance between successive iterates appears to go to infinity while it seems to get stuck at a high level of error in our case.

5.2 Accelerating convergence with momentum

Caldara et al. (2012) report that value function iteration algorithms can be slow compared to other solution methods for models with Epstein-Zin preferences and stochastic volatility. In this section, we describe a simple modification of algorithm 1 that, in some cases, can speed-up convergence by close to two orders of magnitude. This modification exploits a link between value function iteration and gradient descent to introduce momentum in the algorithm. It has recently been studied in the context of Markov Decision Processes with additively separable rewards by Goyal and Grand-Clement (2019). Besides, Vieillard et al. (2019) apply a similar idea to reinforcement learning problems. Algorithm 2 describes the modification that interests us.

Algorithm 2 Fitted Value Function Iteration with Momentum

Input: An approximation operator A

Initialize v_0, $\hat{T} = A \circ T$, $\Delta = 1$ and $i = 0$.
while $\Delta \geqslant$ TOL (a small positive number) **do**
  $v_i \leftarrow (1-\beta)\, v_{i-1} + \beta \hat{T} v_{i-1} + \alpha\, (v_{i-1} - v_{i-2})$
  $\Delta \leftarrow \|v_i - v_{i-1}\|$
  $i \leftarrow i + 1$
Compute a v_{i-1}-greedy policy π.

Note that a fixed point under the update rule implied by algorithm 1 is also a fixed point under the update rule implied by algorithm 2 for any values of α and β.

For our experiment, we consider the model of Bansal and Yaron (2004) described by the following system of equations:

[3] Naturally, we cannot claim that VSDA is uniformly superior to other schemes. For instance, if the researcher knows that the value function is piecewise linear, then it would probably be more sensible to use piecewise linear interpolation.

Table 1: Distance between successive iterates of fitted value iteration

Iterate	Chebyshev	Piecewise linear	Kernel (0.25)	Kernel (0.75)	VDSA	Akima	Mono. Cubic
1	1.479531	3.314650	5.291498	6.836801	3.367189	3.314650	3.314650
2	1.021866	1.220389	0.653361	1.145836	1.221344	1.218222	1.218264
3	0.766832	0.541841	0.615677	1.058076	0.542533	0.542523	0.542597
4	0.756796	0.435744	0.584833	0.996234	0.435801	0.435745	0.435745
5	0.676985	0.413956	0.555585	0.944474	0.414165	0.413955	0.413955
6	0.663027	0.393257	0.527805	0.896857	0.393397	0.393256	0.393257
7	0.822923	0.373594	0.501414	0.851936	0.373711	0.373594	0.373594
8	0.606802	0.354914	0.476344	0.809323	0.355001	0.354913	0.354914
9	0.590811	0.337168	0.452527	0.768854	0.337250	0.337168	0.337168
10	0.547549	0.320310	0.429900	0.730411	0.320383	0.320310	0.320309
11	0.766856	0.304294	0.408405	0.693890	0.304363	0.304294	0.304294
12	0.516323	0.289080	0.387985	0.659196	0.289145	0.289079	0.289079
13	0.511602	0.274626	0.368586	0.626236	0.274688	0.274626	0.274625
14	0.472070	0.260894	0.350156	0.594924	0.260954	0.260894	0.260894
15	0.686196	0.247850	0.332649	0.565178	0.247906	0.247850	0.247850
16	0.437263	0.235457	0.316016	0.536919	0.235510	0.235457	0.235457
17	0.440293	0.223684	0.300215	0.510073	0.223734	0.223684	0.223684
18	0.407225	0.212500	0.285205	0.484569	0.212548	0.212500	0.212500
19	0.605496	0.201875	0.270944	0.460341	0.201920	0.201875	0.201875
20	0.385200	0.191781	0.257397	0.437324	0.191824	0.191781	0.191781
21	0.389553	0.182192	0.244527	0.415458	0.182233	0.182192	0.182192
22	0.364842	0.173083	0.232301	0.394685	0.173121	0.173082	0.173083
23	0.575397	0.164428	0.220686	0.374951	0.164465	0.164428	0.164428
24	0.340736	0.156207	0.209652	0.356203	0.156242	0.156207	0.156207
25	0.370169	0.148397	0.199169	0.338393	0.148430	0.148397	0.148397
26	0.331378	0.140977	0.189211	0.321473	0.141008	0.140977	0.140977
27	0.526831	0.133928	0.179750	0.305400	0.133958	0.133928	0.133928
28	0.303088	0.127232	0.170763	0.290130	0.127260	0.127232	0.127231
29	0.318671	0.120870	0.162224	0.275623	0.120897	0.120870	0.120870
30	0.289859	0.114827	0.154113	0.261842	0.114852	0.114827	0.114827
31	0.498890	0.109085	0.146408	0.248750	0.109110	0.109086	0.109085
32	0.268979	0.103631	0.139087	0.236312	0.103654	0.103631	0.103631
33	0.302569	0.098449	0.132133	0.224497	0.098471	0.098449	0.098450
34	0.266996	0.093527	0.125526	0.213272	0.093548	0.093527	0.093527
35	0.465479	0.088851	0.119250	0.202608	0.088870	0.088851	0.088851
36	0.244831	0.084408	0.113287	0.192478	0.084427	0.084408	0.084408
37	0.263333	0.080188	0.107623	0.182854	0.080206	0.080188	0.080188
38	0.240636	0.076178	0.102242	0.173711	0.076195	0.076179	0.076178
39	0.448945	0.072369	0.097130	0.165026	0.072386	0.072369	0.072369
40	0.221532	0.068751	0.092273	0.156774	0.068766	0.068751	0.068751

$$
\begin{aligned}
V_t &= \left((1-\delta) C_t^{1-\rho} + \delta \mathcal{R}(V_{t+1})^{1-\rho} \right)^{\frac{1}{1-\rho}} \\
\mathcal{R}(V_t) &= \mathbb{E}\left[V_t^{1-\gamma} \right]^{\frac{1}{1-\gamma}} \\
\text{subject to} \quad & x_{t+1} = \rho x_t + \phi_e \sigma_t e_{t+1} \\
& \log\left(\frac{C_{t+1}}{C_t} \right) = \mu + x_t + \sigma_t \eta_{t+1} \\
& \log\left(\frac{D_{t+1}}{D_t} \right) = \mu + \phi x_t + \phi_d \sigma_t u_{t+1} \\
& \sigma_{t+1}^2 = \sigma^2 + \nu_1 \left(\sigma_t^2 - \sigma^2 \right) + \sigma_w w_{t+1} \\
& e_{t+1}, \eta_{t+1}, u_{t+1}, w_{t+1} \sim \mathcal{N}(0,1)
\end{aligned}
$$

We solve this model on a two-dimensional uniform grid with ten points in each dimension. We use continuous piecewise linear interpolation as our function approximation technique. For numerical integration, we use Gauss-Hermite quadrature with ten sample points. The initial condition for the guess of the value function is the constant zero function. We set our stopping criterion TOL equal to 10^{-10}. For parameter values, we use the original calibration of Bansal and Yaron (2004).

Can algorithm 2 significantly outperform algorithm 1 in terms of speed? We investigate this question experimentally by numerically estimating optimal values for α and β for a range of discount factor values. These parameter values are optimal in the sense that they maximize the speed-up magnitude defined as the base 10 logarithm of the ratio of number of iterations needed to satisfy the stopping criterion.[4] Figure 1 plots our results.[5] For high values of the discount factor, the estimated speed-up magnitude approaches 2. We also plot the number of iterations needed to achieve convergence using algorithm 1. A comparison between the two plots reveal that algorithm 2 is most powerful for solving difficult problems. Specifically, as the number of iterations needed to achieve convergence increases, the estimated optimal speed-up magnitude increases, thereby significantly mitigating the

[4]Formally, letting $i_{\alpha,\beta}$ denote the number of iterations using parameter values α and β, we estimate $\max_{\alpha,\beta}\left(-\log_{10}\left(\frac{i_{\alpha,\beta}}{i_{0,1}}\right)\right)$.

[5]To check the validity of our results, we also check that the estimated value function using algorithm 1 and 2 are very close to each other.

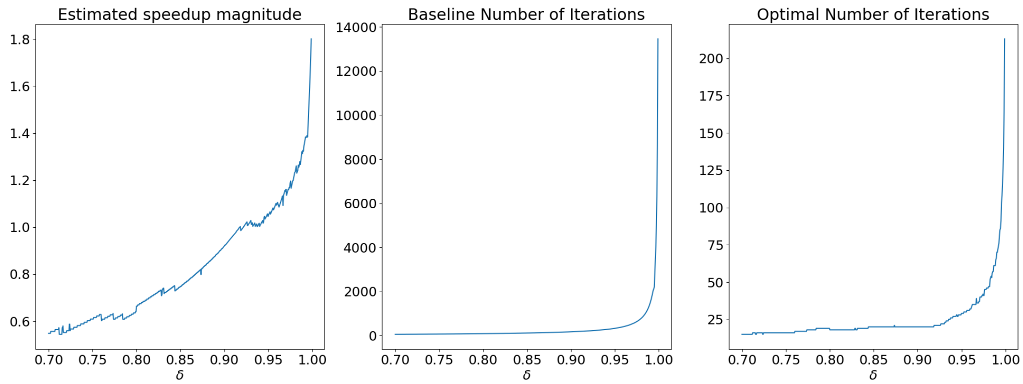

Figure 1: Estimated optimal speed-up magnitude as a function of the discount factor.

explosive increase in the number of iterations needed to satisfy the stopping criterion.

While techniques such as parallelization allow to reduce the time that a given iteration takes, by definition, it cannot help accelerate the part of the value function algorithm that is not parallelizable. In contrast, algorithm 2 operates by reducing the total number of iterations needed to reach a stopping criterion. Given that these two acceleration techniques operate along orthogonal dimensions, our experiment suggests that they can be combined effectively.

Unfortunately, our strategy for estimating optimal parameters is not practical because it requires solving the dynamic program of interest multiple times. As such, for the algorithm to be useful in practice, we need to devise a computational strategy such that the computational cost of using the strategy is lower than that of using the standard algorithm. Ideally, we would want to derive the optimal parameters analytically. We leave this problem to future work.

Nevertheless, we suggest heuristic strategies for choosing parameters. Given that that algorithm 2 is equivalent to algorithm 1 when $\beta = 1$ and $\alpha = 0$, a conservative strategy is to pick a value of β close to 1 and a value of α close to 0. Figure 2 plots the speed-up magnitude for a range of parameters values. This figure shows that one can obtain reasonable speed-ups even for very conservative choices of parameter values.

A more ambitious strategy involves estimating values of α and β that maximize the speed of

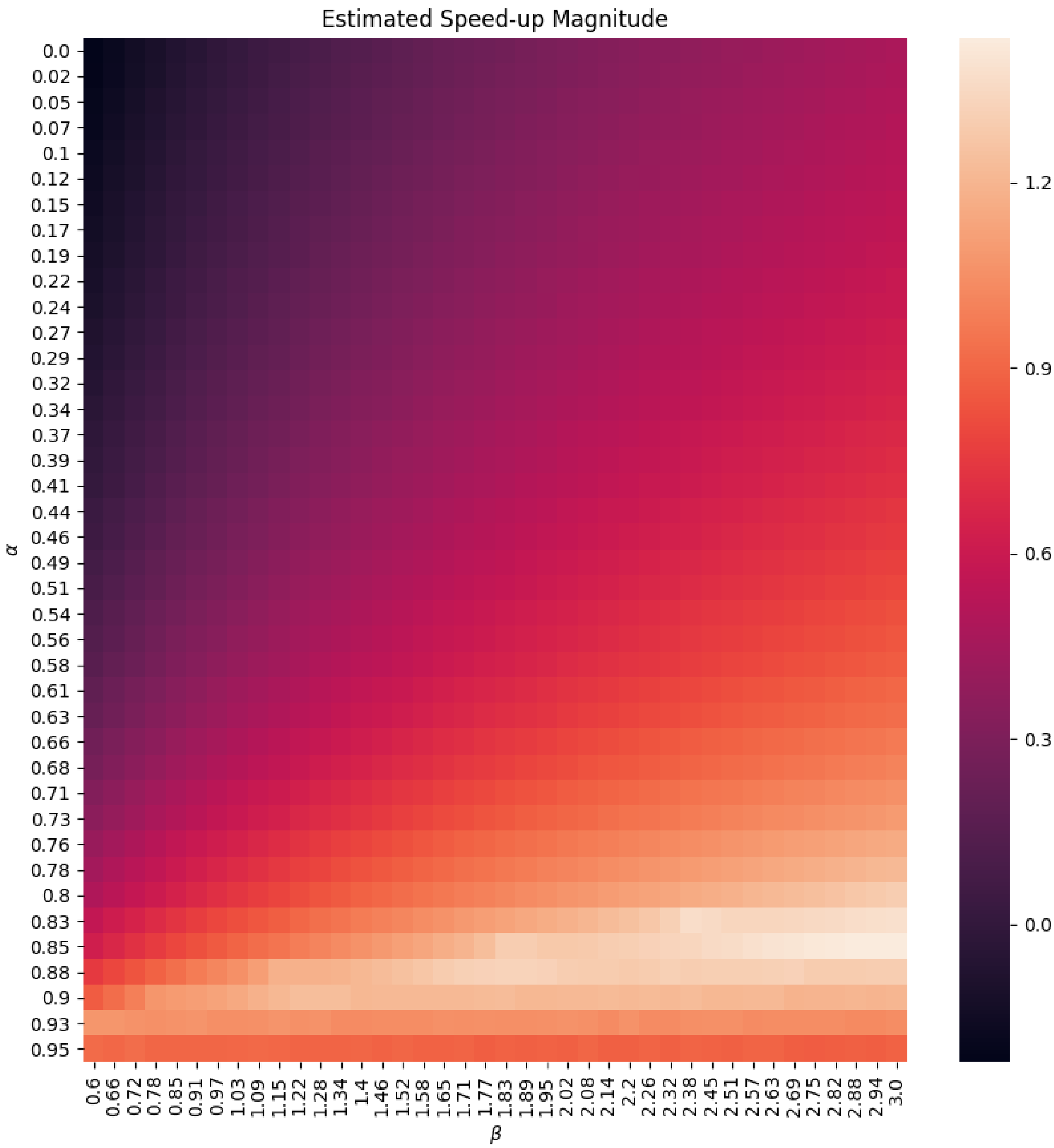

Figure 2: Speed-up magnitude for different values of α and β.

convergence of the algorithm using homotopy methods. We first estimate optimal values for α and β numerically for parameter values that make solving the dynamic program computationally cheap (e.g. a low discount factor or a small number of grid points). We can then use our estimated parameters to solve harder and harder dynamic programs until we converge to our original problem. This strategy will work well if the optimal parameters are relatively invariant to the parameters being modified, which may or may not be true.

A different approach would be to adaptively modify parameter values. For instance, one could construct sequences of parameters α_m and β_m such that $\alpha_m \to 0$ and $\beta_m \to 1$. Alternatively, the distance between successive iterates could be used to guide the process. If the distance between successive iterates increases, then, the parameters would be shrinked towards 0 and 1. Otherwise, a search process could be implemented to try to improve the speed of convergence.

6 Conclusion

This paper shows that isotone and convex/concave approximation operators preserve the stability of value function iteration algorithms with function approximation subject to some boundary conditions. The framework is sufficiently general to accommodate models that feature recursive preferences. Popular polynomial interpolation schemes such as Chebyshev or Hermite polynomials do not satisfy these conditions, but frequently used techniques such as continuous piecewise linear interpolation do. Our numerical experiments suggest that variation diminishing spline approximation is a tool of choice for doing function approximation. We have also shown that introducing momentum in the algorithm can significantly speed-up convergence. An interesting avenue for future research is to study strategies for effectively implementing this modified algorithm in practice.

References

Aldrich, Eric Mark and Howard Kung (2020). "Computational Methods for Production-Based Asset Pricing Models with Recursive Utility". In: *Stud. Nonlinear Dyn. Econom.* ISSN: 15583708. DOI: 10.1515/snde-2017-0003.

Bansal, Ravi and Amir Yaron (2004). *Risks for the long run: A potential resolution of asset pricing puzzles.* DOI: 10.1111/j.1540-6261.2004.00670.x.

Cai, Yongyang and Kenneth Judd (2013). "Shape-preserving dynamic programming". In: *Math. Methods Oper. Res.* ISSN: 14322994. DOI: 10.1007/s00186-012-0406-5.

Caldara, Dario et al. (2012). "Computing DSGE models with recursive preferences and stochastic volatility". In: *Rev. Econ. Dyn.* ISSN: 10942025. DOI: 10.1016/j.red.2011.10.001.

Campbell, John Y. and Robert J. Shiller (1988). "The Dividend-Price Ratio and Expectations of Future Dividends and Discount Factors". In: *Rev. Financ. Stud.* ISSN: 0893-9454. DOI: 10.1093/rfs/1.3.195.

Dou, Winston Wei et al. (2019). "Macro-Finance Models with Nonlinear Dynamics". In: *Univ. Chicago, Becker Friedman Inst. Econ. Work. Pap. Forthcom.*

Du, Yihong (1990). "Fixed points of increasing operators in ordered banach spaces and applications". In: *Appl. Anal.* 38.1-2, pp. 1–20. ISSN: 1563504X. DOI: 10.1080/00036819008839957.

Epstein, Larry G. and Stanley E. Zin (1989). "Substitution, Risk Aversion, and the Temporal Behavior of Consumption and Asset Returns: A Theoretical Framework". In: *Econometrica.* ISSN: 00129682. DOI: 10.2307/1913778.

Gordon, Geoffrey J. (1995). "Stable Function Approximation in Dynamic Programming". In: *Mach. Learn. Proc. 1995.* DOI: 10.1016/b978-1-55860-377-6.50040-2.

Goyal, Vineet and Julien Grand-Clement (2019). *A first-order approach to accelerated value iteration.* arXiv: 1905.09963.

Hansen, Lars Peter and Thomas J. Sargent (2011). *Robustness.* ISBN: 9781400829385. DOI: 10.2307/3007866.

Ju, Nengjiu and Jianjun Miao (2012). "Ambiguity, Learning, and Asset Returns". In: *Econometrica*. ISSN: 0012-9682. DOI: 10.3982/ecta7618.

Judd, Kenneth (1998). *Numerical Methods in Economics*. arXiv: arXiv:1011.1669v3.

Klibanoff, Peter, Massimo Marinacci, and Sujoy Mukerji (2005). "A smooth model of decision making under ambiguity". In: *Econometrica*. ISSN: 00129682. DOI: 10.1111/j.1468-0262.2005.00640.x.

Lorenz, Friedrich, Karl Schmedders, and Malte Schumacher (2020). "Nonlinear Dynamics in Conditional Volatility". In: *SSRN Electron. J.* Pp. 1–48. DOI: 10.2139/ssrn.3575458.

Lyche, Tom and Knut Morken (2018). *Spline Methods*, p. 36.

Pál, Jen and John Stachurski (2013). "Fitted value function iteration with probability one contractions". In: *J. Econ. Dyn. Control*. ISSN: 01651889. DOI: 10.1016/j.jedc.2012.08.003.

Pohl, Walter, Karl Schmedders, and Ole Wilms (2018). "Higher Order Effects in Asset Pricing Models with Long-Run Risks". In: *J. Finance*. ISSN: 15406261. DOI: 10.1111/jofi.12615.

Ren, Guanlong and John Stachurski (2018). "Discrete Time Dynamic Programming with Recursive Preferences: Optimality and Applications". In: 2004, pp. 1–42. arXiv: 1812.05748. URL: http://arxiv.org/abs/1812.05748.

— (2020). "Dynamic Programming with Value Convexity". In: pp. 1–10.

Stachurski, John (2008). "Continuous state dynamic programming via nonexpansive approximation". In: *Comput. Econ.* ISSN: 09277099. DOI: 10.1007/s10614-007-9111-5.

Tsitsiklis, John N. and Benjamin Van Roy (1996). "Feature-based methods for large scale dynamic programming". In: *Mach. Learn.* ISSN: 08856125. DOI: 10.1007/BF00114724.

Vieillard, Nino et al. (2019). *Momentum in reinforcement learning*. arXiv: 1910.09322.