

A Polyhedral Approximation Approach to Concave Numerical Dynamic Programming*

Kenichi Fukushima

Yuichiro Waki

University of Wisconsin, Madison

University of Queensland

November 27, 2012

Abstract

This paper introduces a numerical method for solving concave continuous state dynamic programming problems which is based on a pair of polyhedral approximations of concave functions. The method is globally convergent and produces computable upper and lower bounds on the value function which can in theory be made arbitrarily tight. This is true regardless of the pattern of binding constraints, the smoothness of model primitives, and the dimensionality and rectangularity of the state space. We illustrate the method's performance using an optimal firm management problem subject to credit constraints and partial investment irreversibilities.

1 Introduction

This paper concerns continuous state numerical dynamic programming problems in which the return and constraint functions are continuous and concave. Such problems arise frequently in economics, often in inner loops for algorithms that solve much harder problems. There is therefore a desire for solution methods that are reliable, precise, and efficient.

Existing methods with the broadest applicability and greatest reliability are those based on value iterations, which if implemented exactly will generate a sequence of functions that converges to the value function from any starting point thanks to its contraction mapping property. When the state variables are continuous, however, an exact implementation of the procedure is infeasible as it requires storing infinitely many numbers in memory and solving

*We thank the referees, Paul Klein, Ellen McGrattan, and especially John Stachurski for helpful comments. (First version circulated: June 2011.)

infinitely many optimization problems per iteration. In practice one therefore approximately implements the procedure in one way or another.

There are two standard approaches here. The first is to discretize the state space, that is, simply replace the original state space with one that is finite. This approach is numerically stable—the iterations are guaranteed to converge because their contraction mapping property is preserved—but generally slow. The second approach is to compute the updated function values on a finite grid and then interpolate those values, either exactly or approximately, to generate a function to be used as input in the next iteration. This approach is often faster than the first but is generally less reliable as most interpolation methods break the contraction property of the iterations and can thereby cause non-convergence (see, e.g., Judd, 1998, p. 438). For problems with one-dimensional state spaces there is a satisfactory solution to the latter problem based on shape preserving splines (Judd and Solnick, 1994). However comparable techniques remain relatively scarce for problems with multi-dimensional state spaces. In particular, currently known techniques (cf. Gordon, 1995; and Stachurski, 2008), when applied to concave problems, generally introduce non-concavities which make it difficult to solve the optimization problems reliably and efficiently.

In addition to confronting users with this difficult tradeoff, existing methods are also limited in their ability to tell precisely how accurate the computed solution is. It is now common practice to address this issue by checking if certain necessary conditions for optimality—such as intertemporal Euler equations—hold with high accuracy. There are conditions under which such tests are known to have sound theoretical foundations (Santos, 2000); however it is not straightforward to adapt them to problems with occasionally binding constraints and/or other sources of non-smoothness.

The purpose of this paper is to introduce a method based on a pair of polyhedral approximations of concave functions which improves upon existing methods along these dimensions. In particular, the method is globally convergent, preserves concavity of the problem, and produces computable upper and lower bounds on the value function which can in theory be made arbitrarily tight. Furthermore, these properties hold true regardless of the pattern of binding constraints, the smoothness of model primitives, and the dimensionality and rectangularity of the state space.

These features of our method make it particularly well suited for solving in a robust manner problems with occasionally binding constraints, non-differentiabilities, or multi-dimensional state spaces that may be non-rectangular. One such problem is the optimal firm management problem with credit constraints and partial investment irreversibilities in Khan and Thomas (2011). We use this as an example to test the practical performance of our method and find it to be reasonably efficient.

Our method consists of two components and each have important predecessors in the literature. The first component, which produces lower bounds on the value function, is close to a method based on piecewise affine interpolations analyzed by Santos and Vigo-Aguiar (1998) and the lottery based method of Phelan and Townsend (1991) for solving dynamic contracting problems. The second component, which produces upper bounds on the value function, is similar in some ways to what Nishimura and Stachurski (2009) used to analyze a model of primary commodity markets. Both components also fall into a broad class of methods outlined by Gordon (1995) and Stachurski (2008) for which convergence is guaranteed. Our method is also closely related to Judd, Yeltekin, and Conklin's (2003) method for solving repeated games, and can in fact be viewed as its adaptation to dynamic programming problems. As far as we know, however, no paper has combined these strands in the literature into a general purpose method of the kind that we develop here.

The main limitation of our approach is that it works only with concave problems, and this constraint sometimes does bind in practice. While it is usually possible to get around it by introducing lotteries or other randomization devices, doing so may or may not be reasonable depending on the application.

2 Setup

We focus throughout on a general infinite horizon dynamic programming problem as treated in Stokey, Lucas, and Prescott (1989, Chapter 9). Our terminology on convex analysis follows Rockafellar (1970).

The decision problem is described by the following elements. The endogenous state variable x and control variable y (which becomes the next period's endogenous state) both belong to $X \subset \mathbb{R}^n$, which we take to be a polytope (i.e., the convex hull of a finite set of points). The random shocks z follow a time homogeneous Markov chain with finite state space Z , and the probability of transiting from state z to z' is $\pi(z'|z)$. The discount factor is $\beta \in (0, 1)$. The return function $r : X \times X \times Z \rightarrow \mathbb{R}$ is continuous and concave in its first $2n$ arguments, and we let $r_{\min} := \min r(X \times X \times Z)$ and $r_{\max} := \max r(X \times X \times Z)$. The set of feasible controls at state $(x, z) \in X \times Z$ is $\Gamma(x, z) = \{y \in X : h(x, y, z) \geq 0\}$, where $h : X \times X \times Z \rightarrow \mathbb{R}^m$. Each component of h is continuous and concave in its first $2n$ arguments and $\Gamma(x, z)$ is non-empty for any $(x, z) \in X \times Z$. For later reference we let $P := \mathbb{R}^n$ and define $r^* : P \times P \times Z \rightarrow \mathbb{R}$ as:

$$r^*(p, q, z) = \min_{(x, y) \in X \times X} \{p \cdot x + q \cdot y - r(x, y, z) : h(x, y, z) \geq 0\}.$$

Associated with this problem is the operator \mathbb{T} which maps $v : X \times Z \rightarrow \mathbb{R}$ to $\mathbb{T}v : X \times Z \rightarrow \mathbb{R}$, given by:

$$\mathbb{T}v(x, z) = \sup_{y \in \Gamma(x, z)} \{r(x, y, z) + \beta \mathbb{E}v(y, z)\} \quad (1)$$

where the operator \mathbb{E} maps $v : X \times Z \rightarrow \mathbb{R}$ to $\mathbb{E}v : X \times Z \rightarrow \mathbb{R}$, given by:

$$\mathbb{E}v(y, z) = \sum_{z' \in Z} v(y, z') \pi(z'|z).$$

As is well known, \mathbb{T} is a monotone β -contraction on $B(X \times Z)$, the Banach space of bounded real valued functions on $X \times Z$ equipped with the supremum norm $\|\cdot\|$, and has the value function V as its unique fixed point in $B(X \times Z)$. Standard arguments imply that V is continuous and concave.

3 Approach

We begin by introducing two operators which approximate general concave functions by polyhedral concave functions. One produces approximations from below while the other produces approximations from above.

To set the stage, let $S \subset \mathbb{R}^l$ be a polytope.

The first operator \mathbb{L} (for “lower”) uses a grid \hat{S} on S which contains all vertices of S and maps $f : S \rightarrow \mathbb{R}$ to $\mathbb{L}f : S \rightarrow \mathbb{R}$, given by

$$\mathbb{L}f(s) = \max_{\mu \in M(s, \hat{S})} \sum_{\hat{s} \in \hat{S}} \mu(\hat{s}) f(\hat{s}). \quad (2)$$

where $M(s, \hat{S})$ is the set of probability distributions on \hat{S} with mean s . We will sometimes write $\mathbb{L}_{\hat{S}}$ to make the dependence on \hat{S} explicit.

The second operator \mathbb{U} (for “upper”) uses a grid \hat{D} on $D := \mathbb{R}^l$ such that $0 \in \hat{D}$ and maps $f : S \rightarrow \mathbb{R}$ to $\mathbb{U}f : S \rightarrow \mathbb{R}$, given by

$$\mathbb{U}f(s) = \min_{\hat{d} \in \hat{D}} \{\hat{d} \cdot s - f^*(\hat{d})\}, \quad (3)$$

where f^* is the concave conjugate of f . We will sometimes write $\mathbb{U}_{\hat{D}}$ to make the dependence on \hat{D} explicit.

Figure 1 illustrates how these operators work. They are called inner/outer linearizations in the applied mathematics literature (e.g., Bertsekas and Yu, 2011), and can be viewed as

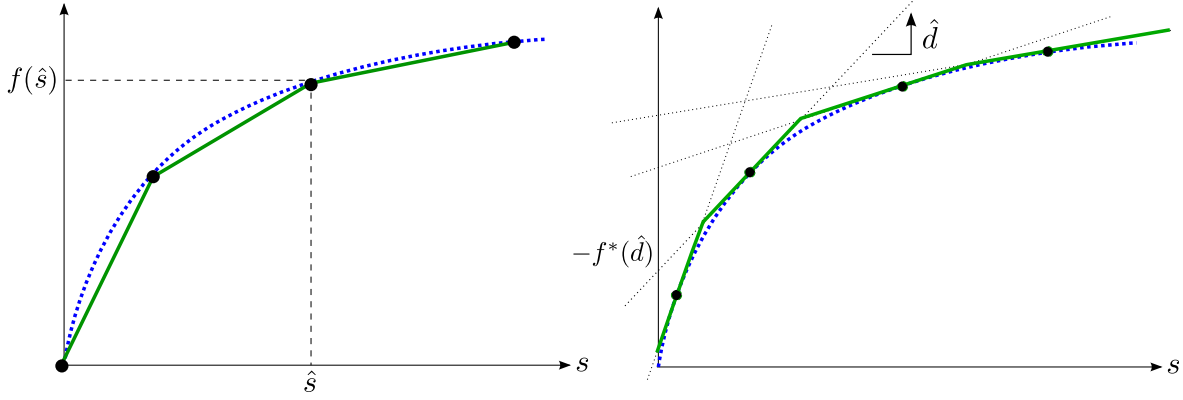


Figure 1: Left panel: The operator \mathbb{L} . The dotted line is the graph of f and the solid line is the graph of $\mathbb{L}f$. Right panel: The operator \mathbb{U} . The dotted line is the graph of f and the solid line is the graph of $\mathbb{U}f$.

applications of Judd, Yeltekin, and Conklin's (2003) inner/outer approximations of convex sets to the subgraph of f .

Note that the approximating functions $\mathbb{L}f$ and $\mathbb{U}f$ are both completely summarized by the finite lists of numbers $f(\hat{S})$ and $f^*(\hat{D})$ respectively, no matter how "complicated" f is. As we will see, this is part of what makes these operators useful for computations.

The following lemmas establish some basic properties of \mathbb{L} and \mathbb{U} . Most of them are intuitively plausible given figure 1. Note that not all parts require concavity or continuity however. (All proofs are in appendix A.)

Lemma 1L. (i) If f is concave then $\mathbb{L}_{\bar{S}}f \leq \mathbb{L}_{\hat{S}}f \leq f$ for any grids \bar{S} and \hat{S} satisfying $\bar{S} \subset \hat{S}$. If f is also continuous then for any $\epsilon \in \mathbb{R}_{++}$ one can choose \bar{S} so that $f - \epsilon \leq \mathbb{L}_{\bar{S}}f$. (ii) If $f \leq f'$ then $\mathbb{L}f \leq \mathbb{L}f'$. (iii) $\mathbb{L}(f + a) = \mathbb{L}f + a$ for $a \in \mathbb{R}$. (iv) If $f \equiv 0$ then $\mathbb{L}f \equiv 0$.

Lemma 1U. (i) If f is concave then $f \leq \mathbb{U}_{\bar{D}}f \leq \mathbb{U}_{\hat{D}}f$ for any grids \bar{D} and \hat{D} satisfying $\bar{D} \subset \hat{D}$. If f is also continuous then for any $\epsilon \in \mathbb{R}_{++}$ one can choose \bar{D} so that $\mathbb{U}_{\bar{D}}f \leq f + \epsilon$. (ii) If $f \leq f'$ then $\mathbb{U}f \leq \mathbb{U}f'$. (iii) $\mathbb{U}(f + a) = \mathbb{U}f + a$ for $a \in \mathbb{R}$. (iv) If $f \equiv 0$ then $\mathbb{U}f \equiv 0$.

We next use the operators \mathbb{L} and \mathbb{U} to define approximations of \mathbb{T} which we denote \mathbb{T}^L and \mathbb{T}^U . What we want them to do, of course, is to approximate \mathbb{T} from below and above respectively in a theoretically reasonable and computationally convenient manner.

The operator \mathbb{T}^L maps $v : X \times Z \rightarrow \mathbb{R}$ to $\mathbb{T}^L v : X \times Z \rightarrow \mathbb{R}$, given by:

$$\mathbb{T}^L v(x, z) = \max_{y \in \Gamma(x, z)} \{r(x, y, z) + \beta \mathbb{E}_{\hat{X}} v(\cdot, z)(y)\} \quad (4)$$

where \hat{X} is a grid on X which contains all of its vertices.

The operator \mathbb{T}^U maps $v : X \times Z \rightarrow \mathbb{R}$ to $\mathbb{T}^U v : X \times Z \rightarrow \mathbb{R}$, given by:

$$\mathbb{T}^U v(x, z) = \max_{y \in \Gamma(x, z)} \{r(x, y, z) + \beta \mathbb{E} \mathbb{U}_{\hat{P}} v(\cdot, z)(y)\} \quad (5)$$

where \hat{P} is a grid on P which contains the zero vector.

The following theorems formalize the sense in which \mathbb{T}^L and \mathbb{T}^U approximate \mathbb{T} from below and above:

Theorem 2L. \mathbb{T}^L is a monotone β -contraction on $B(X \times Z)$ whose unique fixed point V^L satisfies $V^L \leq V$. For any $\epsilon \in \mathbb{R}_{++}$ there exists a grid \bar{X} such that $V - \epsilon \leq V^L$ if $\bar{X} \subset \hat{X}$.

Theorem 2U. \mathbb{T}^U is a monotone β -contraction on $B(X \times Z)$ whose unique fixed point V^U satisfies $V \leq V^U$. For any $\epsilon \in \mathbb{R}_{++}$ there exists a grid \bar{P} such that $V^U \leq V + \epsilon$ if $\bar{P} \subset \hat{P}$.

An important property of \mathbb{T}^L and \mathbb{T}^U from a computational standpoint is that the finite lists of numbers $v(\hat{X} \times Z)$ and $v^*(\hat{P} \times Z)$ (where $v^*(\cdot, z)$ is the concave conjugate of $v(\cdot, z)$ for each $z \in Z$) completely summarize the data contained in v needed to compute $\mathbb{T}^L v$ and $\mathbb{T}^U v$. This means that we can implement fixed point iterations on these operators by simply keeping track of and updating these lists. The following theorems give the updating formulas.

Theorem 3L. For each $(\hat{x}, z) \in \hat{X} \times Z$:

$$\mathbb{T}^L v(\hat{x}, z) = \max_{(y, \mu) \in \Gamma(\hat{x}, z) \times \mathbb{R}_+^{|\hat{X}| \times |Z|}} \left\{ r(\hat{x}, y, z) + \beta \sum_{z' \in Z} \sum_{\hat{y} \in \hat{X}} \mu(\hat{y}, z') v(\hat{y}, z') \pi(z'|z) : \right. \\ \left. \sum_{\hat{y} \in \hat{X}} \mu(\hat{y}, z') = 1, \forall z' \in Z; \quad \sum_{\hat{y} \in \hat{X}} \mu(\hat{y}, z') \hat{y} = y, \forall z' \in Z \right\} \quad (6)$$

Theorem 3U. For each $(\hat{p}, z) \in \hat{P} \times Z$:

$$(\mathbb{T}^U v)^*(\hat{p}, z) = \max_{(q, \lambda) \in P \times \mathbb{R}_+^{|\hat{P}| \times |Z|}} \left\{ r^*(\hat{p}, q, z) + \sum_{z' \in Z} \sum_{\hat{q} \in \hat{P}} \lambda(\hat{q}, z') v^*(\hat{q}, z') : \right. \\ \left. \sum_{\hat{q} \in \hat{P}} \lambda(\hat{q}, z') = \beta \pi(z'|z), \forall z' \in Z; \quad \sum_{\hat{q} \in \hat{P}} \sum_{z' \in Z} \lambda(\hat{q}, z') \hat{q} = -q \right\} \quad (7)$$

Note here that the maximization problems in (6) and (7) simultaneously take care of the “interpolation step” (where one constructs $\mathbb{L}v$ and $\mathbb{U}v$) and the “optimization step” (where

one solves the maximization problems in (4) and (5)). In (7), the maximization problem also takes care of the conjugate operation that maps $\mathbb{T}^U v$ to $(\mathbb{T}^U v)^*$.

Thus, given any v_0^L and v_0^U we can compute $v_N^L = (\mathbb{T}^L)^N v_0^L$ and $v_N^U = (\mathbb{T}^U)^N v_0^U$ for each $N \in \mathbb{N}$. Because \mathbb{T}^L and \mathbb{T}^U are contractions, the sequences $(v_N^L)_{N \in \mathbb{N}}$ and $(v_N^U)_{N \in \mathbb{N}}$ generated in this way are guaranteed to converge to V^L and V^U respectively as $N \rightarrow \infty$.

We already know that the limiting functions V^L and V^U bound the true value function V from below and above. But because neither is computable in a finite number of steps, we need to go a step further if we are to make this property useful in practice. Our suggestion is to exploit the following monotone convergence results:

Theorem 4L. *Suppose v_0^L satisfies $v_0^L \leq \mathbb{T}^L v_0^L$ (one example being the constant function $v_0^L \equiv r_{\min}/(1 - \beta)$). Let $v_N^L = \mathbb{T}^L v_{N-1}^L$ for $N \in \mathbb{N}$. Then $v_N^L \uparrow V^L$ as $N \rightarrow \infty$.*

Theorem 4U. *Suppose v_0^U satisfies $v_0^U \geq \mathbb{T}^U v_0^U$ (one example being the constant function $v_0^U \equiv r_{\max}/(1 - \beta)$). Let $v_N^U = \mathbb{T}^U v_{N-1}^U$ for $N \in \mathbb{N}$. Then $v_N^U \downarrow V^U$ as $N \rightarrow \infty$.*

These results, combined with the previous ones, give us the following recipe for computing lower and upper bounds on the value function V : start from initial values v_0^L and v_0^U that satisfy the hypotheses of Theorems 4L/U, compute v_N^L and v_N^U for some N using the formulas in Theorems 3L/U, and set $v^L(\cdot, z) := \mathbb{L}v_N^L(\cdot, z)$ and $v^U(\cdot, z) := \mathbb{U}v_N^U(\cdot, z)$ for each $z \in Z$. Our results guarantee that $v^L \leq V^L \leq V \leq V^U \leq v^U$ and that these bounds can be made arbitrarily tight by refining the grids and making N large.

The final step now is to calculate a policy function $g : X \times Z \rightarrow X$. One reasonable approach here is to use v^L as an estimate of V and let g be v^L -greedy, namely:

$$g(x, z) \in \operatorname{argmax}_{y \in \Gamma(x, z)} \{r(x, y, z) + \beta \mathbb{E}v^L(y, z)\}. \quad (8)$$

In this case the following theorem provides a bound on the suboptimality of g which is in principle computable:

Theorem 5. *The policy g is ϵ -optimal, where $\epsilon = \|v^U - v^L\|$.*

4 Implementation

We turn next to some techniques for efficiently implementing our method.

4.1 LP approximations

A key step in implementing our method is to efficiently handle the maximization problems in (6) and (7). Both are non-linear programs with many variables, and they can be costly to solve when the non-linear functions r , h , and r^* are hard to evaluate and/or insufficiently smooth. In our experience, an effective way to handle this part is to convert these maximization problems to linear programs by applying polyhedral approximations to all non-linear functions.

For instance, one could approximate $\mathbb{T}^L v$ by applying \mathbb{L} to r and h to obtain:

$$\begin{aligned} \tilde{\mathbb{T}}^L v(\hat{x}, z) = & \max_{(y, \mu_r, \mu_h, \mu_v) \in X \times \mathbb{R}_+^{|\hat{X}|} \times \mathbb{R}_+^{|\hat{X}|} \times \mathbb{R}_+^{|\hat{X}| \times |Z|}} \\ & \left\{ \sum_{\hat{y} \in \hat{X}} \mu_r(\hat{y}) r(\hat{x}, \hat{y}, z) + \beta \sum_{z' \in Z} \sum_{\hat{y} \in \hat{X}} \mu_v(\hat{y}, z') v(\hat{y}, z') \pi(z'|z) : \right. \\ & \sum_{\hat{y} \in \hat{X}} \mu_r(\hat{y}) = 1; \sum_{\hat{y} \in \hat{X}} \mu_r(\hat{y}) \hat{y} = y; \\ & \sum_{\hat{y} \in \hat{X}} \mu_h(\hat{y}) h(\hat{x}, \hat{y}, z) \geq 0; \sum_{\hat{y} \in \hat{X}} \mu_h(\hat{y}) = 1; \sum_{\hat{y} \in \hat{X}} \mu_h(\hat{y}) \hat{y} = y; \\ & \left. \sum_{\hat{y} \in \hat{X}} \mu_v(\hat{y}, z') = 1, \forall z' \in Z; \sum_{\hat{y} \in \hat{X}} \mu_v(\hat{y}, z') \hat{y} = y, \forall z' \in Z \right\}. \end{aligned}$$

The problem above is a linear program, and it follows from $r(x, \cdot, z) \geq \mathbb{L}r(x, \cdot, z)$ and $h(x, \cdot, z) \geq \mathbb{L}h(x, \cdot, z)$ that $\tilde{\mathbb{T}}^L v \leq \mathbb{T}^L v$ for any v .

For \mathbb{T}^U , a straightforward approach is to proceed similarly and approximate $\mathbb{T}^U v$ by applying \mathbb{L} to r^* to obtain:

$$\begin{aligned} (\tilde{\mathbb{T}}^U v)^*(\hat{p}, z) = & \max_{(q, \lambda_{r^*}, \lambda_v) \in P \times \mathbb{R}_+^{|\hat{P}|} \times \mathbb{R}_+^{|\hat{P}| \times |Z|}} \left\{ \sum_{\hat{q} \in \hat{P}} \lambda_{r^*}(\hat{q}) r^*(\hat{p}, \hat{q}, z) + \sum_{z' \in Z} \sum_{\hat{q} \in \hat{P}} \lambda_v(\hat{q}, z') v^*(\hat{q}, z') : \right. \\ & \sum_{\hat{q} \in \hat{P}} \lambda_{r^*}(\hat{q}) = 1; \sum_{\hat{q} \in \hat{P}} \lambda_{r^*}(\hat{q}) \hat{q} = q; \\ & \left. \sum_{\hat{q} \in \hat{P}} \lambda_v(\hat{q}, z') = \beta \pi(z'|z), \forall z' \in Z; \sum_{\hat{q} \in \hat{P}} \sum_{z' \in Z} \lambda_v(\hat{q}, z') \hat{q} = -q \right\}. \end{aligned}$$

This problem again is a linear program, and from $r^*(p, \cdot, z) \geq \mathbb{L}r^*(p, \cdot, z)$ we know that $(\mathbb{T}^U v)^* \geq (\tilde{\mathbb{T}}^U v)^*$ and hence $\mathbb{T}^U v \leq \tilde{\mathbb{T}}^U v$ for any v .

For many problems, however, the following alternative approximation of \mathbb{T}^U works better than the one listed above, although its derivation is somewhat more complicated. This is because it is often easier to evaluate the values and (sub)gradients of r and h than it is to evaluate r^* (which generally requires non-linear programming). First, approximate r^* by:

$$\tilde{r}^*(p, q, z) = \min_{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n} \{p \cdot x + q \cdot y - \mathbb{U}r(\cdot, \cdot, z)(x, y) : \mathbb{U}h(\cdot, \cdot, z)(x, y) \geq 0\}.$$

Here, the grids for \mathbb{U} are taken so that:

$$\begin{aligned} \mathbb{U}r(\cdot, \cdot, z)(x, y) &= \min_{(\hat{x}, \hat{y}) \in \hat{X}^2} \{d_r(\hat{x}, \hat{y}, z) \cdot (x, y) - c_r(\hat{x}, \hat{y}, z)\} \\ \mathbb{U}h(\cdot, \cdot, z)(x, y) &= \min_{(\hat{x}, \hat{y}) \in \hat{X}^2} \{d_h(\hat{x}, \hat{y}, z) \cdot (x, y) - c_h(\hat{x}, \hat{y}, z)\} \end{aligned}$$

where $d_r(\hat{x}, \hat{y}, z)$ and $d_h(\hat{x}, \hat{y}, z)$ are (sub)gradients of $r(\cdot, \cdot, z)$ and $h(\cdot, \cdot, z)$ at (\hat{x}, \hat{y}) , and

$$\begin{aligned} c_r(\hat{x}, \hat{y}, z) &= d_r(\hat{x}, \hat{y}, z) \cdot (\hat{x}, \hat{y}) - r(\hat{x}, \hat{y}, z), \\ c_h(\hat{x}, \hat{y}, z) &= d_h(\hat{x}, \hat{y}, z) \cdot (\hat{x}, \hat{y}) - h(\hat{x}, \hat{y}, z). \end{aligned}$$

Next rewrite this as a linear program in epigraph form:

$$\begin{aligned} \tilde{r}^*(p, q, z) &= \min_{(x, y, \tau_r, \tau_h) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}_+} \{p \cdot x + q \cdot y - \tau_r : \\ \tau_r &\leq d_r(\hat{x}, \hat{y}, z) \cdot (x, y) - c_r(\hat{x}, \hat{y}, z), \quad \forall (\hat{x}, \hat{y}) \in \hat{X}^2; \\ \tau_h &\leq d_h(\hat{x}, \hat{y}, z) \cdot (x, y) - c_h(\hat{x}, \hat{y}, z), \quad \forall (\hat{x}, \hat{y}) \in \hat{X}^2\} \end{aligned}$$

and use duality to obtain:

$$\begin{aligned} \tilde{r}^*(p, q, z) &= \max_{(\lambda_r, \lambda_h) \in \mathbb{R}_+^{|\hat{X}|^2} \times \mathbb{R}_+^{|\hat{X}|^2}} \left\{ \sum_{(\hat{x}, \hat{y}) \in \hat{X}^2} [\lambda_r(\hat{x}, \hat{y}) c_r(\hat{x}, \hat{y}, z) + \lambda_h(\hat{x}, \hat{y}) c_h(\hat{x}, \hat{y}, z)] : \right. \\ &\quad \left. \sum_{(\hat{x}, \hat{y}) \in \hat{X}^2} \lambda_r(\hat{x}, \hat{y}) = 1; \quad \sum_{(\hat{x}, \hat{y}) \in \hat{X}^2} [\lambda_r(\hat{x}, \hat{y}) d_r(\hat{x}, \hat{y}, z) + \lambda_h(\hat{x}, \hat{y}) d_h(\hat{x}, \hat{y}, z)] = (p, q) \right\}. \end{aligned}$$

Finally, replace r^* in (7) by \tilde{r}^* to obtain the approximation:

$$\begin{aligned}
(\tilde{\mathbb{T}}^U v)^*(\hat{p}, z) = & \max_{(q, \lambda_r, \lambda_h, \lambda_v) \in P \times \mathbb{R}_+^{|\hat{X}|^2} \times \mathbb{R}_+^{|\hat{X}|^2} \times \mathbb{R}_+^{|\hat{P}| \times |Z|}} \\
& \left\{ \sum_{(\hat{x}, \hat{y}) \in \hat{X}^2} [\lambda_r(\hat{x}, \hat{y}) c_r(\hat{x}, \hat{y}, z) + \lambda_h(\hat{x}, \hat{y}) c_h(\hat{x}, \hat{y}, z)] + \sum_{z' \in Z} \sum_{\hat{q} \in \hat{P}} \lambda_v(\hat{q}, z') v^*(\hat{q}, z') : \right. \\
& \sum_{(\hat{x}, \hat{y}) \in \hat{X}^2} \lambda_r(\hat{x}, \hat{y}) = 1; \quad \sum_{(\hat{x}, \hat{y}) \in \hat{X}^2} [\lambda_r(\hat{x}, \hat{y}) d_r(\hat{x}, \hat{y}, z) + \lambda_h(\hat{x}, \hat{y}) d_h(\hat{x}, \hat{y}, z)] = (\hat{p}, q); \\
& \left. \sum_{\hat{q} \in \hat{P}} \lambda_v(\hat{q}, z') = \beta \pi(z'|z), \quad \forall z' \in Z; \quad \sum_{\hat{q} \in \hat{P}} \sum_{z' \in Z} \lambda_v(\hat{q}, z') \hat{q} = -q \right\}.
\end{aligned}$$

Once again this is a linear program, and we have $r^* \geq \tilde{r}^*$ which implies $(\mathbb{T}^U v)^* \geq (\tilde{\mathbb{T}}^U v)^*$ and hence $\mathbb{T}^U v \leq \tilde{\mathbb{T}}^U v$ for any v .

It is straightforward to check that each of the above approximations $\tilde{\mathbb{T}}^L$ and $\tilde{\mathbb{T}}^U$ are monotone β -contractions on $B(X \times Z)$. It follows from this and the fact that $\tilde{\mathbb{T}}^L v \leq \mathbb{T}^L v \leq \mathbb{T} v \leq \mathbb{T}^U v \leq \tilde{\mathbb{T}}^U v$ for any v that one can use $\tilde{\mathbb{T}}^L$ and $\tilde{\mathbb{T}}^U$ just like \mathbb{T}^L and \mathbb{T}^U to calculate bounds on V from below and above.

Several factors contribute to the effectiveness of this scheme. One is that it can leverage well established techniques for linear programming. Simplex methods are especially effective here thanks to the availability of warm starts (for example, the solution to the maximization problem at one point in the state space is typically close to that at a nearby point). Another is that it makes it possible to pre-compute the relevant values of r and h before the main iterations, which is beneficial when those functions are costly to evaluate. This approach can also handle any non-smoothness in r or h with ease.

Finally, while the size of the linear programs above may be problematic for very large problems, our experience is that it is often possible to mitigate this issue by tuning the above formulas to the problem at hand by, for instance, using different grids to approximate different functions and/or exploiting special structure such as partial linearities or separabilities in r and/or h .

4.2 Combining with other acceleration techniques

One strength of our method is that it works well with a number of standard acceleration techniques. We give a list of examples below.

Parallelizing. Our method can be parallelized at two levels: First, one can carry out the iterations on \mathbb{T}^L and \mathbb{T}^U independently. Second, one can distribute the maximization problems that need to be solved at each grid point across a number of separate processors (as is standard). The fact that our method does not have an independent, hard-to-parallelize “interpolation step” helps with scaling in the latter case.

Policy function iterations. Policy function iterations, both in pure and modified forms, can be combined with our method in the usual manner. Doing so does not interfere with the robustness of our method as it preserves its monotone convergence properties. Moreover, the policy iterations in this case require only sparse linear algebra (as in finite state problems), which helps with efficiency.

Multigrid methods. One can also combine our method with a multigrid method along the lines of Chow and Tsitsiklis (1991). Again, doing so preserves our method’s monotone convergence properties as the value function approximations can only improve as one refines the grids.

5 Example

We now use an example to illustrate a use of our method and its performance.

5.1 Problem description

We consider an optimal firm management problem subject to credit constraints and partial investment irreversibilities. We essentially took the problem from Khan and Thomas (2011), who embed it in a general equilibrium model to study the cyclical implications of credit market imperfections.

The Bellman equation for the problem is:

$$\begin{aligned} v(k, b, z) &= \max_{d, k', b'} \{d + \beta \mathbb{E}v(k', b', z)\} \\ \text{s.t.:} \quad &0 \leq d \leq zk^\alpha - \phi(k, k') - b + \beta b' \\ &b' \leq \theta k \\ &k' \geq 0 \end{aligned}$$

Here, k is capital, b is debt, z is productivity, d is dividends, zk^α is production, and β is the inverse of the gross interest rate. The first constraints are budget/limited liability constraints,

the second constraint is a Kiyotaki-Moore (1997) credit constraint with tightness parameter $\theta \geq 0$, and ϕ is an Abel-Eberly (1996) investment cost function given by:

$$\phi(k, k') = \begin{cases} k' - (1 - \delta)k & \text{if } k' \geq (1 - \delta)k \\ \gamma(k' - (1 - \delta)k) & \text{if } k' < (1 - \delta)k \end{cases}$$

where $\gamma \in [0, 1]$ is the price at which uninstalled capital can be sold on the market and δ is the depreciation rate. The baseline parameters are: $\beta = 0.96$, $\alpha = 0.27$, $\delta = 0.065$, $\theta = 1.28$, $\gamma = 0.95$, and z follows a 7 state Tauchen (1986) discretization of $\log(z') = 0.653 \times \log(z) + \eta$, $\eta \sim N(0, 0.135^2)$.

A subtle but important feature of this problem is that it is not appropriate to take $\mathbb{R}_+ \times \mathbb{R}$ to be the state space for (k, b) because the constraint set is empty—meaning the firm is insolvent—if b is too high relative to k . We therefore use instead:

$$\{(k, b) \in \mathbb{R}_+ \times \mathbb{R} : b \leq b_{\max}(k)\}$$

where $b_{\max} : \mathbb{R}_+ \rightarrow \mathbb{R}$ solves the functional equation:

$$b_{\max}(k) = \max_{k', b'} \{z_{\min} k^\alpha - \phi(k, k') + \beta b' : k' \geq 0, b' \leq \theta k, b' \leq b_{\max}(k')\} \quad (9)$$

with z_{\min} being the minimum value of z . This ensures that the set of feasible controls is non-empty at any given state. It is straightforward to show using standard contraction mapping arguments that b_{\max} is uniquely determined and concave, so our state space is well defined and convex.

This problem has several characteristics that make it non-trivial to solve using standard methods: (i) there are multiple constraints that bind only occasionally; (ii) there is a kink in ϕ which makes the problem non-differentiable; (iii) there are two continuous state variables; and (iv) the state space is non-rectangular. Properties (i)-(ii) pose a challenge for methods that exploit first order conditions, while properties (iii)-(iv) pose a challenge for many methods based on value iteration. Property (ii) also makes it difficult to evaluate the accuracy of the solution using standard metrics such as Euler equation errors.

Our theoretical analysis indicates that none of these characteristics are problematic for our method, however. Since the theorems do not depend in any way on the pattern of binding constraints, the smoothness of model primitives, or the dimensionality and rectangularity of the state space, our method should, at least in theory, be able to solve the problem as precisely as desired and provide computable error bounds on the solution despite these characteristics. We are not aware of other methods that can accomplish both tasks for

problems of this kind.

5.2 Code and computing environment

To evaluate how well our method can handle the problem in practice, we implemented a version of it which uses the linear programming approach from section 4.1 to compute the value functions, formula (8) to compute the policy function, and a straightforward adaptation of \mathbb{T}^L to handle (9) (which produces a polytope approximation of the theoretical state space within which the constraint set is guaranteed to be non-empty). The code, which is written in Fortran and uses ILOG CPLEX for linear programming and the sparse BLAS from Intel's MKL to handle the modified policy function iterations, is available for download at:

<http://www.ssc.wisc.edu/~kfukushi/papers.htm>

<http://sites.google.com/site/yuichirowaki/research>

We obtained the results below by compiling this code using the Intel Fortran compiler and running the executable on a desktop equipped with a 3.10 GHz Intel Core i5-2400 processor and 4 GB of RAM.

5.3 Accuracy and speed

Table 1 summarizes the performance of our method for several parameter configurations and grid sizes.

The second column reports an estimate of $\|v^U - v^L\|$, which we computed by simulating the solution for 50,000 periods and then calculating the maximum value of $|v^U(k_t, b_t, z_t) - v^L(k_t, b_t, z_t)|$ across all (k_t, b_t, z_t) realizations. This serves as an accuracy measure of both the value function and the policy function (cf. Theorem 5). In the table we express this quantity as a fraction of average firm value, so 1.0e-3 means 0.1% of average firm value, 1.0e-4 means 0.01% of average firm value, and so on. To put these numbers into perspective, we note that the coefficient of variation of firm value was about 3.5%. The numbers here indicate that the method was able to solve the problem reasonably accurately with moderate sized grids and that doubling the grid size reduced the errors by about 50% in most cases.

The third and fourth columns report how many seconds it took to solve the model without and with 200 modified policy function iterations, respectively. In each case, the computation time appears to grow faster than linearly but slightly slower than quadratically with the grid size, which is consistent with the typical behavior of the simplex method. The numbers also highlight the significant speed gains we obtained from modified policy function iterations, which allowed us to solve the model quite accurately in a matter of seconds. In each case,

Grid size	Error bound $\max v^U(t) - v^L(t) $	Time without modified policy iterations (seconds)	Time with modified policy iterations (seconds)
A. Baseline parameters			
100	2.8e-3	57.8	1.4
200	1.0e-3	162.8	3.8
400	4.8e-4	535.2	13.4
800	2.0e-4	1909.8	47.1
B. Low discounting ($\beta = 0.99$)			
100	1.7e-3	182.3	1.6
200	5.8e-4	496.1	4.5
400	1.8e-4	1685.0	15.4
800	1.5e-4	6407.4	61.5
C. High curvature in production function ($\alpha = 0.15$)			
100	2.5e-3	59.3	1.3
200	7.9e-4	155.8	3.6
400	1.8e-4	508.1	12.7
800	2.0e-4	1961.4	46.0
D. High productivity risk ($\text{Var}(\eta) = 1.5 \times 0.135^2$)			
100	5.0e-3	61.0	1.4
200	2.1e-3	162.7	3.9
400	8.0e-4	527.6	13.5
800	3.7e-4	1938.5	52.3
E. Irreversible investment ($\gamma = 0$)			
100	3.4e-3	64.3	1.7
200	1.2e-3	167.9	3.6
400	3.8e-4	538.9	13.4
800	2.6e-4	1921.6	47.0
F. Tight credit limit ($\theta = 0.5$)			
100	2.3e-3	53.3	1.3
200	6.3e-4	144.4	3.2
400	2.2e-4	483.9	11.4
800	1.6e-4	1757.0	42.7

Table 1: Performance benchmarks.

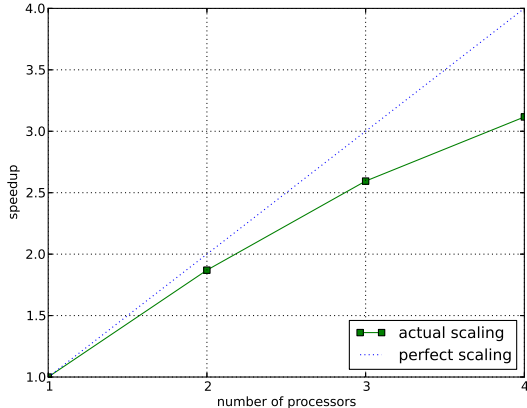


Figure 2: Parallel scaling.

about 1/2 of the time was spent on computing v^L and the remaining 1/2 on computing v^U . Hence the time requirements would have been about 1/2 of the reported values if (a) we had computed v^L and v^U in parallel, or (b) we had computed only v^L (as we would if we were not interested in obtaining the error bounds).

The table also shows that the performance of our method was more or less consistent across the parameter configurations we considered. There is, however, a noticeable speed reduction when the discount factor β is increased to 0.99, which is as expected given that our algorithm uses fixed point iterations on β -contractions, as well as an accuracy loss when the variance of productivity shocks z is increased by 50%.

In addition to this we also tested how well our method parallelizes when we distribute the maximization problems across a number of separate processors. We were able to obtain speedups of about 80-95% of what one should get under perfect scaling; figure 2 plots the results for the configuration reported in the first row of table 1.

5.4 Comparison with alternative methods

We next compare the performance of our method with two standard alternatives: (i) a pure discretization method (which simply replaces the state space with one that is finite), and (ii) a fitted value iteration method that uses bilinear interpolations to represent the value function and grid search for maximization. We focus here on the lower approximation part of our method only, as its upper approximation part has no counterpart in the alternative solution methods we consider.

To make this comparison we need a measure of solution accuracy that is comparable across different solution methods. Euler equation errors are often used in the literature

Grid size for state variables	Grid size for control variables	Value function error $\max v(t) - \hat{v}^U(t) $	Time with modified policy iterations (seconds)
A. Polyhedral approximation, lower approximation part only			
100	n/a	9.2e-4	0.7
200	n/a	6.6e-4	2.0
400	n/a	1.6e-4	7.9
800	n/a	9.4e-5	24.8
B. Discretization			
900	900	1.3e-2	0.3
2500	2500	7.1e-3	1.8
4900	4900	5.1e-3	6.6
10000	10000	3.4e-3	25.9
40000	40000	1.8e-3	423.5
C. Fitted value iteration with bilinear approximations and grid search			
900	4500	6.1e-4	4.6
2500	12500	2.6e-4	34.8
4900	24500	1.9e-4	140.2
10000	50000	1.5e-4	591.8
40000	200000	9.0e-5	9487.7

Table 2: Comparison with alternative methods.

for this purpose, but those are unavailable here due to the problem’s non-differentiability. We therefore look instead at “value function errors,” defined as $\|v - \hat{v}^U\|$, where v is the computed value function and \hat{v}^U is a tight upper bound on the true value function V , which we can obtain using our polyhedral approximation method with a fine grid. For the solution methods we consider we always have $v \leq V \leq \hat{v}^U$, so $\|v - \hat{v}^U\|$ is an upper bound on $\|v - V\|$ which is in principle computable.

Table 2 shows the results under the baseline parametrization. The first two columns list the grid size(s). The third column shows the value function errors $\|v - \hat{v}^U\|$, estimated using a 50,000 period simulation and expressed as a fraction of average firm value as in the previous subsection. The function \hat{v}^U we use here was computed using 1600 grid points and approximately satisfies $\|\hat{v}^U - V\| < 0.009\%$ of average firm value. The fourth column reports the computation time in seconds, with each method accelerated using 200 modified policy function iterations.¹ Overall, our method required significantly fewer grid points than its alternatives to attain a given level of accuracy, and, in part because of this, it was able to

¹In comparing the numbers in panel A with those in table 1, we note that (a) the value function errors reported in panel A are smaller than the error bounds reported in table 1 because the former is an estimate of $\|v^L - \hat{v}^U\|$, the latter is an estimate of $\|v^U - v^L\|$, and $v^L \leq V \leq \hat{v}^U < v^U$; and (b) the computations in panel A run faster than those in table 1 because here we are focusing on the lower approximation part of our method only.

deliver substantial speed gains, especially when high accuracy was requested.²

5.5 Solution characteristics

To conclude our discussion we indicate what the computed solution looks like. Figure 3 plots the policy functions. Each row here corresponds to a particular productivity level z : the top row is for low z (43% below median), the middle row is for median z , and the bottom row is for high z (43% above median). The left column then plots the policy for next period capital k' , the middle column plots the sign function of optimal gross investment $\text{sgn}(k' - (1 - \delta)k)$ (which equals 1 when investment is positive, 0 when investment is zero, and -1 when investment is negative), and the right column plots the policy for next period debt b' .

According to the left and middle columns, the firm typically: (i) invests (resp. disinvests) when k is sufficiently low (resp. high) to maintain a “target” level of capital; (ii) makes zero investments for intermediate levels of k ; (iii) sharply disinvests when it is highly leveraged (i.e., b is high relative to k); and (iv) is more likely to invest (resp. disinvest) when z is high (resp. low). The policy for k' displays clear kinks at the boundaries of these cases.

The right column indicates that optimal next period debt b' is typically increasing in current capital k . It turns out that the credit constraint $b' \leq \theta k$ is binding at most places, which explains this and the function’s apparent linearity. An exception arises however when productivity z is low and current capital k is high. In such states the firm chooses to deleverage and leave the credit constraint slack ($b' < \theta k$). The policy function again displays a kink at the boundary where this happens.

Figure 4 shows these features in action by plotting a particular simulated history. The left panel plots productivity z_t , the middle panel plots gross investment $k_{t+1} - (1 - \delta)k_t$, and the right panel plots next period debt b_{t+1} along with the credit limit θk_t . The figure shows that the firm usually chooses either positive or zero investment, and that the credit constraint $b_{t+1} \leq \theta k_t$ usually binds. Exceptions arise however between periods $t = 20$ and $t = 30$ when productivity declines sharply. During this “crisis” period the firm chooses to disinvest, although only by a relatively small amount, and to rapidly deleverage, rendering the credit constraint slack. The relative strength of the latter effect compared to the former is a natural consequence of the partial irreversibility built into the investment cost function ϕ .

²An obvious but important caveat here is that benchmark results of this sort inevitably depend not only on the relative merits of the methods but also on how efficiently they are implemented. We have tried to limit the influence of this problem here by employing any optimization tricks we know of to speed up the two alternative methods we consider (the most important being the extensive use of lookup tables) and by using similar grids (essentially equally spaced in each dimension) for each solution method.

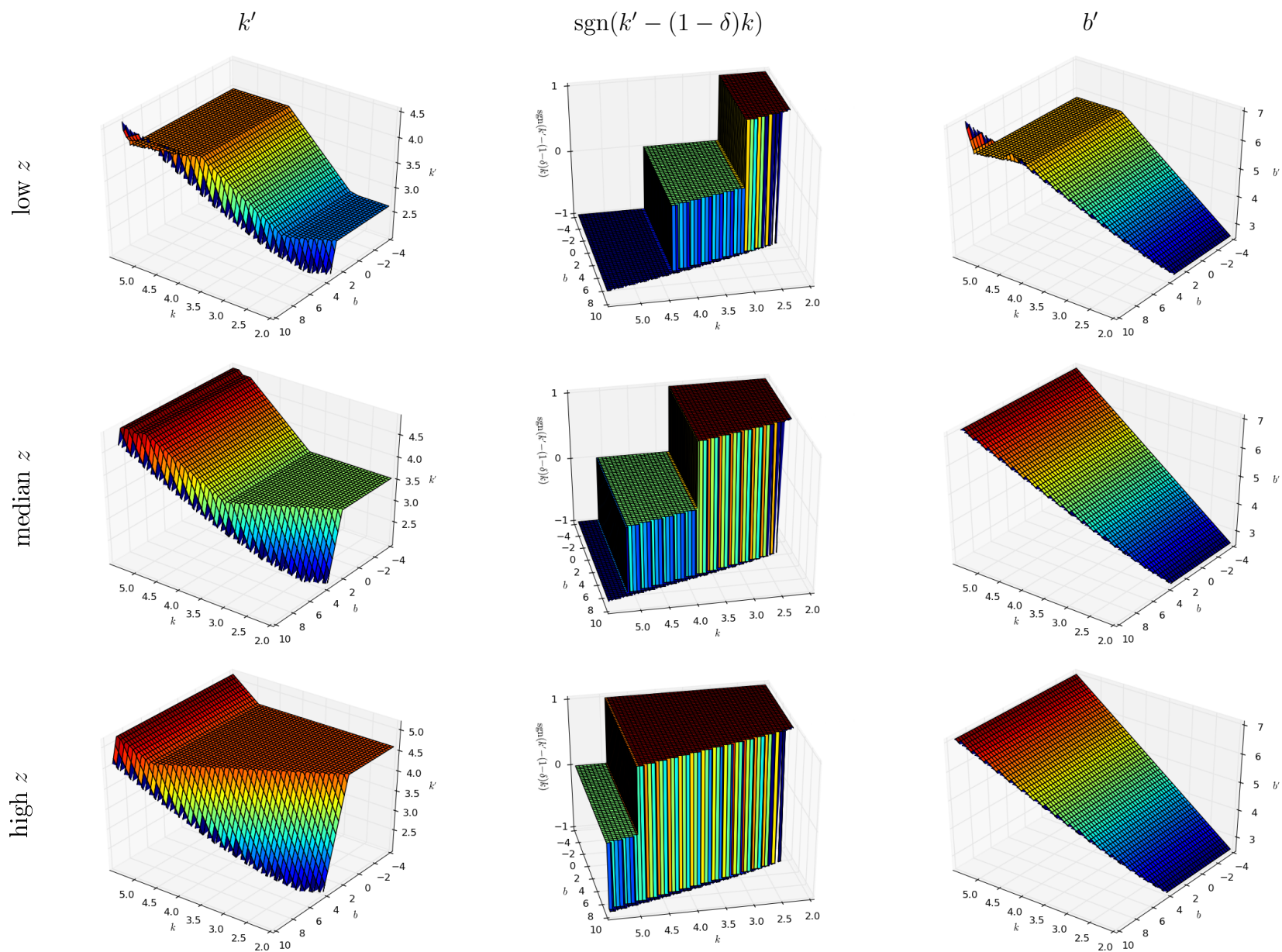


Figure 3: Policy functions for selected productivity values z .

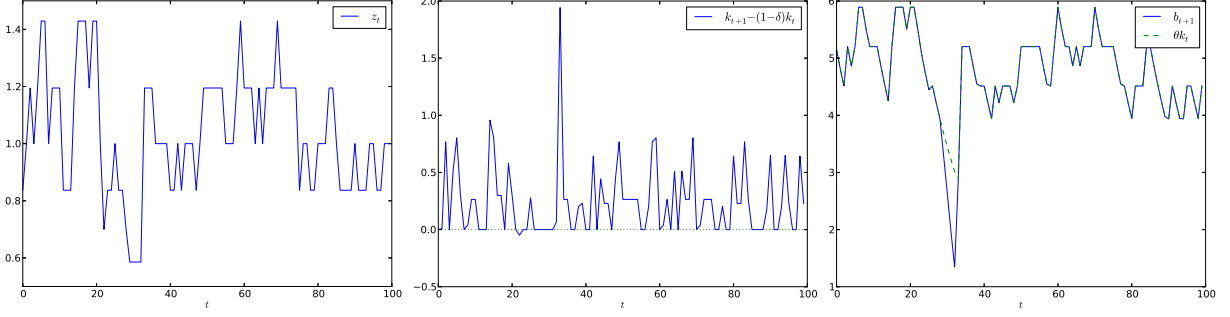


Figure 4: A simulated history.

A Proofs

A.1 Proof of Lemma 1L

(i) Let f be concave and let $\bar{S} \subset \hat{S}$. Fix $s \in S$. Let μ solve the maximization problem in (2). Then:

$$\mathbb{L}_{\hat{S}}f(s) = \sum_{\hat{s} \in \hat{S}} \mu(\hat{s})f(\hat{s}) \leq f\left(\sum_{\hat{s} \in \hat{S}} \mu(\hat{s})\hat{s}\right) = f(s)$$

where the weak inequality follows from the concavity of f . Next let $\bar{\mu} \in M(s, \bar{S})$ solve the problem in (2) for $\mathbb{L}_{\bar{S}}f(s)$. Define $\hat{\mu} \in M(s, \hat{S})$ by setting $\hat{\mu}(\tilde{s}) = \bar{\mu}(\tilde{s})$ if $\tilde{s} \in \bar{S}$ and $\hat{\mu}(\tilde{s}) = 0$ otherwise. Then

$$\mathbb{L}_{\bar{S}}f(s) = \sum_{\bar{s} \in \bar{S}} \bar{\mu}(\bar{s})f(\bar{s}) = \sum_{\hat{s} \in \hat{S}} \hat{\mu}(\hat{s})f(\hat{s}) \leq \max_{\mu \in M(s, \hat{S})} \sum_{\hat{s} \in \hat{S}} \mu(\hat{s})f(\hat{s}) = \mathbb{L}_{\hat{S}}f(s).$$

Now let f be concave and continuous and fix $\epsilon > 0$. Define the following three subsets of \mathbb{R}^{l+1} :

$$E = S \times [\min f(S) - \epsilon, \max f(S)]$$

$$A = \{(s, t) \in E : t \leq f(s) - \epsilon\}$$

$$C = \{(s, t) \in E : t \leq f(s)\}$$

We claim that there is a polytope $O \subset \mathbb{R}^{l+1}$ such that $A \subset O \subset C$. The construction goes as follows. First, because f is continuous and $\epsilon > 0$, we can for each $a \in A$ choose an open rectangle $R(a) \subset \mathbb{R}^{l+1}$ such that $a \in R(a)$ and $R(a) \cap E \subset C$. And because $\{R(a) : a \in A\}$ is an open cover of A which is compact, there exists a finite subset of A , say A_0 , such that $A \subset \cup_{a_0 \in A_0} R(a_0)$. Set $O = \text{co}(\cup_{a_0 \in A_0} R(a_0) \cap E)$.

Let \bar{S} denote the set of s -coordinates of the vertices of O . Then let for each $\bar{s} \in \bar{S}$:

$$\bar{T}(\bar{s}) = \{\bar{t} \in \mathbb{R} : (\bar{s}, \bar{t}) \text{ is a vertex of } O\}, \quad \tau(\bar{s}) = \max \bar{T}(\bar{s}).$$

For any $(s, t) \in O$ there exists a probability distribution ν on the set of vertices of O with mean (s, t) , so:

$$t = \sum_{\bar{s} \in \bar{S}} \sum_{\bar{t} \in \bar{T}(\bar{s})} \nu(\bar{s}, \bar{t}) \bar{t} \leq \sum_{\bar{s} \in \bar{S}} \sum_{\bar{t} \in \bar{T}(\bar{s})} \nu(\bar{s}, \bar{t}) \tau(\bar{s}) \leq \max_{\mu \in M(s, \bar{S})} \sum_{\bar{s} \in \bar{S}} \mu(\bar{s}) \tau(\bar{s}).$$

Also from $O \subset C$ we know that $\tau(\bar{s}) \leq f(\bar{s})$ for any $\bar{s} \in \bar{S}$.

We therefore have for each s :

$$\begin{aligned} f(s) - \epsilon &= \max\{t \in \mathbb{R} : (s, t) \in A\} \leq \max\{t \in \mathbb{R} : (s, t) \in O\} \\ &\leq \max_{\mu \in M(s, \bar{S})} \sum_{\bar{s} \in \bar{S}} \mu(\bar{s}) \tau(\bar{s}) \leq \max_{\mu \in M(s, \bar{S})} \sum_{\bar{s} \in \bar{S}} \mu(\bar{s}) f(\bar{s}) = \mathbb{L}_{\bar{S}} f(s). \end{aligned}$$

(ii) Suppose $f \leq f'$. Then for each s , the objective in (2) for $\mathbb{L}f(s)$ is no greater than that for $\mathbb{L}f'(s)$ at any given μ . Hence $\mathbb{L}f \leq \mathbb{L}f'$.

(iii) Using (2) we have for any s :

$$\mathbb{L}(f + a)(s) = \max_{\mu \in M(s, \hat{S})} \sum_{\hat{s} \in \hat{S}} \mu(\hat{s}) (f(\hat{s}) + a) = \left(\max_{\mu \in M(s, \hat{S})} \sum_{\hat{s} \in \hat{S}} \mu(\hat{s}) f(\hat{s}) \right) + a = \mathbb{L}f(s) + a.$$

(iv) Immediate from (2).

A.2 Proof of Lemma 1U

(i) Let f be concave and let $\bar{D} \subset \hat{D}$. Fix $s \in S$. We have:

$$\mathbb{U}_{\hat{D}} f(s) = \min_{\hat{d} \in \hat{D}} \{\hat{d} \cdot s - f^*(\hat{d})\} \geq \inf_{d \in \bar{D}} \{d \cdot s - f^*(d)\} = f^{**}(s) \geq f(s).$$

Also from $\bar{D} \subset \hat{D}$ we have:

$$\mathbb{U}_{\bar{D}} f(s) = \min_{\bar{d} \in \bar{D}} \{\bar{d} \cdot s - f^*(\bar{d})\} \leq \min_{\bar{d} \in \bar{D}} \{\bar{d} \cdot s - f^*(\bar{d})\} = \mathbb{U}_{\bar{D}} f(s).$$

Now let f be concave and continuous and fix $\epsilon > 0$. Define:

$$\begin{aligned} E &= S \times [\min f(S), \max f(S) + \epsilon] \\ A &= \{(s, t) \in E : t \leq f(s)\} \\ C &= \{(s, t) \in E : t \leq f(s) + \epsilon\} \end{aligned}$$

The exact same argument as in the proof of Lemma 1L (i) implies that there is a polytope $O \subset \mathbb{R}^{l+1}$ that satisfies $A \subset O \subset C$.

We next partition the set of normals of O into \bar{D} , \bar{D}_- , and \bar{D}_0 so that $(s, t) \in O$ if and only if:

$$\begin{aligned} t &\leq \bar{d} \cdot s + \psi(\bar{d}), \quad \forall \bar{d} \in \bar{D} \\ -t &\leq \bar{d} \cdot s + \psi(\bar{d}), \quad \forall \bar{d} \in \bar{D}_- \\ 0 &\leq \bar{d} \cdot s + \psi(\bar{d}), \quad \forall \bar{d} \in \bar{D}_0 \end{aligned}$$

Note that for $\bar{d} \in \bar{D}$ we can assume without loss of generality:

$$\psi(\bar{d}) = \max_{(s,t) \in O} \{t - \bar{d} \cdot s\} \geq \max_{(s,t) \in A} \{t - \bar{d} \cdot s\} = \max_{s \in S} \{f(s) - \bar{d} \cdot s\} = -f^*(\bar{d}).$$

We therefore have for each s :

$$\begin{aligned} f(s) + \epsilon &= \max\{t \in \mathbb{R} : (s, t) \in C\} \geq \max\{t \in \mathbb{R} : (s, t) \in O\} \\ &= \min_{\bar{d} \in \bar{D}} \{\bar{d} \cdot s + \psi(\bar{d})\} \geq \min_{\bar{d} \in \bar{D}} \{\bar{d} \cdot s - f^*(\bar{d})\} = \mathbb{U}_D f(s). \end{aligned}$$

(ii) If $f \leq f'$, we have $f^* \geq f'^*$ and hence $\mathbb{U}f \leq \mathbb{U}f'$ from (3).

(iii) For any $d \in D$:

$$(f + a)^*(d) = f^*(d) - a.$$

So for any $s \in S$ we have

$$\mathbb{U}(f + a)(s) = \min_{\hat{d} \in \hat{D}} \{\hat{d} \cdot s - (f + a)^*(\hat{d})\} = \min_{\hat{d} \in \hat{D}} \{\hat{d} \cdot s - f^*(\hat{d})\} + a = \mathbb{U}f(s) + a.$$

(iv) Let $f \equiv 0$ and pick any $s \in S$. From (i) we have $\mathbb{U}f(s) \geq f(s) = 0$. We also have $f^*(0) = 0$, which together with $0 \in \hat{D}$ implies

$$\mathbb{U}f(s) = \min_{\hat{d} \in \hat{D}} \{\hat{d} \cdot s - f^*(\hat{d})\} \leq 0 \cdot s - f^*(0) = 0.$$

Hence $\mathbb{U}f(s) = 0$.

A.3 Proof of Theorem 2L

If $v \in B(X \times Z)$, then from (4) and Lemma 1L (ii), (iii), and (iv) we have:

$$r_{\min} + \beta \inf v(X \times Z) \leq \mathbb{T}^L v \leq r_{\max} + \beta \sup v(X \times Z)$$

so $\mathbb{T}^L v \in B(X \times Z)$. Also from (4) and Lemma 1L (ii) we know that \mathbb{T}^L is monotone. And from (4) and Lemma 1L (iii) we know that if $v \in B(X \times Z)$ and $a \in \mathbb{R}_{++}$ then $\mathbb{T}^L(v + a) = \mathbb{T}^L v + \beta a$. Blackwell's sufficient conditions hold, so \mathbb{T}^L is a β -contraction.

In proceeding, we observe that from (1), (4), Lemma 1L (i), and the concavity of V we have $V = \mathbb{T}V \geq \mathbb{T}^L V$.

Now let $v_N^L = (\mathbb{T}^L)^N V$ for $N \in \mathbb{N}$. We just observed that $v_1^L = \mathbb{T}^L V \leq V = v_0^L$. And if $v_N^L \leq v_{N-1}^L$ then $v_{N+1}^L = \mathbb{T}^L v_N^L \leq \mathbb{T}^L v_{N-1}^L = v_N^L$ by the monotonicity of \mathbb{T}^L . It follows by induction that $(v_N^L)_{N \in \mathbb{N}}$ is monotonically decreasing. The contraction property of \mathbb{T}^L implies that $\|v_N^L - V^L\| \rightarrow 0$ as $N \rightarrow \infty$, so we have $V^L \leq V$.

Finally, let $\epsilon > 0$ be given. Since V is continuous and concave, we can use Lemma 1L (i) to choose \bar{X} so that for each $z \in Z$:

$$\mathbb{L}_{\bar{X}} V(\cdot, z) \geq V(\cdot, z) - (1 - \beta)\epsilon/\beta.$$

Let $\bar{X} \subset \hat{X}$. We then have for each $(x, z) \in X \times Z$:

$$\begin{aligned} \mathbb{T}^L V(x, z) &= \max_{y \in \Gamma(x, z)} \{r(x, y, z) + \beta \mathbb{E} \mathbb{L}_{\bar{X}} V(\cdot, z)(y)\} \\ &\geq \max_{y \in \Gamma(x, z)} \{r(x, y, z) + \beta \mathbb{E} \mathbb{L}_{\bar{X}} V(\cdot, z)(y)\} \\ &\geq \max_{y \in \Gamma(x, z)} \{r(x, y, z) + \beta \mathbb{E} V(y, z)\} - (1 - \beta)\epsilon \\ &= \mathbb{T}V(x, z) - (1 - \beta)\epsilon \\ &= V(x, z) - (1 - \beta)\epsilon. \end{aligned}$$

From this and $V \geq \mathbb{T}^L V$ we have $\|V - \mathbb{T}^L V\| \leq (1 - \beta)\epsilon$. Combining this with

$$\|V - V^L\| = \|V - \mathbb{T}^L V\| + \|\mathbb{T}^L V - \mathbb{T}^L V^L\| \leq \|V - \mathbb{T}^L V\| + \beta \|V - V^L\|$$

we obtain:

$$\|V - V^L\| \leq \frac{1}{1 - \beta} \|V - \mathbb{T}^L V\| \leq \epsilon.$$

A.4 Proof of Theorem 2U

Essentially identical to that of Theorem 2L.

A.5 Proof of Theorem 3L

We can rewrite (4) as:

$$\mathbb{T}^L v(\hat{x}, z) = \max_{y \in \Gamma(\hat{x}, z)} \left\{ r(\hat{x}, y, z) + \beta \sum_{z' \in Z} \left(\max_{\mu(\cdot, z') \in M(y, \hat{X})} \sum_{\hat{y} \in \hat{X}} \mu(\hat{y}, z') v(\hat{y}, z') \right) \pi(z'|z) \right\}$$

which is equivalent to (6).

A.6 Proof of Theorem 3U

Let us first rewrite (5) as:

$$\mathbb{T}^U v(x, z) = \max_{y \in \Gamma(x, z)} \max_{\tau \in \mathbb{R}^{|Z|}} \left\{ r(x, y, z) + \beta \sum_{z' \in Z} \tau(z') \pi(z'|z) : \tau(z') \leq \mathbb{U}_{\hat{P}} v(y, z'), \forall z' \in Z \right\}$$

and note that

$$\tau(z') \leq \mathbb{U}_{\hat{P}} v(y, z'), \forall z' \in Z \iff \tau(z') \leq \hat{q} \cdot y - v^*(\hat{q}, z'), \forall (\hat{q}, z') \in \hat{P} \times Z.$$

Substituting these into the right hand side of the definition for $(\mathbb{T}^U v)^*$, namely

$$(\mathbb{T}^U v)^*(\hat{p}, z) = \inf_{x \in X} \{ \hat{p} \cdot x - \mathbb{T}^U v(x, z) \},$$

we obtain:

$$(\mathbb{T}^U v)^*(\hat{p}, z) = \inf_{x, y, \tau \in X \times X \times \mathbb{R}^{|Z|}} \left\{ \hat{p} \cdot x - r(x, y, z) - \beta \sum_{z' \in Z} \tau(z') \pi(z'|z) : \right. \\ \left. \tau(z') \leq \hat{q} \cdot y - v^*(\hat{q}, z'), \forall (\hat{q}, z') \in \hat{P} \times Z; h(x, y, z) \geq 0 \right\}.$$

By strong duality we may rewrite the right hand side as:

$$\sup_{\lambda \in \mathbb{R}_+^{|\hat{P}| \times |Z|}} \inf_{x, y, \tau \in X \times X \times \mathbb{R}^{|Z|}} \left\{ \hat{p} \cdot x - r(x, y, z) - \beta \sum_{z' \in Z} \tau(z') \pi(z'|z) \right. \\ \left. + \sum_{\hat{q} \in \hat{P}} \sum_{z' \in Z} \lambda(\hat{q}, z') (\tau(z') - \hat{q} \cdot y + v^*(\hat{q}, z')) : h(x, y, z) \geq 0 \right\}.$$

By the linearity in τ , this equals:

$$\sup_{\lambda \in \mathbb{R}_+^{|\hat{P}| \times |Z|}} \inf_{x, y \in X \times X} \left\{ \hat{p} \cdot x - r(x, y, z) + \sum_{\hat{q} \in \hat{P}} \sum_{z' \in Z} \lambda(\hat{q}, z') (-\hat{q} \cdot y + v^*(\hat{q}, z')) : \right. \\ \left. h(x, y, z) \geq 0; \sum_{\hat{q} \in \hat{P}} \lambda(\hat{q}, z') = \beta \pi(z'|z), \forall z' \in Z \right\}$$

which can be rewritten as:

$$\sup_{(q, \lambda) \in P \times \mathbb{R}_+^{|\hat{P}| \times |Z|}} \inf_{x, y \in X \times X} \left\{ \hat{p} \cdot x + q \cdot y - r(x, y, z) + \sum_{\hat{q} \in \hat{P}} \sum_{z' \in Z} \lambda(\hat{q}, z') v^*(\hat{q}, z') : \right. \\ \left. h(x, y, z) \geq 0; \sum_{\hat{q} \in \hat{P}} \lambda(\hat{q}, z') = \beta \pi(z'|z), \forall z' \in Z; \sum_{\hat{q} \in \hat{P}} \sum_{z' \in Z} \lambda(\hat{q}, z') \hat{q} = -q \right\}.$$

Solving out for the inner minimization over (x, y) and using the compactness of the constraint set for (q, λ) , we obtain (7).

A.7 Proof of Theorem 4L

Let $(v_N^L)_{N \in \mathbb{N}}$ satisfy the hypotheses. We know from the contraction property of \mathbb{T}^L that $\|v_N^L - V^L\| \rightarrow 0$ as $N \rightarrow \infty$, so it is enough to show that $v_N^L \leq v_{N+1}^L$ for all N . This holds for $N = 0$ by assumption. And if it holds for N , we can apply \mathbb{T}^L to both sides and use its monotonicity to get $v_{N+1}^L = \mathbb{T}^L v_N^L \leq \mathbb{T}^L v_{N+1}^L = v_{N+2}^L$. The result follows by induction.

Suppose $v_0^L \equiv r_{\min}/(1 - \beta)$. Then we have from (4) and Lemma 1L (ii), (iii), and (iv):

$$\mathbb{T}^L v_0^L(x, z) \geq r_{\min} + \beta \frac{r_{\min}}{1 - \beta} = \frac{r_{\min}}{1 - \beta} = v_0^L(x, z).$$

A.8 Proof of Theorem 4U

Essentially identical to that of Theorem 4L.

A.9 Proof of Theorem 5

Let \mathbb{G} map $v : X \times Z \rightarrow \mathbb{R}$ to $\mathbb{G}v : X \times Z \rightarrow \mathbb{R}$ as:

$$\mathbb{G}v(x, z) = r(x, g(x, z), z) + \beta \mathbb{E}v(g(x, z), z).$$

Standard arguments imply that \mathbb{G} is a monotone β -contraction on $B(X \times Z)$ and that its unique fixed point V_g is the value of policy g .

From $v^L \leq \mathbb{T}^L v^L \leq \mathbb{T} v^L = \mathbb{G}v^L$ and the monotone contraction property of \mathbb{G} we have $v^L \leq \mathbb{G}^N v^L \uparrow V_g$ as $N \rightarrow \infty$. Also from $v^U \geq \mathbb{T}^U v^U \geq \mathbb{T} v^U \geq \mathbb{G}v^U$ and the monotone contraction property of \mathbb{G} we have $v^U \geq \mathbb{G}^N v^U \downarrow V_g$ as $N \rightarrow \infty$. Hence $v^L \leq V_g \leq v^U$. From this and $v^L \leq V \leq v^U$ we have $\|V_g - V\| \leq \|v^U - v^L\|$. Since $g(x, z) \in \Gamma(x, z)$ for each $(x, z) \in X \times Z$ by definition, the result follows.

References

- Abel, A. B., and J. C. Eberly (1996): “Optimal Investment with Costly Reversibility,” *Review of Economic Studies*, 63(4), 581–593.
- Bertsekas, D. P., and H. Yu (2011): “A Unifying Polyhedral Approximation Framework for Convex Optimization,” *SIAM Journal on Optimization*, 21(1), 333–360.
- Chow, C.-S., and J. Tsitsiklis (1991): “An Optimal One-way Multigrid Algorithm for Discrete-time Stochastic Control,” *IEEE Transactions on Automatic Control*, 36(8), 898–914.
- Gordon, G. J. (1995): “Stable Function Approximation in Dynamic Programming,” in *Proceedings of the Twelfth International Conference on Machine Learning*, ed. by A. Prieditis, and S. J. Russell, pp. 261–268.
- Judd, K. L. (1998): *Numerical Methods in Economics*. MIT Press, Cambridge, MA.
- Judd, K. L., and A. Solnick (1994): “Numerical Dynamic Programming with Shape-Preserving Splines,” Working paper, Hoover Institution and Stanford University.

- Judd, K. L., S. Yeltekin, and J. Conklin (2003): “Computing Supergame Equilibria,” *Econometrica*, 71(4), 1239–1254.
- Khan, A., and J. Thomas (2011): “Credit Shocks and Aggregate Fluctuations in an Economy with Production Heterogeneity,” NBER Working Paper 17311, National Bureau of Economic Research.
- Kiyotaki, N., and J. Moore (1997): “Credit Cycles,” *Journal of Political Economy*, 105(2), 211–248.
- Nishimura, K., and J. Stachurski (2009): “Equilibrium Storage with Multiple Commodities,” *Journal of Mathematical Economics*, 45(1-2), 80–96.
- Phelan, C., and R. M. Townsend (1991): “Computing Multi-Period, Information-Constrained Optima,” *Review of Economic Studies*, 58(5), 853–881.
- Rockafellar, R. T. (1970): *Convex Analysis*. Princeton University Press.
- Santos, M. S. (2000): “Accuracy of Numerical Solutions Using the Euler Equation Residuals,” *Econometrica*, 68(6), 1377–1402.
- Santos, M. S., and J. Vigo-Aguiar (1998): “Analysis of a Numerical Dynamic Programming Algorithm Applied to Economic Models,” *Econometrica*, 66(2), 409–426.
- Stachurski, J. (2008): “Continuous State Dynamic Programming via Nonexpansive Approximation,” *Computational Economics*, 31(2), 141–160.
- Stokey, N. L., R. E. Lucas, Jr, and E. C. Prescott (1989): *Recursive Methods in Economic Dynamics*. Harvard University Press, Cambridge.
- Tauchen, G. (1986): “Finite State Markov-Chain Approximations to Univariate and Vector Autoregressions,” *Economics Letters*, 20(2), 177–181.