AN OPTIMAL MULTIGRID ALGORITHM FOR DISCRETE–TIME STOCHASTIC CONTROL †

Chee–Seng Chow††
John N. Tsitsiklis††

Abstract
We consider the numerical solution of discrete-time, stationary, infinite horizon, discounted stochastic control problems, for the case where the state space is continuous and the problem is to be solved approximately, within a desired accuracy. After a discussion of problem discretization, we introduce a multigrid version of the successive approximation algorithm, and analyze its computational requirements as a function of the desired accuracy and of the discount factor. We show that the multigrid algorithm improves upon the complexity of its single-grid variant and is, in a certain sense, optimal. We also study the effects of a certain mixing (accessibility) condition on the problem’s complexity.

† Research supported by the NSF under Grant ECS–8552419, with matching funds from Bellcore and Dupont, and by the ARO under Grant DAAL03–86–K–0171. A preliminary version of this paper was presented at the 27th IEEE Conference on Decision and Control.

†† Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139.
1. INTRODUCTION AND SUMMARY

This paper deals with the computational aspects of continuous-state, discounted-cost Markov
Decision Problems (MDPs), as they arise in discrete-time stochastic control ([7], [8]) and is a
continuation of a research effort ([20], [21]) aimed at the understanding of the computational
complexity of control problems.

In a typical MDP, we are given a controlled discrete-time system that evolves in a state space
$S \subset \mathbb{R}^n$ and we are interested in computing a fixed point $J^*$ of the dynamic programming operator
$T$ (acting on a space of functions on the set $S$) defined by

$$(TJ)(x) = \inf_{u \in C} \left[ g(x, u) + \alpha \int_S J(y)P(y|x, u) \, dy \right], \quad \forall x \in S. \quad (1.1)$$

Here, $C \subset \mathbb{R}^m$ is the control space, $g(x, u)$ is the cost incurred if the current state is $x$ and
control $u$ is applied, $\alpha \in (0,1)$ is a discount factor, and $P(y|x, u)$ is a stochastic kernel that
specifies the probability distribution of the next state $y$, when the current state is $x$ and control
$u$ is applied. Then, $J^*(x)$ is interpreted as the value of the expected discounted cost, starting
from state $x$, and provided that the control actions are chosen optimally (see Section 2 for more
details). Unfortunately, even if the problem data (the functions $g$ and $P$) are given in closed
form, the equation $TJ^* = J^*$ does not usually admit closed form solutions and must be solved
numerically. This can be accomplished by discretizing the continuous problem to obtain an MDP
with finite state and control spaces. Then, the resulting discrete problem can be solved by means
of several algorithms such as successive approximation (value iteration), policy iteration, or linear
programming [5]. Furthermore, there are bounds available on how fine the discretization should be
in order to achieve a desired accuracy (see e.g. [27] and the references therein).

The computational requirements of continuous-state MDPs are substantial and for this reason,
past research has focused on the finite-state case (see e.g. [15] and the references therein). However,
the availability of more powerful computer hardware might make the solution of such problems
feasible, including real-time applications (e.g. in robotics [23]).

The main contribution of this paper is the introduction of a multigrid variant of the successive
approximation algorithm, together with a detailed analysis of its computational requirements. This
algorithm proceeds by solving the problem approximately on a coarse grid and by using the coarse-
grid solution as a starting point for the solution on a finer grid. Thus, most of the work takes place
on coarse grids with a complexity reduction resulting. In particular, our algorithm has optimal
complexity, in a certain sense to be made precise later.

A novelty in our complexity analysis is that we simultaneously consider the dependence on the
desired accuracy $\epsilon$ and on the discount factor $\alpha$ (as $\epsilon \downarrow 0$ and $\alpha \uparrow 1$). The dependence on $\alpha$ is
interesting for both theoretical and practical reasons. From the theoretical point of view, when $\alpha$
approaches 1, the problem converges in a certain sense to an "average cost" problem [5]. From the
practical point of view, if one discretizes a continuous-time discounted stochastic control problem,
the discount factor in the resulting discrete-time problem approaches 1 as the discretization step becomes finer. It will be shown that the dependence of the complexity on \( \alpha \) is significantly affected by the presence or absence of a certain accessibility (mixing) condition.

Multigrid methods have been studied extensively, primarily for the numerical solution of partial differential equations and have been found, both theoretically and experimentally, to offer substantial computational savings (see e.g. [9], [14]). In the context of stochastic control, multigrid methods have been independently introduced in [1,2] and [16]. (Also see [6] and the references therein for related works.) However, our work is different in a number of important respects to be discussed in Section 7.3.

Outline of the paper

The paper is organized as follows:

In Section 2, we introduce our notation and review some basic facts about monotone contraction operators. We state our assumptions and define the problem of interest.

In Section 3, we describe a discretization procedure related to the one introduced in [27], and quantify the resulting approximation error.

In Section 4, we introduce an "accessibility condition" which is a continuous-state formulation of a "scrambling-type" condition discussed in [13,24]. We show that the accessibility condition leads to faster convergence of successive approximation methods and to better discretization error bounds. We also show that if a continuous-state problem satisfies an accessibility condition, then this property is inherited by the discretized version of the problem.

In Section 5, we review some error bounds for the successive approximation algorithm, introduce our model of computation, and develop some estimates on the computational cost of a typical iteration.

In Section 6, we analyze the complexity of the classical (single-grid) successive approximation algorithm. The analysis in this (and the next section) is carried out twice: for general problems, as well as for problems satisfying the accessibility condition.

In Section 7, we introduce our multigrid version of the successive approximation algorithm and analyze its complexity. We compare our approach with the one introduced in [1,2] and [16]. We also discuss the optimality of our methods.

In Section 8, we consider the computation of a policy whose cost is within \( \epsilon \) of the optimal.

Finally, in Section 9, we discuss several extensions and generalizations of our results and suggest certain directions for future research.
2. MARKOV DECISION PROBLEMS

In this section, we give a precise definition of Markov Decision Problems (MDPs), and state our assumptions. We start by introducing some notation and with a review of some basic concepts.

2.1. Notation, Norms, and Operators

Let \( \{f_\alpha \}_{\alpha \in I} \) be a collection of real-valued functions, indexed by \( \alpha \in I \), defined on a set \( X \). We use \( \inf_{\alpha \in I} f_\alpha \) to denote the real-valued function defined by

\[
\inf_{\alpha \in I} f_\alpha (x) = \inf_{\alpha \in I} \{ f_\alpha (x) \}, \quad x \in X.
\]  

(2.1)

The notation \( \sup_{\alpha \in I} f_\alpha \) is defined similarly.

Let \( S \) be a Borel measurable subset of the Euclidean space \( \mathbb{R}^n \). We use \( \mathcal{B}(S) \) [respectively, \( \mathcal{C}(S) \)] to denote the space of all bounded Borel measurable (respectively, bounded continuous) functions on \( S \). When comparing two functions \( J, J' \in \mathcal{B}(S) \), we use the notation \( J \leq J' \) which is to be interpreted as \( J(x) \leq J'(x) \) for all \( x \in S \).

We view the Euclidean space \( \mathbb{R}^n \) as a normed vector space by endowing it with the sup-norm \( \| \cdot \|_\infty \). We will also use \( \| \cdot \|_\infty \) to denote the sup-norm on \( \mathcal{B}(S) \) which is defined by

\[
\| J \|_\infty = \sup_{x \in S} | J(x) |, \quad J \in \mathcal{B}(S).
\]  

(2.2)

It is well known that \( \mathcal{B}(S) \) is complete with respect to the sup-norm \( \| \cdot \|_\infty \) (see e.g. [3]) and is therefore a Banach Space. Similarly, \( \mathcal{C}(S) \) is also a Banach Space under the same norm (see e.g. [17]); hence, \( \mathcal{C}(S) \) is a closed subspace of \( \mathcal{B}(S) \). We define

\[
\| J \|_Q \overset{\text{def}}{=} \sup_{x \in S} J(x) - \inf_{x \in S} J(x), \quad J \in \mathcal{B}(S).
\]  

(2.3)

The function \( \| \cdot \|_Q \) is called the span norm in [13,24] and can be easily shown to satisfy the triangle inequality. (In fact, it is a quasi-norm.) It also satisfies

\[
\| J \|_Q \leq 2 \| J \|_\infty, \quad \forall J \in \mathcal{B}(S).
\]  

(2.4)

An operator \( A : \mathcal{B}(S) \mapsto \mathcal{B}(S) \) is called a monotone operator if \( J \leq J' \) implies \( AJ \leq AJ' \). Furthermore, if there exists some \( \alpha \in (0, 1) \) such that \( \| AJ - AJ' \|_\infty \leq \alpha \| J - J' \|_\infty \) for all \( J, J' \in \mathcal{B}(S) \), then \( A \) is called a contraction operator on \( \mathcal{B}(S) \), with contraction factor \( \alpha \). Operators that satisfy both properties are called monotone contraction operators ([12], [27]).

2.2. Specification of a Markov Decision Problem

An MDP is defined as follows. We are given state space \( S \subset \mathbb{R}^n \) on which a controlled stochastic process evolves, and a control space \( C \subset \mathbb{R}^m \) from which control actions will be chosen. We assume
that $S$ and $C$ are bounded and measurable and, without loss of generality, we can make the further assumption that $S \subseteq [0, 1]^n$ and $C = [0, 1]^m$. The dynamics of the system are described by a Borel measurable function $P : S \times S \times C \mapsto [0, \infty)$. In particular, $P(y|x, u)$ is to be interpreted as the probability density of the next state $y$ when the current state is $x$ and the control $u$ is applied.

We incorporate state-dependent constraints in our formulation. In particular, for each $x \in S$, we are given a nonempty set $U(x) \subseteq C$ of admissible controls. Let

$$\Gamma \overset{\text{def}}{=} \{(x, u) \mid x \in S \text{ and } u \in U(x)\}. \quad (2.5)$$

We assume that $\Gamma$ is the intersection of a closed subset of $\mathbb{R}^n \times \mathbb{R}^m$ with the set $S \times C$. That is, $\Gamma$ is closed with respect to the induced topology on $S \times C$.

If at some stage $k$, the state is $x$ and control $u$ is applied, then a cost $\alpha^k g(x, u)$ is incurred, where $g : S \times C \mapsto \mathbb{R}$ is a bounded measurable function, and $\alpha \in (0, 1)$ is the discount factor. A Markov Decision Problem is specified by the tuple $(S, C, \{U(x)\}_{x \in S}, P, g, \alpha)$.

2.3. Assumptions

We assume that there exists a constant $K \geq 1$ such that:

A.1: $|g(x, u) - g(x', u')| \leq K(\|x - x'\|_\infty + \|u - u'\|_\infty)$, for all $x, x' \in S$ and $u, u' \in C$.

A.2: $|P(y|x, u) - P(y'|x', u')| \leq K(\|y - y'\|_\infty + \|x - x'\|_\infty + \|u - u'\|_\infty)$, for all $x, x', y, y' \in S$ and $u, u' \in C$.

A.3: For any $x, x' \in S$ and any $u' \in U(x')$, there exists some $u \in U(x)$ such that $\|u - u'\|_\infty \leq K\|x - x'\|_\infty$.

A.4: $0 \leq P(y|x, u) \leq K$ and $\int_S P(y|x, u)dy = 1$, for all $x, y \in S$ and $u \in C$.

The first two assumptions state that $g$ and $P$ are Lipschitz continuous. The third is the same as an assumption used in [4], and is a continuity condition on the point-to-set mapping $x \mapsto U(x)$. The last assumption reflects the fact that $P(\cdot|x, u)$ is a probability density. Unless otherwise stated, Assumptions A.1 - A.4 will always be in effect. (Generalizations of Assumptions A.2 and A.4 are discussed in Section 9.) Under these assumptions, our MDP is a special case of the lower semi-continuous model studied in [7].

2.4. Policies and the Optimal Cost Function

Let

$$\Pi \overset{\text{def}}{=} \{\mu : S \mapsto C \mid \mu \text{ is Borel measurable and } \mu(x) \in U(x), \forall x \in S\}. \quad (2.6)$$

Let $\Pi^\infty$ be the set of all sequences $\pi = (\mu_0, \mu_1, \ldots)$ of elements of $\Pi$. Each element of $\Pi^\infty$ is called a policy and is interpreted as a prescription for choosing control actions as a function of time.

---

1. The choice of norm for $\mathbb{R}^n$, $\mathbb{R}^{n+m}$, and $\mathbb{R}^{2n+m}$ does not affect our complexity analysis because by norm equivalence [17] it only changes the Lipschitz constant by a constant factor; in contrast, the choice of norm for $\mathcal{B}(S)$ matters.
and of the current state. In particular if the state at time \( t \) is equal to some \( x \) and policy \( \pi \) is used, then control \( \mu_t(x) \) is applied. Once a particular policy is fixed, we can construct a Markov process \( \{x_t^\pi \mid t = 0, 1, \ldots \} \) by letting \( P(\cdot \mid x_t^\pi, \mu_t(x_t^\pi)) \) be the probability density function of \( x_{t+1}^\pi \), conditioned on \( x_t^\pi \).

For any policy \( \pi \in \Pi^\infty \), we define its cost \( J_\pi(x) \), as a function of the initial state, by letting

\[
J_\pi(x) = E \left\{ \sum_{t=0}^{\infty} \alpha^t g(x_t^\pi, \mu_t(x_t^\pi)) \mid x_0^\pi = x \right\}, \quad x \in S. \tag{2.7}
\]

The optimal cost function \( J^* : S \rightarrow \mathbb{R} \) is defined by

\[
J^*(x) = \inf_{\pi \in \Pi^\infty} J_\pi(x), \quad x \in S. \tag{2.8}
\]

Accordingly, a policy \( \pi \) is called optimal if \( J_\pi = J^* \).

2.5. The Dynamic Programming Operator and Bellman's Equation

We define the dynamic programming operator \( T : \mathcal{B}(S) \rightarrow \mathcal{B}(S) \), by letting

\[
TJ(x) = \inf_{u \in U(x)} \left\{ g(x, u) + \alpha \int_S J(y)P(y|x, u)dy \right\}, \quad x \in S. \tag{2.9}
\]

It is well known (and is easily shown) that \( T \) is a monotone contraction operator (see e.g. [12], [27]). Furthermore, as a consequence of Assumption A.4, \( T \) has the property

\[
T(J + c1_S) = TJ + \alpha c1_S, \quad \forall J \in \mathcal{B}(S), \quad \forall c \in \mathbb{R}, \tag{2.10}
\]

where \( 1_S \) denotes a function defined on the set \( S \) that is identically equal to 1. A last useful property is

\[
\|TJ - TJ'\|_Q \leq \alpha\|J - J'\|_Q, \quad \forall J, J' \in \mathcal{B}(S), \tag{2.11}
\]

as can be verified by a simple calculation.

Of interest to us is the following lemma:

Lemma 2.1: \( T \) maps \( \mathcal{B}(S) \) into \( \mathcal{C}(S) \). In particular, \( T \) maps \( \mathcal{C}(S) \) into itself.

Proof: Fix some \( J \in \mathcal{B}(S) \). We define

\[
H(x, u) = g(x, u) + \alpha \int_S J(y)P(y|x, u)dy, \quad \forall x \in S, u \in C.
\]

Using Assumptions A.1–A.2 and the fact \( \int_S dy \leq 1 \), we have

\[
\left| H(x, u) - H(x', u') \right| \leq (K + \alpha K \|J\|_\infty)(\|x - x'\|_\infty + \|u - u'\|_\infty), \quad \forall x, x' \in S, u, u' \in C.
\]
Fix some $\epsilon > 0$, $x, x' \in S$, and let $v' \in U(x')$ be such that $H(x', v') \leq \inf_{u' \in U(x')} H(x', u') + \epsilon$. According to Assumption A.3, there exists some $v \in U(x)$ such that $\|v - v\|_{\infty} \leq K\|x - x'\|_{\infty}$. Then,

$$TJ(x) - TJ(x') = \inf_{u \in U(x)} H(x, u) - \inf_{u' \in U(x')} H(x', u')$$

$$\leq H(x, v) - H(x', v') + \epsilon$$

$$\leq K(1 + \alpha\|J\|_{\infty})(\|x - x'\|_{\infty} + \|v - v'\|_{\infty}) + \epsilon$$

$$\leq K(1 + \alpha\|J\|_{\infty})(1 + K)\|x - x'\|_{\infty} + \epsilon.$$  

By symmetry, the same upper bound holds for $TJ(x') - TJ(x)$ as well. Furthermore, since $\epsilon$ is arbitrary, we may let $\epsilon$ decrease to zero to obtain

$$|TJ(x) - TJ(x')| \leq K(1 + K)(1 + \alpha\|J\|_{\infty})\|x - x'\|_{\infty}, \quad \forall x, x' \in S.$$  

This shows that $TJ$ is Lipschitz continuous and, in particular, $TJ \in \mathcal{C}(S)$. q.e.d.

Since $\mathcal{B}(S)$ is a Banach Space and $T$ is a contraction operator on $\mathcal{B}(S)$, the dynamic programming equation (Bellman's Equation) $J = TJ$ has a unique solution in $\mathcal{B}(S)$ (see e.g. [17]); it follows from Lemma 2.1 that such a solution actually belongs to $\mathcal{C}(S)$. Furthermore, by Corollary 9.17.2 of [7], the solution is the same as $J^*$. Hence, $J^*$ is a continuous function and could have been defined as the unique fixed point of the dynamic programming operator $T$.

2.6. Stationary Policies and Associated Operators

For any $\mu \in \Pi$, a policy of the form $\pi = (\mu, \mu, \ldots)$ is called a stationary policy. When dealing with a stationary policy, we abuse notation and use $\mu$ to denote the policy and $J_\mu$ to denote its expected cost function (instead of using $\pi$ and $J_\pi$, respectively). For any $\mu \in \Pi$, we define the operator $T_\mu : \mathcal{B}(S) \mapsto \mathcal{B}(S)$, by letting

$$T_\mu J(x) \overset{\text{def}}{=} g(x, \mu(x)) + \alpha \int_S J(y)P(y|x, \mu(x))dy, \quad x \in S. \quad (2.12)$$

Similarly with $T$, $T_\mu$ is a monotone contraction operator, and satisfies Eqs. (2.10)-(2.11).

It follows from Proposition 7.29 of [7] that $T_\mu$ maps $\mathcal{B}(S)$ into itself. Again since $T_\mu$ is a contraction operator on the Banach space $\mathcal{B}(S)$, $T_\mu$ must have a unique fixed point in $\mathcal{B}(S)$; it is then easily shown that the unique fixed point of $T_\mu$ is $J_\mu$ (see e.g. [7]).

More importantly, it is shown in Corollary 9.17.2 of [7] that there exists an optimal stationary policy. Thus, we can restrict attention to stationary policies and from now on, the word “policy” should be interpreted as “stationary policy”.

2.7. Problem Statement

We are interested in the computation of $J^*$ and of a corresponding optimal policy $\mu^*$. This can be accomplished, in principle, by solving the Bellman Equation $J = TJ$. However, since
Bellman's Equation is infinite-dimensional and nonlinear, we have to be content with computing approximations to $J^*$ and $\mu^*$.

Let $\epsilon > 0$. If a function $J \in B(S)$ satisfies the inequality $\|J - J^*\|_{\infty} \leq \epsilon$, we call it an $\epsilon$-approximation of $J^*$ or an $\epsilon$-optimal cost function. Subsequent sections (Sections 3–7) deal with the computation of an $\epsilon$-optimal cost function for a given MDP. We defer the question of approximating an optimal policy to Section 8.
3. DISCRETIZATION PROCEDURES

The computation of an ε-approximation of $J^*$ is usually accomplished by “discretizing" the original problem and by constructing a new MDP that has finite state and control spaces. However, since we will be comparing functions corresponding to different discretization levels, it is both conceptually and notationally simpler for us to consider MDPs that involve simple functions on $S$ rather than functions on finite subsets of $S$. In this section, we construct such a discretization and estimate the resulting inaccuracy as a function of the grid-spacing and of the discount parameter $\alpha$.

3.1. Discretization of the State and Control Spaces

Let $h \in (0,1]$ be a scalar that parametrizes the coarseness of our discretizations; we call $h$ the grid-size or the grid-level. We start by partitioning the unit interval $I = [0, 1]$ into a collection $I_h$ of subsets. In particular, $I_h$ consists of the set $[0, h]$ together with all nonempty sets of the form $(ih, (i+1)h] \cap I, i = 1, 2, \ldots$. We then partition the unit $n$-dimensional cube $[0,1]^n$ into a collection $I_h^n$ of subsets defined by

$$I_h^n \overset{\text{def}}{=} I_h \times \cdots \times I_h.$$

We discretize the state space by partitioning it into a finite collection of subsets. Each set in this partition is the intersection of $S$ with an element of $I_h^n$. More precisely, we let $S_h$ be the set of all nonempty sets $\sigma$ of the form $\sigma = S \cap \iota, \iota \in I_h^n$, and these sets form the desired partition. We choose a representative element from each $\sigma \in S_h$ and we let $\tilde{S}_h$ be the set of all representatives. For any $x \in S$, we let $\sigma_x$ be the element of $S_h$ to which $x$ belongs. We also use $\tilde{\sigma}_x$ to denote the representative of the set $\sigma_x$.

The control space is discretized by letting $\tilde{C}_h$ be the set of all $(u_1, \ldots, u_m) \in C$ such that each $u_i$ is an integer multiple of $h$. The set of admissible discretized controls is defined by

$$\tilde{U}_h(x) = \{ \tilde{u} \in \tilde{C}_h \mid \| u - \tilde{u}\|_\infty \leq \frac{h}{2} \text{ for some } u \in U(\sigma_x) \}, \quad x \in S. \quad (3.1)$$

For any $x \in S$, the set $U(\sigma_x)$ is nonempty, by assumption. Furthermore, using the definition of $\tilde{C}_h$, for any $u \in U(\sigma_x)$ there exists some $\tilde{u} \in \tilde{C}_h$ such that $\| u - \tilde{u}\|_\infty \leq h/2$. Thus, the set $\tilde{U}_h(x)$ is nonempty for each $x \in S$. It is also easy to see that

$$\tilde{U}_h(x) = \tilde{U}_h(x') = \tilde{U}_h(\sigma_x), \quad \forall x \in S, \forall x' \in \sigma_x. \quad (3.2)$$

3.2. Discretization of the Cost and the Dynamics

We are primarily interested in the case where $h$ is small. We can therefore assume that $h \leq 1/2K$, where $K \geq 1$ is the constant of Assumptions A.1 – A.4. Given some $h \in (0,1/2K]$, we define the
functions $\tilde{g}_h : S \times \tilde{C}_h \mapsto \mathbb{R}$ and $\tilde{P}_h : S \times S \times \tilde{C}_h \mapsto [0, \infty)$ by letting

$$\tilde{g}_h(x, \tilde{u}) \overset{\text{def}}{=} g(x, \tilde{u}), \quad (3.3)$$

$$\tilde{P}_h(y|x, \tilde{u}) \overset{\text{def}}{=} \frac{P(\sigma_y|\sigma_x, \tilde{u})}{\int_S P(\sigma_y|\sigma_x, \tilde{u})dy} \quad (3.4)$$

We verify that $\tilde{P}_h$ is well-defined by checking that the denominator in Eq. (3.4) is nonzero. Indeed, using Assumption A.2,

$$\left| P(\sigma_y|\sigma_x, \tilde{u}) - P(y|\sigma_x, \tilde{u}) \right| \leq K\|y - \sigma_y\|_{\infty} \leq Kh.$$

Thus,

$$\int_S P(\sigma_y|\sigma_x, \tilde{u})dy \geq \int_S P(y|\sigma_x, \tilde{u})dy - \int_S Kh dy \geq 1 - Kh \geq \frac{1}{2},$$

where we have made use of Assumption A.4 and the obvious fact that the volume of $S$ is bounded by 1.

We note that for each $(x, \tilde{u}) \in S \times \tilde{C}_h$, the function $\tilde{P}_h(\cdot|x, \tilde{u})$ is a probability density on $S$. Furthermore, $\tilde{P}_h(y|x, \tilde{u})$ can be viewed as a sample of $P(\cdot|x, \tilde{u})$ at the points $\sigma_y, \sigma_x$, except that the samples are suitably normalized.

We have so far constructed a discretized MDP $(S, \tilde{C}_h, \{\tilde{U}_h(x)\}, \tilde{g}_h, \tilde{P}_h, \alpha)$. The dynamic programming operator $T_h : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ corresponding to this problem is defined by

$$T_hJ(z) \overset{\text{def}}{=} \min_{\tilde{u} \in \tilde{U}_h(z)} \left\{ \tilde{g}_h(z, \tilde{u}) + \alpha \int_S J(y)\tilde{P}_h(y|x, \tilde{u})dy \right\}, \quad J \in \mathcal{B}(S). \quad (3.5)$$

Similarly with $T$, $T_h$ is also a monotone contraction operator and satisfies Eqs. (2.10)-(2.11).

### 3.3. The Discretized Dynamic Programming Equation

Given the partition $\mathcal{S}_h$ of the state space $S$, we say that a function $f$ with domain $S$ is a simple function on $\mathcal{S}_h$ if $f$ is constant on each element of $\mathcal{S}_h$. That is, $f(x) = f(x')$ for every $\sigma \in \mathcal{S}_h$ and every $x, x' \in \sigma$.

For any fixed $\tilde{u} \in \tilde{C}_h$, the functions $\tilde{g}_h(\cdot, \tilde{u})$ and $\int_S J(y)\tilde{P}_h(y|\cdot, \tilde{u})dy$ are simple on $\mathcal{S}_h$. It follows from Eq. (3.5) that for any $J \in \mathcal{B}(S)$, $T_hJ$ is a simple function on $\mathcal{S}_h$.

Since simple functions on $\mathcal{S}_h$ form a complete normed space, the fixed point of $T_h$ must also be a simple function on $\mathcal{S}_h$; in particular, there exists a unique simple function on $\mathcal{S}_h$, denoted by $J^*_h$, that solves the discretized Bellman Equation $J = T_hJ$.

It is not difficult to see that the discretized problem $(S, \tilde{C}_h, \{\tilde{U}_h(x)\}, \tilde{g}_h, \tilde{P}_h, \alpha)$ is equivalent to an MDP whose state space is the finite set $\tilde{S}_h$. To this latter problem, we can associate an optimal cost function $\tilde{J}_h^* : \tilde{S}_h \mapsto \mathbb{R}$ and we have the relation

$$\tilde{J}_h^*(\tilde{s}_x) = J^*_h(x), \quad \forall x \in S.$$
For our purposes, however, it is easier to work with the state space $S$, rather than $\tilde{S}_h$, because $J^*_h$ and $J^*$ are defined on the same set $S$ and can be directly compared.

3.4. Discretization Error Bounds

Our main discretization error estimate is the following:

**Theorem 3.1**: There exist constants $K_1$ and $K_2$ (depending only on the constant $K$ of Assumptions A.1 – A.4) such that for all $h \in (0, 1/2K]$, and all $J \in \mathcal{B}(S)$,

$$\|TJ - T_h J\|_\infty \leq (K_1 + \alpha K_2 \|J\|_Q) h. \quad (3.6)$$

Furthermore,

$$\|J^* - J^*_h\|_\infty \leq \frac{1}{1 - \alpha} (K_1 + \alpha K_2 \|J^*\|_Q) h. \quad (3.7)$$

**Proof**: We start with the following lemma:

**Lemma 3.1**: There exists some constant $K_P$ (depending only on the constant $K$ of Assumptions A.1–A.4) such that

$$|\tilde{P}_h(y|x, \tilde{u}) - P(y|x, \tilde{u})| \leq K_P h, \quad \forall (y, x, \tilde{u}) \in S \times S \times \tilde{C}_h, \quad \forall h \in (0, 1/2K]. \quad (3.8)$$

**Proof**: Fix some $(y, x, \tilde{u}) \in S \times S \times \tilde{C}_h$ and some $h \in (0, 1/2K]$. Using Assumption A.2, we have

$$\left| \int_S P(\tilde{\sigma}_y|\tilde{\sigma}_x, \tilde{u}) dy - 1 \right| = \left| \int_S P(\tilde{\sigma}_y|\tilde{\sigma}_x, \tilde{u}) dy - \int_S P(y|\tilde{\sigma}_x, \tilde{u}) dy \right|$$

$$\leq \int_S \left| P(\tilde{\sigma}_y|\tilde{\sigma}_x, u) - P(y|\tilde{\sigma}_x, u) \right| dy$$

$$\leq K h. \quad (3.9)$$

Hence,

$$|\tilde{P}_h(y|x, \tilde{u}) - P(y|x, \tilde{u})| \leq |\tilde{P}_h(y|x, \tilde{u}) - P(\tilde{\sigma}_y|\tilde{\sigma}_x, \tilde{u})| + |P(\tilde{\sigma}_y|\tilde{\sigma}_x, \tilde{u}) - P(y|x, \tilde{u})|$$

$$\leq P(\tilde{\sigma}_y|\tilde{\sigma}_x, \tilde{u}) \left| 1 - \int_S P(\tilde{\sigma}_y|\tilde{\sigma}_x, \tilde{u}) dy \right| + 2K h$$

$$\leq K \frac{K h}{\int_S P(\tilde{\sigma}_y|\tilde{\sigma}_x, \tilde{u}) dy} + 2K h$$

$$\leq K \frac{K h}{1 - K h} + 2K h$$

$$\leq (2K^2 + 2K) h, \quad (3.10)$$

which proves the desired result, with $K_P = (2K^2 + 2K)$. **q.e.d.**

We continue with the proof of Theorem 3.1. Fix some $J \in \mathcal{B}(S)$ and some $x \in S$. We define

$$H(u) = g(x, u) + \alpha \int_S J(y) P(y|x, u) dy, \quad u \in C, \quad (3.11)$$
\[ \tilde{H}_h(\tilde{u}) = \tilde{g}_h(x, \tilde{u}) + \alpha \int \mathcal{S} J(y) \tilde{P}_h(y|x, \tilde{u}) \, dy, \quad \tilde{u} \in \tilde{C}_h. \] 

(3.12)

Fix some \( \epsilon > 0 \) and let \( v \in U(x) \) be such that \( H(v) \leq \inf_{u \in U(x)} H(u) + \epsilon \). Using Assumption A.3, there exists some \( v' \in U(\tilde{x}) \) such that \( ||v - v'||_{\infty} \leq K||x - \tilde{x}||_{\infty} \leq Kh \). Finally, choose some \( \tilde{v} \in \tilde{U}_h(\tilde{x}) = \tilde{U}_h(x) \) such that \( ||v' - \tilde{v}'||_{\infty} \leq h/2 \). [This is possible because of the way that \( \tilde{U}_h(\tilde{x}) \) is defined.] Notice that

\[ ||v - \tilde{v}||_{\infty} \leq \left( K + \frac{1}{2} \right) h. \]

We now have

\[ T_h J(x) - T J(x) = \inf_{\tilde{u} \in \tilde{U}_h(x)} \tilde{H}_h(\tilde{u}) - \inf_{u \in U(x)} H(u) \]

\[ \leq \tilde{H}_h(\tilde{v}) - H(v) + \epsilon \]

\[ \leq |\tilde{H}_h(\tilde{v}) - H(\tilde{v})| + |H(\tilde{v}) - H(v)| + \epsilon. \]

We bound the two terms in the right-hand side of Eq. (3.13). For the first term,

\[ |\tilde{H}_h(\tilde{v}) - H(\tilde{v})| \leq |\tilde{g}_h(x, \tilde{v}) - g(x, \tilde{v})| + \alpha \int |J(y)| \cdot |\tilde{P}_h(y|x, \tilde{v}) - P(y|x, v)| \, dy \]

\[ \leq |g(\tilde{x}, \tilde{v}) - g(x, \tilde{v})| + \alpha ||J||_{\infty} \int |\tilde{P}_h(y|x, \tilde{v}) - P(y|x, v)| \, dy \]

\[ \leq K||x - \tilde{x}||_{\infty} + \alpha ||J||_{\infty} K_P h \]

\[ \leq (K + \alpha K_P ||J||_{\infty}) h, \]

where \( K_P \) is the constant of Lemma 3.1.

For the second term,

\[ |H(\tilde{v}) - H(v)| \leq |g(x, \tilde{v}) - g(x, v)| + \alpha ||J||_{\infty} \int |P(y|x, \tilde{v}) - P(y|x, v)| \, dy \]

\[ \leq K||\tilde{v} - v||_{\infty} + \alpha ||J||_{\infty} K||\tilde{v} - v||_{\infty} \]

\[ \leq (K + \alpha K ||J||_{\infty}) \left( K + \frac{1}{2} \right) h. \]

We now use the bounds (3.14) and (3.15) in Eq. (3.13) to obtain

\[ T_h J(x) - T J(x) \leq (K_1 + \alpha K'_2 ||J||_{\infty}) h + \epsilon, \]

(3.16)

where \( K_1 \) and \( K'_2 \) are suitable constants.

We now prove a similar inequality for \( T J(x) - T_h J(x) \). Choose some \( \tilde{v} \in \tilde{U}_h(x) \) such that \( H(\tilde{v}) = \inf_{u \in \tilde{U}_h(x)} H(\tilde{u}). \) [Such a \( \tilde{v} \) exists because the set \( \tilde{U}_h(\tilde{x}) \) is finite.] By the definition of \( \tilde{U}_h(x) \), there exists some \( v' \in U(\tilde{x}) \) such that \( ||\tilde{v} - v'||_{\infty} \leq h/2 \). Furthermore, by Assumption A.3, there exists some \( v \in U(x) \) such that \( ||v - v'||_{\infty} \leq Kh \). Then,

\[ T J(x) - T_h J(x) = \inf_{u \in U(x)} H(u) - \inf_{\tilde{u} \in \tilde{U}_h(x)} \tilde{H}_h(\tilde{u}) \]

\[ \leq H(v) - \tilde{H}_h(\tilde{v}). \]
We then follow the steps that led to Eq. (3.16) to obtain

\[ TJ(x) - ThJ(x) \leq (K_1 + \alpha K_2 \| J \|_\infty)h. \] (3.18)

We combine inequalities (3.16) and (3.18) to obtain

\[ |TJ(x) - ThJ(x)| \leq (K_1 + \alpha K_2 \| J \|_\infty)h + \epsilon. \] (3.19)

This inequality holds for every \( \epsilon > 0 \) and it must therefore hold for \( \epsilon = 0 \) as well. Taking the supremum over all \( x \in S \), we obtain

\[ \| TJ - ThJ \|_\infty \leq (K_1 + \alpha K_2 \| J \|_\infty)h. \] (3.20)

To complete the proof of the first part of the theorem, let

\[ c = -\frac{1}{2} \left[ \sup_{x \in S} J(x) + \inf_{x \in S} J(x) \right]. \]

Since \( T \) and \( Th \) satisfy Eq. (2.10), we have

\[ \| TJ - ThJ \|_\infty = \| T(J + c1_S) - Th(J + c1_S) \|_\infty \]
\[ \leq (K_1 + \alpha K_2 \| J + c1_S \|_\infty)h, \] (2.21)

where the last step made use of Eq. (3.20). It is easily seen that \( \| J + c1_S \|_\infty = \| J \|_Q/2 \), and we obtain

\[ \| TJ - ThJ \|_\infty \leq (K_1 + \alpha K_2 \| J \|_Q)h, \] (3.22)

where \( K_2 = K'_2/2 \). Thus, Eq. (3.6) has been established.

**Lemma 3.2:** Let \( T_1, T_2 : B(S) \to B(S) \) be contraction operators with contraction factor \( \alpha \) and let \( J_1, J_2 \in B(S) \) be their respective fixed points. Then,

\[ \| J_1 - J_2 \|_\infty \leq \frac{1}{1 - \alpha} ||T_1J_1 - T_2J_1||_\infty. \]

**Proof:** Using the triangle inequality, we have

\[ \| J_1 - J_2 \| = \| T_1J_1 - T_2J_2 \|_\infty \]
\[ \leq \| T_1J_1 - T_2J_1 \|_\infty + \| T_2J_1 - T_2J_2 \|_\infty \]
\[ \leq \| T_1J_1 - T_2J_1 \|_\infty + \alpha \| J_1 - J_2 \|_\infty, \] (3.23)

from which the result follows. \( \text{q.e.d.} \)

We now use Lemma 3.2 (with \( T_1 = T, T_2 = Th, J_1 = J^*, J_2 = J^*_h \)) and Eq. (3.22), with \( J = J^* \), to obtain

\[ \| J^* - J^*_h \|_\infty \leq \frac{1}{1 - \alpha} ||TJ^* - ThJ^* \|_\infty \leq \frac{1}{1 - \alpha} (K_1 + \alpha K_2 \| J^* \|_Q)h, \]
which completes the proof of the theorem. Q.E.D.

Let $J^0(x) = 0$ for all $x \in S$. It follows from Assumption A.1 that $\|TJ^0\|_Q \leq 2K$. Since $T$ is a contraction operator with respect to the quasi-norm $\cdot \|_Q$ [cf. Eq. (2.11)], we have

$$\|J^*\|_Q = \|TJ^* - J^0\|_Q = \|TJ^* - TJ^0\|_Q + \|TJ^0 - J^0\|_Q \leq \alpha\|J^*\|_Q + \|TJ^0\|_Q,$$

which implies that,

$$\|J^*\|_Q \leq \frac{2K}{1 - \alpha}. \quad (3.24)$$

By an identical argument, we also get

$$\|J^*_h\|_Q \leq \frac{2K}{1 - \alpha}. \quad (3.25)$$

Using Eq. (3.24), the discretization error bound of Theorem 3.1 yields:

Corollary 3.1: There exists some constant $K'$ (depending only on the constant $K$ of Assumptions A.1–A.4) such that for every $h \in (0, 1/2K]$ we have

$$\|J^* - J^*_h\|_Q \leq \frac{K'h}{(1 - \alpha)^2}, \quad (3.26)$$

where

$$K' = K_1 + 2KK_2. \quad (3.27)$$

In the next section, we show that under certain assumptions, $\|J^*\|_Q$ can be bounded by a constant independent of $\alpha$ in which case the bound of Eq. (3.26) can be sharpened.

The following result will be also needed later:

Lemma 3.3: For every $J \in B(S)$ and every $h > 0$, we have

$$\|T_h J\|_Q \leq 2K + \alpha\|J\|_Q.$$

Proof: Using the definition of $T_h$, it is evident that

$$\sup_{x \in S} T_h J(x) \leq \max_{(x,u) \in S \times C} g(x,u) + \alpha \sup_{x \in S} J(x)$$

and

$$\inf_{x \in S} T_h J(x) \geq \min_{(x,u) \in S \times C} g(x,u) + \alpha \inf_{x \in S} J(x).$$

By subtracting these two inequalities, we have

$$\|T_h J\|_Q \leq \max_{(x,u) \in S \times C} g(x,u) - \min_{(x,u) \in S \times C} g(x,u) + \alpha\|J\|_Q \leq 2K + \alpha\|J\|_Q,$$

where the last inequality follows from Assumption A.1. q.e.d.
3.5. Remarks

1. The bounds of Theorem 3.1 are similar to those in Theorem 6.1 of [27], although our discretization procedure is different. A key difference is that our discretized problems are defined on the same state space $S$ (unlike [27]) and all of the operators $T_h$ act on the same function space $\mathcal{B}(S)$. This greatly facilitates the grid-level changes in the multigrid algorithms to be introduced later. For example, in our framework, two iterations on different grids correspond to the application of an operator of the form $T_h T_{h'}$. In contrast, in the framework in [27] a grid-level change requires the application of certain interpolation and projection operators.

2. The normalization of $P_h$ [cf. Eq. (3.4)] ensures that $T_h$ remains a contraction operator with contraction factor $\alpha$ for all $h$. (In particular, the contraction factor is independent of grid-size.) This fact is essential for the validity of our subsequent complexity analysis for the case where $\alpha$ approaches 1.

3. The assumptions that the control space $C$ is equal to $[0, 1]^m$ and that the functions $g$ and $P$ are defined (and are Lipschitz continuous) on the entire set $C$, allow us to use a uniform discretization of $C$, independent of the constraint sets $U(x)$. This idea was used in [4], where the additional requirement $\bar{U}_h(\bar{z}) \subset U(\bar{z})$ was imposed. However, such a requirement is not necessary in our framework.
4. THE ACCESSIBILITY CONDITION

In this section, we introduce and discuss the implications of a *k-stage accessibility condition*. This is a continuous-state formulation of the "scrambling-type" condition discussed in [13,24].

For any $\mu \in \Pi$, let $P_\mu(y|x) = P(y|x, \mu(x))$. Let $\mu_0, \mu_1, \ldots$ be a sequence of elements of $\Pi$. We define a function $P_{\mu_1}\cdots P_{\mu_0} : S \times S \mapsto \mathbb{R}$ by means of the recursive formula

$$
(P_{\mu_i}\cdots P_{\mu_0})(x_{i+1}|x_0) = \int_S P_{\mu_i}(x_{i+1}|x_i)(P_{\mu_{i-1}}\cdots P_{\mu_0})(x_i|x_0)dx_i.
$$

(4.1)

We can interpret $(P_{\mu_i}\cdots P_{\mu_1}P_{\mu_0})(x_{i+1}|x_0)$ as the probability density of the state $x_{i+1}$ at time $i+1$, given that the initial state is $x_0$ and that policy $\pi = (\mu_0, \mu_1, \ldots)$ is used.

We fix some positive integer $k$ and we define

$$
r(y) = \inf_{x \in S, \mu_{k-1}, \ldots, \mu_0 \in \Pi} P_{\mu_{k-1}}\cdots P_{\mu_0}(y|x).
$$

(4.2)

We say that $P$ satisfies a *k-stage accessibility condition with accessibility rate* $\rho > 0$ if

$$
\int_S r(y)dy \geq \rho.
$$

(4.3)

[The integral in Eq. (4.3) is well-defined because $r(\cdot)$ can be shown to be Lipschitz continuous, hence measurable.] Intuitively, Eq. (4.3) states that there exists a subset of $S$ (of non-zero Lebesgue measure) such that the probability density of the random variable $x_k$ is positive on that subset, no matter what the initial state is and what policy is used. Differently said, there is a subset of $S$ that cannot be avoided by judicious choice of a policy and initial state. This condition is easy to check and typically holds when the system being controlled is "sufficiently noisy".

We introduce a similar accessibility condition for the discretized problems constructed in Section 3. For the discretized problems, the set of policies is defined as

$$
\tilde{\Pi}_h = \{ \tilde{\mu}_h : S \mapsto \tilde{C}_h \mid \tilde{\mu}_h \text{ is a simple function on } S_h \text{ and } \tilde{\mu}_h(z) \in \tilde{U}_h(z), \forall z \in S \}.
$$

(4.4)

Notice that $\tilde{\Pi}_h$ is a finite set. For any $\tilde{\mu} \in \tilde{\Pi}_h$, we use the notation $\tilde{P}_{\tilde{\mu}}(y|x) = \tilde{P}_{\tilde{\mu}}(y|x, \tilde{\mu}(x))$. We say that $\tilde{P}_{\tilde{\mu}}$ satisfies a *k-stage accessibility condition with accessibility rate* $\rho > 0$ if

$$
\int_S \tilde{r}(y)dy \geq \rho,
$$

(4.5)

where

$$
\tilde{r}(y) = \min_{x \in S, \tilde{\mu}_{k-1}, \ldots, \tilde{\mu}_0 \in \tilde{\Pi}_h} \tilde{P}_{\tilde{\mu}_{k-1}}\cdots \tilde{P}_{\tilde{\mu}_0}(y|x).
$$

(4.6)

[We use "min" instead of "inf" because $\tilde{\Pi}_h$ is finite and the expression in Eq. (4.6) is a simple function of $x$.]
Our interest in the accessibility condition stems from Theorems 4.1 and 4.2 that are proved next.

**Theorem 4.1:** If $P$ satisfies a $k$-stage accessibility condition with accessibility rate $\rho$, then:

a) $\|T^k J - T^k J'\|_Q \leq \alpha^k (1 - \rho)\|J - J'\|_Q$, for all $J, J' \in \mathcal{B}(S)$.

b) There exists a constant $K_3$ (depending only on $\rho$, $K$, and $k$) such that $\|J^*\|_Q \leq K_3$.

Similarly, if $P_h$ satisfies a $k$-stage accessibility condition with accessibility rate $\rho$, then:

a') $\|T_h^k J - T_h^k J'\|_Q \leq \alpha^k (1 - \rho)\|J - J'\|_Q$, for all $J, J' \in \mathcal{B}(S)$.

b') There exists a constant $K'_3$ (depending only on $\rho$, $K$, and $k$) such that $\|J'_h\|_Q \leq K'_3$.

**Proof:** Using Eq. (2.9), we have

$$TJ = \inf_{\mu \in \Pi} T_{\mu} J, \quad \forall J \in \mathcal{B}(S). \quad (4.7)$$

(Actually, there are some measurability issues that must be addressed in order to establish Eq. (4.7). These are handled in Proposition 7.33 of [7].)

We introduce some notation. For any $x, y \in S$, let $p(y|x) \overset{\text{df}}{=} r(y)$. The function $p$ defines an operator from $\mathcal{B}(S)$ into itself by means of the formula

$$pJ(x) \overset{\text{df}}{=} \int p(y|x)J(y)dy, \quad J \in \mathcal{B}(S).$$

Similarly, each function of the form $P_{\mu_1} \cdots P_{\mu_0}$ [cf. Eq. (4.1)] can be viewed as an operator from $\mathcal{B}(S)$ into itself by letting

$$((P_{\mu_1} \cdots P_{\mu_0})J)(x) = \int_{\mathcal{S}} J(y)(P_{\mu_1} \cdots P_{\mu_0})(y|x)dy.$$

Using Fubini's Theorem, one obtains relations like $P_{\mu_1}((P_{\mu_2} \cdots P_{\mu_0})J) = (P_{\mu_1} P_{\mu_2} \cdots P_{\mu_0})J$ and we can therefore omit parentheses without ambiguity. For any $\mu \in \Pi$, we let $g_{\mu}(x) \overset{\text{df}}{=} g(x, \mu(x))$.

Finally, we use $T^t$ to denote the composition of $t$ replicas of $T$.

Using Eq. (4.7), we obtain

$$T^k J - T^k J' = \inf_{\mu} \{g_{\mu} + \alpha P_{\mu} T^{k-1} J\} - \inf_{\mu} \{g_{\mu} + \alpha P_{\mu} T^{k-1} J'\}$$

$$\leq \sup_{\mu} \{g_{\mu} + \alpha P_{\mu} T^{k-1} J - g_{\mu} - \alpha P_{\mu} T^{k-1} J'\}$$

$$= \alpha \sup_{\mu} \{P_{\mu}(T^{k-1} J - T^{k-1} J')\}.$$

Repeating the above procedure $k - 1$ more times,

$$T^k J - T^k J' \leq \alpha^k \sup_{\mu_{k-1}, \ldots, \mu_0} \{P_{\mu_{k-1}} \cdots P_{\mu_0} (J - J')\}$$

$$= \alpha^k \sup_{\mu_{k-1}, \ldots, \mu_0} \{(P_{\mu_{k-1}} \cdots P_{\mu_0} - p)(J - J') + p(J - J')\}$$

$$\leq \alpha^k \sup_{\mu_{k-1}, \ldots, \mu_0} \{(P_{\mu_{k-1}} \cdots P_{\mu_0} - p)\sup_{x \in S}(J - J')(x)1_S + p(J - J')\}$$

$$= \alpha^k \{(1 - \rho)\sup_{x}(J - J')(x)1_S + r_0 1_S\},$$

17
where \( r_0 = \int_S r(y)(J - J')(y) \, dy \). A symmetrical argument yields,

\[
T^k J - T^k J' \geq \alpha^k \{(1 - \rho) \inf_x (J - J')(x) \mathbf{1}_S + r_0 \mathbf{1}_S \}.
\]

Combining the two bounds and canceling the \( r_0 \mathbf{1}_S \) term,

\[
\|T^k J - T^k J'\|_Q = \sup_x (T^k J - T^k J')(x) - \inf_x (T^k J - T^k J')(x) \\
\leq \alpha^k (1 - \rho) \{(\sup_x (J - J')(x) - \inf_x (J - J')(x)) \} \\
= \alpha^k (1 - \rho) \|J - J'\|_Q,
\]

which establishes part (a).

To prove that \( \|J^*\|_Q \leq K_3 \), let \( J^0(x) = 0 \), for all \( x \in S \). Using the triangle inequality,

\[
\|J^*\|_Q = \|J^* - J^0\|_Q \\
\leq \|T^k J^* - T^k J^0\|_Q + \|T^k J^0 - J^0\|_Q \\
\leq (1 - \rho) \|J^*\|_Q + \|T^k J^0\|_Q,
\]

which implies that

\[
\|J^*\|_Q \leq \frac{1}{\rho} \|T^k J^0\|_Q.
\]

Finally,

\[
\|T^k J^0\|_Q \leq \sum_{i=1}^k \|T^i J^0 - T^{i-1} J^0\|_Q \leq \sum_{i=1}^k \alpha^{i-1} \|T^i J^0\|_Q \leq 2kK.
\]

Hence, \( \|J^*\|_Q \leq K_3 \), where \( K_3 = 2kK/\rho \) is independent of \( \alpha \).

The proof of parts (a') and (b') is identical and is omitted. Q.E.D.

Theorem 4.1 states that, under an accessibility condition, \( T_h \) is a \( k \)-stage contraction with respect to the quasi-norm \( \| \cdot \|_Q \) and the contraction factor is independent of \( \alpha \), even if \( \alpha \) increases to 1. Furthermore, the bound \( \|J^*\|_Q \leq K_3 \), together with Theorem 3.1, leads to a tighter discretization error bound:

Corollary 4.1: There exists some constant \( K'' \) (depending only on the constant \( K \) of Assumptions A.1–A.4, \( k \), and \( \rho \)) such that for every \( h \in (0, 1/2K] \) we have

\[
\|J^* - J^*_h\|_{\infty} \leq \frac{K''h}{1 - \alpha}.
\]

Our next result shows that an accessibility condition on the continuous problem is inherited by the discretized problems, when the discretization is sufficiently fine.

Theorem 4.2: Suppose that \( P \) satisfies a \( k \)-stage accessibility condition with accessibility rate \( 2\rho \). Then there exists some \( h_a > 0 \) (depending only on \( K \), \( k \), and \( \rho \)) such that for all \( h \in (0, h_a] \), \( \tilde{P}_h \)
satisfies a $k$-stage accessibility condition with rate $\rho$.

Proof: We will need the following lemma:

**Lemma 4.1:** For any $\tilde{\mu} \in \tilde{\Pi}_h$, there exists some $\mu \in \Pi$ such that $\|\tilde{\mu} - \mu\|_{\infty} \leq (K + 1)h$.

Proof: Fix some $\tilde{\mu} \in \tilde{\Pi}_h$ and some $x_0 \in S$. The partition that contains $x_0$ is $\sigma_{x_0}$ and its representative is $\tilde{\sigma}_{x_0}$. Let $\tilde{u}_0 = \tilde{\mu}(x_0) = \tilde{\mu}(\tilde{\sigma}_{x_0})$, where the second equality holds because $\tilde{\mu}$ is constant on the set $\sigma_{x_0}$. By the definition of $\tilde{\Pi}_h$, there exists some $u_0 \in U(\tilde{\sigma}_{x_0})$ such that $\|u_0 - \tilde{u}_0\|_{\infty} \leq h/2$.

Let $G \overset{\text{def}}{=} \{u \in C | \|u - \tilde{u}_0\|_{\infty} \leq (K + 1/2)h\}$. By Assumption A.3, $G \cap U(x)$ is nonempty, for all $x \in \sigma_{x_0}$. Thus, for every $x \in \sigma_{x_0}$, we can choose some $\mu(x) \in U(x)$ such that $\|\mu(x) - \tilde{\mu}(x)\|_{\infty} = \|\mu(x) - \tilde{u}_0\|_{\infty} \leq (K + 1/2)h$. By repeating this argument for each set in the partition of $S$ we obtain a function $\mu$ that satisfies the desired inequality. There is one final issue that has to be dealt with: according to the definition of $\Pi$, $\mu$ must be a measurable function. This can be accomplished by appealing to a suitable measurable selection theorem (Proposition 7.33 of [7]). q.e.d.

We now proceed to the proof of the theorem. Let $\tilde{\mu}_0, \tilde{\mu}_1, \ldots, \tilde{\mu}_{k-1}$ be a sequence of elements of $\tilde{\Pi}_h$. Let $\mu_0, \mu_1, \ldots, \mu_{k-1}$ be elements of $\Pi$ such that

$$\|\mu_i - \tilde{\mu}_i\|_{\infty} \leq (K + 1)h, \quad i = 0, 1, \ldots, k - 1.$$  

(They exist by Lemma 4.1.)

Let $\mu$ and $\tilde{\mu}$ be elements of $\Pi$ and $\tilde{\Pi}_h$, respectively. Using Lemma 3.1, there exists a constant $K_P$ such that

$$\|P(y|x, \tilde{\mu}(x)) - P(y|x, \tilde{\mu}(x))\| \leq K_P h, \quad \forall x, y \in S.$$  

(4.8)

Furthermore, by Assumption A.2,

$$|P(y|x, \tilde{\mu}(x)) - P(y|x, \mu(x))| \leq K|\tilde{\mu}(x) - \mu(x)| \leq K\|\mu - \tilde{\mu}\|_{\infty}, \quad \forall x, y \in S.$$  

(4.9)

Combining Eqs. (4.8) and (4.9), we have

$$|\tilde{P}_\mu(y|x) - P_\mu(y|x)| \leq K_P h + K\|\mu - \tilde{\mu}\|_{\infty}, \quad \forall x, y \in S.$$  

In particular,

$$|\tilde{P}_{\mu_i}(y|x) - P_{\mu_i}(y|x)| \leq (K_P + K^2 + K)h, \quad \forall x, y \in S, \quad i = 0, 1, \ldots, k - 1.$$  

Using this inequality and the definition of $P_{\mu_{k-1}} \cdots P_{\mu_1} P_{\mu_0}$ it follows easily that there exists a constant $K_4$ (depending only on $K$ and $k$) such that

$$(\tilde{P}_{\mu_{k-1}} \cdots \tilde{P}_{\mu_1} \tilde{P}_{\mu_0})(y|x) \geq (P_{\mu_{k-1}} \cdots P_{\mu_1} P_{\mu_0})(y|x) - K_4 h, \quad \forall x, y \in S.$$  

Hence,

$$\int_S \tilde{r}(y) \, dy \geq \int_S r(y) \, dy - \int_S K_4 h \, dy \geq 2\rho - K_4 h \geq \rho,$$

provided that $h \leq h_a \overset{\text{def}}{=} \rho/K_4$. Q.E.D.
5. SUCCESSIVE APPROXIMATION ALGORITHMS.

In this section, we introduce the successive approximation algorithm. We review some known bounds on its speed of convergence and study the effects of the accessibility condition. We then introduce a model of computation and analyze the computational requirements of a typical iteration of the algorithm.

5.1 Successive Approximation Error Bounds

The successive approximation algorithm for a discretized problem proceeds as follows. We start with some function \( J \in \mathcal{B}(S) \) which is simple on \( S_h \), and we compute \( T^t_h J \) (\( t = 1, 2, \ldots \)), where \( T^t_h \) stands for the composition of \( t \) replicas of \( T_h \) and \( T^0_h \) represents the identity operator. Since \( T_h \) is a contraction operator (with contraction factor \( \alpha \)) and since \( J_h^\star \) is (by definition) a fixed point of \( T_h \), we have

\[
\| J_h^\star - T^t_h J \|_\infty \leq \alpha^t \| J_h^\star - J \|_\infty. \tag{5.1}
\]

In particular, \( T^t_h J \) converges to \( J_h^\star \). A further consequence of the contraction property of \( T_h \) is the following well-known error bound [12]:

\[
\| J_h^\star - T^t_h J \|_\infty \leq \frac{\alpha}{1 - \alpha} \| T^t_h J - T^{t-1}_h J \|_\infty \leq \frac{\alpha^t}{1 - \alpha} \| T_h J - J \|_\infty. \tag{5.2}
\]

In contrast to Eq. (5.1), the bounds of Eq. (5.2) can be computed with information available to the algorithm.

Since \( T_h \) is also a monotone operator and satisfies Eq. (2.10), the convergence rate of the algorithm can be accelerated by using the following error bounds (see e.g. [5]), that are valid for any \( J \in \mathcal{B}(S) \):

\[
J_h^\star \leq T^{t+1}_h J + \frac{\alpha}{1 - \alpha} \max_{x \in S} \{(T^{t+1}_h J - T^t_h J)(x)\}, \tag{5.3}
\]

\[
J_h^\star \geq T^{t+1}_h J + \frac{\alpha}{1 - \alpha} \min_{x \in S} \{(T^{t+1}_h J - T^t_h J)(x)\}. \tag{5.4}
\]

(We have used "max" and "min" because \( T^t_h J \) and \( T^{t+1}_h J \) are simple functions.) The following is an approximation to \( J_h^\star \) that exploits the bounds of Eqs. (5.3)–(5.4):

\[
J^{t+1} = T^{t+1}_h J + \frac{\alpha}{2(1 - \alpha)} \left[ \min_x \{(T^{t+1}_h J - T^t_h J)(x)\} + \max_x \{(T^{t+1}_h J - T^t_h J)(x)\} \right]. \tag{5.5}
\]

We subtract Eq. (5.3) or (5.4) from Eq. (5.5) to obtain

\[
\| J_h^\star - J^{t+1} \|_\infty \leq \frac{\alpha}{2(1 - \alpha)} \| T^{t+1}_h J - T^t_h J \|_Q \leq \frac{\alpha^{t+1}}{2(1 - \alpha)} \| T_h J - J \|_Q. \tag{5.6}
\]

This bound is not much better than the bound of Eq. (5.2). However, if we assume that a \( k \)-stage accessibility condition holds, Theorem 4.1 yields

\[
\| T^{tk+1}_h J - T^{tk}_h J \|_Q \leq \alpha^{tk}(1 - \rho)^t \| T_h J - J \|_Q \leq (1 - \rho)^t \| T_h J - J \|_Q. \tag{5.7}
\]

20
Combining with Eq. (5.6), we obtain

$$\|J_h^k - J_h^{k+1}\|_\infty \leq \frac{(1 - \rho)^k}{2(1 - \alpha)} \|T_h J - J\|_Q,$$

(5.8)

Thus, the distance of $J^k$ from $J_h^*$ contracts by a factor of at least $(1 - \rho)$ every $k$ iterations. In particular, the convergence rate is independent of $\alpha$.

Before we can analyze the computational requirements (the complexity) of a typical iteration of the algorithm, we must first define our model of computation, which is done in the next subsection.

5.2. Model of Computation

Given that we are dealing with problems involving continuous variables, discrete model of computation (such as Turing machines [18]) are not suitable. We shall use instead a continuous model in which arithmetic operations are performed on infinite precision real numbers (see [19] and [25] for related models).

Our model consists of three components:

(a) A mechanism for reading the input.

The input to the computation is provided by means of an "oracle" that works as follows:

(i) To obtain information about $S$, a computer submits to the oracle "queries" consisting of an element $t \in J_h^n$. If $t \cap S$ is empty then the oracle returns a special symbol to indicate this fact; otherwise, the oracle returns an element in $t \cap S$ and the volume $\lambda_n(t \cap S)$ of that set, where $\lambda_n(\cdot)$ stands for the Lebesgue measure.

(ii) To obtain information about $U(x)$, a computer submits to the oracle a pair $(h, x)$ and the oracle returns a list of the elements of the set $U_h(x)$.

(iii) Finally, to obtain values of $g$ and $P$ at some specific points, the computer submits to the oracle a triple $(x, y, u)$, and the oracle returns the values of $P(y|x, u)$ and $g(x, u)$.

(b) The nature of the allowed computations.

We consider a computing machine, or simply a "computer" that has the capability of performing comparisons and elementary arithmetic operations on infinite precision real numbers. Furthermore, the computer can use the results of earlier computations to decide what queries to submit to the oracle. The rules by which the computing machine decides at each step what to do next will be referred to as an "algorithm".

(c) A format for representing the output of the computation.

In our case, the output of the computation is a function $J$ which is simple on $S_h$, where the discretization parameter $h$ is to be decided by the computer itself. One possible format is the following. The computer first outputs the value of $h$, which implicitly specifies the partition $S_h$ of $S$. It then outputs the pair $(\bar{x}, J_h(\bar{x}))$, for every $\bar{x} \in \bar{S}_h$. 

21
There are some additional assumptions that have to be made in our particular context: The computer is provided the values of the dimensions \( m \) and \( n \) (of \( C \) and \( S \), respectively), the discount factor \( \alpha \), the desired accuracy \( \epsilon \), and the constant \( K \) of Assumptions A.1–A.4. Furthermore, if a \( k \)-stage accessibility condition is assumed, the computer is also given the values of \( k \) and of the accessibility rate \( \rho \).

The computational cost of an algorithm (also called its complexity) will be counted in a very simple manner: each query to the oracle costs one unit; similarly, each arithmetic operation or comparison costs one unit. [In a variation of this model, a query asking for the elements of a set \( \tilde{U}_h(x) \) could have cost equal to the cardinality of the set returned by the oracle. Our complexity estimates, however, are not sensitive to minor variations of this type.]

Let us fix the dimensions \( m, n \), of \( S \) and \( C \), respectively, the constant \( K \) of Assumptions A.1–A.4, and the constants \( k \) and \( \rho \) involved in the accessibility condition of Section 4. Once these parameters are fixed, let \( P(\alpha) \) be the set of all MDPs with discount factor \( \alpha \) and let \( P = \cup_{\alpha \in (0,1)} P(\alpha) \). Let us consider an algorithm \( \gamma \) that given any \( \epsilon > 0 \) and any MDP in \( P \), returns an \( \epsilon \)-optimal cost function. We use \( C(\gamma;\alpha,\epsilon) \) to denote the worst case running time of this algorithm for a particular value of \( \epsilon \) and where the worst case is taken over all MDPs belonging to \( P(\alpha) \). We then define the complexity \( C(\alpha,\epsilon) \) of solving MDPs as the minimum of \( C(\gamma;\alpha,\epsilon) \) over all algorithms \( \gamma \) with the above mentioned properties.

There is a similar definition of the complexity of solving MDPs that satisfy a \( k \)-stage accessibility condition with accessibility rate \( \rho \). The details of this definition are analogous to the one in the preceding paragraph. We use \( C_{\text{acc}}(\alpha,\epsilon) \) to denote this complexity.

It is convenient to only consider order of magnitude estimates when arguing about algorithm or problem complexity. We thus introduce the following notation:

(a) Let \( f, g : (0,1] \rightarrow [0,\infty) \) be functions of the grid-size \( h \). We write \( f = O(g) \) if there exist constants \( c \) and \( h_0 > 0 \) such that \( f(h) \leq cg(h) \) for all \( h \in (0,h_0] \). We also write \( f = \Omega(g) \) if \( g = O(f) \).

(b) Let \( f, g : (0,1) \times (0,1] \rightarrow [0,\infty) \) be functions of \( \alpha \) and \( \epsilon \). We write \( f = O(g) \), if there exist constants \( c, \epsilon_0 > 0 \), and \( \alpha_0 < 1 \) such that \( f(\epsilon,\alpha) \leq cg(\epsilon,\alpha) \), for all \( \epsilon \in (0,\epsilon_0] \) and \( \alpha \in [\alpha_0,1) \). We also write \( f = \Omega(g) \) if \( g = O(f) \).

5.3. The Complexity of Evaluating \( T_hJ \)

We estimate here the complexity of evaluating \( T_hJ \) according to the formula

\[
T_hJ(x) = \min_{\tilde{u} \in \tilde{U}_h(x)} \left\{ \tilde{g}_h(x,\tilde{u}) + \alpha \int_S J(y)\tilde{P}_h(y|x,\tilde{u})dy \right\}, \tag{5.9}
\]

for the case where \( J \) is a simple function on \( S_h \). Since \( T_hJ \) also turns out to be a simple function
on \( S_h \), we only need to determine the values of \( T_h J \) for \( \bar{x} \in \tilde{S}_h \). Thus, \( T_h J \) is determined by

\[
T_h J(\bar{x}) = \min_{\bar{u} \in \tilde{U}(\bar{x})} \left\{ \tilde{g}_h(\bar{x}, \bar{u}) + \alpha \sum_{\tilde{y} \in \tilde{S}_h} J(\tilde{y}) \tilde{P}_h(\tilde{y}|\bar{x}, \bar{u}) \lambda_n(\sigma_{\tilde{y}}) \right\}, \quad \bar{x} \in \tilde{S}_h, \tag{5.10}
\]

where \( \lambda_n \) stands for the \( n \)-dimensional Lebesgue measure.\(^1\)

We make the following observations. Since \( |\tilde{S}_h| = O(h^{-n}) \), and \( |\tilde{U}_h(\bar{x})| \leq |\tilde{C}_h| = O(h^{-m}) \), there are \( O(h^{-(n+m)}) \) different pairs \((\bar{x}, \bar{u})\). Also, for any fixed \( \bar{x} \) and \( \bar{u} \), the right-hand side of Eq. (5.10) can be computed with \( O(h^{-n}) \) operations, with most of the work needed for the summation. Thus, the total time spent in arithmetic operations and comparisons is \( O(h^{-(2n+m)}) \).

Furthermore, \( O(h^{-(2n+m)}) \) oracle queries are sufficient for obtaining the required values of the functions \( \tilde{g}_h, \tilde{P}_h \), and of the elements of the sets \( \tilde{U}_h(\bar{x}) \). We have therefore proved the following:

**Lemma 5.1:** If \( J \) is a simple function on \( S_h \), then the complexity of computing \( T_h J \) is \( O(1/h^{2n+m}) \).

In our estimates, we have assumed that the minimization with respect to \( \bar{u} \) is carried out by exhaustive enumeration. In practice, the dependence on \( u \) may have a special structure that can be exploited to reduce the computational requirements. Nevertheless, our analysis will be carried out for the general case where no special structure is assumed.

---

\(^1\) This formula should explain why we have assumed that the oracle can provide information on the volume of certain sets [see item (a)(i) in Subsection 5.2]. If such volume information were not directly available, then it should be somehow estimated. Although this could be an important issue in practice, its theoretical aspects are somewhat tangential to the present work.
6. SINGLE-GRID SUCCESSIVE APPROXIMATION AND ITS COMPLEXITY

In this section, we describe the single-grid successive approximation algorithm and analyze its complexity using the model of computation of Subsection 5.2. We consider separately: (i) the general case, where the problem is not assumed to satisfy an accessibility condition, and (ii) the special case, where the problem is assumed to satisfy a k-stage accessibility condition with accessibility rate $\rho$.

The basic idea in single-grid successive approximation is that we choose a grid-size $h_f$ so that $\| J^* - J_{h_f}^* \|_{\infty}$ is small. We then keep applying the operator $T_h$, until a sufficiently accurate approximation of $J_{h_f}^*$ is obtained.

6.1. The General Case.

Let $\epsilon$ be the desired accuracy. From the discretization error bound of Corollary 3.1, we have

$$\| J^* - J_{h_f}^* \|_{\infty} \leq \frac{K'}{(1-\alpha)^2} h_f. \quad (6.1)$$

Thus, if we let

$$h_f = \frac{(1-\alpha)^2 \epsilon}{2K'}, \quad (6.2)$$

we obtain $\| J^* - J_{h_f}^* \|_{\infty} \leq \epsilon/2$. [Actually, Corollary 3.1 has the condition $h \leq 1/2K$. This of no concern because we are interested in the cases where $\epsilon \downarrow 0$ and/or $\alpha \uparrow 1$. In these cases, Eq. (6.2) shows that $h_f$ becomes arbitrarily small.] With our choice of $h_f$, the complexity of evaluating $T_{h_f} J$ for some $J$ that is simple on $S_h$, is $O\left(1/((1-\alpha)^2 \epsilon)^{2n+m}\right)$ (Lemma 5.1).

Let $J^0(x) = 0$ for all $x \in S$, and apply $T_{h_f}$ on $J^0$ for $t$ times, where $t$ is the smallest integer satisfying

$$\frac{\alpha^t}{2(1-\alpha)} \| T_{h_f} J^0 \|_{Q} \leq \frac{\epsilon}{2}.$$

Let $J^t$ be as defined in Eq. (5.5). Then, Eq. (5.6) yields $\| J_{h_f}^* - J^t \|_{\infty} \leq \epsilon/2$, and the triangle inequality shows that

$$\| J^* - J^t \|_{\infty} \leq \| J^* - J_{h_f}^* \|_{\infty} + \| J_{h_f}^* - J^t \|_{\infty} \leq \epsilon,$$

as desired.

We now bound the complexity of this algorithm. Since $\| T_{h_f} J^0 \|_{Q} \leq 2K$, it is seen that

$$t \leq \frac{\log \left[2K/((1-\alpha)\epsilon)\right]}{\log \alpha} + 1 = O\left(\frac{\log \frac{1}{(1-\alpha)\epsilon}}{\log \alpha}\right).$$

Therefore, the complexity of the algorithm is

$$O\left(\frac{\log \frac{1}{(1-\alpha)\epsilon}}{\log \alpha} \left[ \frac{1}{(1-\alpha)^2 \epsilon} \right]^{2n+m}\right).$$
6.2. The Special Case

We now impose an accessibility condition, with accessibility rate $\rho$. Corollary 4.1 yields

$$
\| J^* - J^*_{h_f} \|_\infty \leq \frac{K''}{(1 - \alpha)} h_f.
$$

We wish to have $\| J^* - J^*_{h_f} \|_\infty \leq \epsilon/2$ and this can be accomplished by letting

$$
h_f = \frac{(1 - \alpha)\epsilon}{2K''}.
$$

Accordingly, the complexity of each iteration is $O \left( 1 / ((1 - \alpha)\epsilon)^{2n+m} \right)$.

Let again $J^0(x) = 0$ for all $x \in S$, and apply $T_{h_f}$ on $J^0$ for $l k + 1$ times. Equation (5.8) yields

$$
\| J^*_{h_f} - J^{ik+1} \|_\infty \leq \frac{(1 - \rho)^i}{2(1 - \alpha)} \| T_{h_f} J^0 \|_Q \leq \frac{(1 - \rho)^i}{2(1 - \alpha)} 2K
$$

(6.3)

We now bound the complexity of the algorithm. We desire to have $\| J^*_{h_f} - J^{ik+1} \|_\infty \leq \epsilon/2$ and, from Eq. (6.3), this can be achieved with

$$
l \leq \frac{\log \frac{2K}{(1 - \alpha)^{\epsilon}}}{|\log(1 - \rho)|} + 1 = O \left( \log \frac{1}{(1 - \alpha)\epsilon} \right).
$$

So, the complexity of the algorithm is

$$
O \left( \log \frac{1}{(1 - \alpha)\epsilon} \left[ \frac{1}{(1 - \alpha)\epsilon} \right]^{2n+m} \right).
$$

We summarize our results in the following theorem:

**Theorem 6.1:** There holds

$$
C(\alpha, \epsilon) = O \left( \frac{1}{|\log \alpha|} \left[ \frac{1}{(1 - \alpha)^2\epsilon} \right]^{2n+m} \right),
$$

$$
C_{\text{acc}}(\alpha, \epsilon) = O \left( \log \frac{1}{(1 - \alpha)\epsilon} \left[ \frac{1}{(1 - \alpha)\epsilon} \right]^{2n+m} \right).
$$

Furthermore, the complexity of the single-grid successive approximation algorithm is within these bounds.
7. COMPLEXITY OF MULTIGRID SUCCESSIVE APPROXIMATION

In this section, we introduce a multigrid version of the algorithm of Section 6 and estimate its complexity. The first iterations of this algorithm are executed with a relatively large value of \( h \) (coarse grid) and the value of \( h \) is gradually reduced (grid refinement) as the algorithm proceeds. The basic idea is that the results of the initial iterations are fairly inaccurate approximations of \( J^* \), so the use of a very fine grid is unnecessary. Thus, most iterations are executed on relatively coarse grids, with much less computational costs, and the overall complexity of the algorithm is improved.

Multigrid methods have been extensively studied in the context of partial differential equations, and have been found to lead to substantially faster convergence (both theoretically, and in practice) [9], [14]. As far as dynamic programming problems are concerned, our method seems to be new. Some alternative methods ([1,2], [16]) are discussed in Subsection 7.3.

As in Section 6, we will analyze the complexity of the multigrid algorithm for the general case and for the special case where an accessibility condition is imposed. Our results show that the complexity of multigrid successive approximation is (for both cases) better than that of the single-grid method by a factor of \( \log \left( \frac{1}{((1 - \alpha)\epsilon)} \right) \), and is optimal in a sense to be discussed in Subsection 7.5.

7.1 The General Case

The algorithm starts by fixing an appropriate coarsest grid-level (discretization parameter) \( h_0 \). The choice of \( h_0 \) is independent of \( \alpha \) and \( \epsilon \), but we require that \( h_0 \leq \frac{1}{2K} \), so that the discretization error bound of Corollary 3.1 applies. We then compute the function \( J^*_{h_0} \) exactly, and let \( J^F_{h_0} = J^*_{h_0} \). We switch to a new grid-level by replacing \( h_0 \) by \( h_0/2 \), and use \( J^F_{h_0} \) to initialize the computations at the new grid-level.

More generally, at any grid-level \( h \), we do the following. We start with an initial estimate \( J^I_h \) and we compute \( T^t_h J^I_h \), \( t = 1, 2, \ldots, t(h), \) where \( t(h) \) is the smallest positive integer such that

\[
\| T^{t_t(h)}_h J^I_h - T^{t(h)-1}_h J^I_h \|_Q \leq \frac{2K'h}{\alpha(1 - \alpha)}.
\]  

[The fact that such a \( t(h) \) exists is evident because \( T^t_h J^I_h \) converges.] At that point, we let

\[
J^F_h = T^{t(h)}_h J^I_h + \frac{2(1 - \alpha)}{\alpha} \left[ \min_x \left( (T^{t(h)}_h J^I_h - T^{t(h)-1}_h J^I_h)(x) \right) + \max_x \left( (T^{t(h)}_h J^I_h - T^{t(h)-1}_h J^I_h)(x) \right) \right],
\]  

which is our final estimate at the current grid-level. Then, Eq. (5.6) yields

\[
\| J^*_h - J^F_h \|_\infty \leq \frac{\alpha}{2(1 - \alpha)} \| T^{t(h)}_h J^I_h - T^{t(h)-1}_h J^I_h \|_Q \leq \frac{\alpha t(h)}{2(1 - \alpha)} \| T_h J^I_h - J^I_h \|_Q.
\]  

If

\[
\frac{K'h}{(1 - \alpha)^2} \leq \epsilon
\]  

(7.4)
the algorithm terminates. Otherwise, we replace $h$ by $h/2$ and use the final function $J_h^F$ of the current grid-level to initialize the computations at the next grid-level. That is, $J_{h/2}^I = J_h^F$.

It is clear that after a finite number of grid-level changes, Eq. (7.4) will be satisfied, and this shows that the algorithm eventually terminates.

We now verify correctness of the algorithm. Let $h_f$ be the final grid-level at which the algorithm terminates. Using Corollary 3.1, we have

$$\|J_{h_f}^* - J^*\|_\infty \leq \frac{K'h_f}{(1-\alpha)^2} \leq \frac{\epsilon}{2}. \tag{7.5}$$

Furthermore, Eqs. (7.1) and (7.3) yield

$$\|J_{h_f}^* - J_{h_f}^F\|_\infty \leq \frac{\alpha}{2(1-\alpha)} \|T_{h_f}^{t(h_f)} J_{h_f}^I - T_{h_f}^{t(h_f)} J_{h_f}^I\|_Q \leq \frac{K'h_f}{(1-\alpha)^2} \leq \frac{\epsilon}{2}. \tag{7.6}$$

Equations (7.5)-(7.6) and the triangle inequality yield $\|J^* - J_{h_f}^F\|_\infty \leq \epsilon$, as desired.

In order to develop a complexity estimate, we need to bound the number $t(h)$ of iterations at each grid-level. This is done in the following two lemmas.

**Lemma 7.1:** For $h \in \{h_0/2, h_0/4, \ldots, h_f\}$, and every $t \in \{0, 1, \ldots, t(h)\}$, we have $\|T_{h}^IJ_{h}^I\|_Q \leq 2K/(1-\alpha)$.

**Proof:** The proof proceeds by induction. We have $\|J_{h_0/2}^I\|_Q = \|J_{h_0}^*\|_Q \leq 2K/(1-\alpha)$, by Eq. (3.25). Assume that $\|J_{h}^I\|_Q \leq 2K/(1-\alpha)$ for some $h \in \{h_0/2, h_0/4, \ldots, h_f\}$. Then, using Lemma 3.3, $\|T_{h}J_{h}^I\|_Q \leq 2K + \alpha 2K/(1-\alpha) = 2K/(1-\alpha)$ and, continuing inductively, the same bound holds for $\|T_{h}^IJ_{h}^I\|_Q$, $t = 1, \ldots, t(h)$. It is seen from Eq. (7.2) that

$$\|J_{h/2}^I\|_Q = \|J_{h}^I\|_Q = \|T_{h/2}^IJ_{h/2}^I\|_Q \leq \frac{2K}{1-\alpha}. \tag{7.8}$$

q.e.d.

**Lemma 7.2:** There exists a constant $c$, independent of $\alpha$ and $\epsilon$, such that $t(h) \leq c/\|\log \alpha\|_\epsilon$, for $h = h_0/2, h_0/4, \ldots, h_f$.

**Proof:** Fix some $h \in \{h_0/4, h_0/8, \ldots, h_f\}$ and let $\hat{J} = T_{2h}^{t(2h)}J_{2h}^I$. (Thus, $\hat{J}$ is the function available just before the last iteration at grid-level $2h$.) Then, Eq. (7.1) yields

$$\|T_{2h}\hat{J} - J_{2h}^\hat{J}\|_Q \leq \frac{4K'h}{\alpha(1-\alpha)} \tag{7.7}$$

Using the triangle inequality, the fact that $\|\cdot\|_Q \leq 2\|\cdot\|_\infty$, Eq. (7.7), and Theorem 3.1, we have

$$\|T_{h}T_{2h}\hat{J} - T_{2h}\hat{J}\|_Q \leq \|T_{h}T_{2h}\hat{J} - T_{2h}\hat{J}\|_Q + \|TT_{2h}\hat{J} - T_{2h}T_{2h}\hat{J}\|_Q + \|T_{2h}T_{2h}\hat{J} - T_{2h}\hat{J}\|_Q$$

$$\leq 2\left(\|T_{h}T_{2h}\hat{J} - T_{2h}\hat{J}\|_\infty + \|TT_{2h}\hat{J} - T_{2h}T_{2h}\hat{J}\|_\infty\right) + \alpha\|T_{2h}\hat{J} - J_{2h}\hat{J}\|_Q$$

$$\leq 2\left((K_1 + \alpha K_2\|T_{2h}\hat{J}\|_Q)h + (K_1 + \alpha K_2\|T_{2h}\hat{J}\|_Q)2h\right) + \alpha\frac{4K'h}{\alpha(1-\alpha)}. \tag{7.8}$$
By Lemma 7.1, we have $\|T_{2h}J\|_{\infty} \leq \frac{2K}{1-\alpha}$. Using this inequality in Eq. (7.8), we obtain

$$\|T_{h}T_{2h}J - T_{2h}J\|_{Q} \leq 6\left(K_{1} + \alpha K_{2} \frac{2K}{1-\alpha}\right) h + \frac{4K'h}{(1-\alpha)} \leq \frac{10K'h}{1-\alpha}$$

(7.9)

where the last inequality follows from the fact $K' = K_{1} + 2K_{2}K$ [cf. Eq. (3.27)]. Notice that the left-hand side of Eq. (7.9) is equal to $\|T_{h}J_{h} - J_{h}'\|_{Q}$. Using Eq. (7.9) and the fact that $T_{h}$ is a contraction operator, with contraction factor $\alpha$, with respect to the quasi-norm $\| \cdot \|_{Q}$, we obtain

$$\|T_{h}^{t}J_{h}^{t} - T_{h}^{t-1}J_{h}^{t-1}\|_{Q} \leq \alpha^{t-1}\|T_{h}J_{h}^{t} - J_{h}'\|_{Q} \leq \alpha^{t-1}\frac{10K'h}{1-\alpha}.$$  

(7.10)

In particular, if $t$ is chosen so that $10\alpha^{t} \leq 2$, then the termination condition of Eq. (7.1) is satisfied. This shows that $t(h)$ is no larger than the smallest $t$ such that $10\alpha^{t} \leq 2$ and, therefore, $t(h) \leq c/|\log \alpha|$, where $c$ is some absolute constant.

The proof for the case $h = h_{0}/2$ is identical, provided that we define $J' = J_{ho/2} = J_{ho}^{*}$. We then have $\|T_{2h}J' - J'\|_{Q} = \|T_{h}J_{ho}^{*} - J_{ho}^{*}\|_{Q} = 0$ and inequality (7.7) is trivially true. The rest of the argument holds without any changes. q.e.d.

Notice that at each grid-level $h$ we start with a function $J_{h}^{t}$ that is simple on $S_{2h}$ and, therefore, simple on $S_{h}$. Since only simple functions are involved, Lemma 5.1 provides an estimate of the complexity of each iteration. Using also Lemma 7.1 to estimate the number of iterations at each grid-level, the total complexity of the algorithm is

$$C(\alpha, \epsilon) = O\left(\frac{1}{|\log \alpha|} \left[(1/hf)^{2n+m} + (1/2hf)^{2n+m} + (1/4hf)^{2n+m} + \ldots\right]\right)$$

$$= O\left(\frac{1}{|\log \alpha|} \left(\frac{1}{hf}\right)^{2n+m} \left[1 + \frac{1}{2} + \frac{1}{4} + \ldots\right]\right)$$

$$= O\left(\frac{1}{|\log \alpha|} \left(\frac{1}{(1-\alpha)^{2}\epsilon}\right)^{2n+m}\right).$$

(7.11)

[The last step in Eq. (7.11) uses the relation $hf = \Omega(\epsilon(1 - \alpha)^{2})$ which is a consequence of the termination criterion (7.4).] Notice that we have ignored the computations involved at the first grid-level $h_{0}$. This is justifiable because we can compute $J_{ho}^{*}$ with a number of operations that is independent of $\alpha$ and $\epsilon$ (e.g., using linear programming or policy iteration) and let $J_{ho}^{*} = J_{ho}^{*}$. In practice, we might only compute an approximation of $J_{ho}^{*}$, e.g., by using the successive approximation algorithm at grid-level $h_{0}$. It is easily verified that such a modification does not change our complexity estimate.

7.2. The Special Case

We now assume that the problem satisfies a $k$-stage accessibility condition with accessibility rate $\rho$. The algorithm is almost the same except for the following differences. The initial grid-size $h_{0}$ is
chosen to satisfy \( h_0 \leq \min(1/2K, h_a) \), where \( K \) is the constant of Assumptions A.1–A.4 and \( h_a \) is the constant of Theorem 4.2. Furthermore, the termination criterion of Eq. (7.4) is replaced by

\[
\frac{K''h}{1 - \alpha} \leq \frac{\epsilon}{2},
\]

(7.12)

where \( K'' \) is the constant of Corollary 4.1.

The proof of termination is the same as in Subsection 7.1. Correctness of the algorithm also follows similarly, except that we have to invoke Corollary 4.1 instead of Corollary 3.1. We now bound the number of iterations at each grid-level.

**Lemma 7.3:** Under the accessibility condition, there exists a constant \( c \), independent of \( \alpha \) and \( \epsilon \), such that \( t(h) \leq c \), for \( h = h_0/2, h_0/4, \ldots, h_f \).

**Proof:** The proof is identical with the proof of Lemma 7.2. The only difference is that, under the \( k \)-stage accessibility condition, Eq. (7.10) gets replaced by

\[
\|T_k^{L+1}J_k^I - T_k^{L}J_k^I\|_Q \leq (1 - \rho)^{L}\|T_k^{L}J_k^I - J_k^I\|_Q.
\]

As \( \alpha \) is replaced by the absolute constant \( 1 - \rho \), it follows that \( t(h) \) is also bounded by an absolute constant independent of \( \alpha \). q.e.d.

We now use Lemma 7.3 to estimate the complexity of the algorithm. We obtain

\[
C_{\text{acc}}(\alpha, \epsilon) = O \left( \left(\frac{1}{h_f}\right)^{2n+m} + \left(\frac{1}{2h_f}\right)^{2n+m} + \left(\frac{1}{4h_f}\right)^{2n+m} + \cdots \right)
\]

\[
= O \left( \left[ \frac{1}{h_f} \left(1 + \frac{1}{2} + \frac{1}{4} + \cdots \right) \right]^{2n+m} \right)
\]

\[
= O \left( \left[ \frac{1}{(1 - \alpha)\epsilon} \right]^{2n+m} \right).
\]

We have used in the last step the fact \( h_f = \Omega(\epsilon(1 - \alpha)) \) which is a consequence of Eq. (7.12). We summarize our conclusions:

**Theorem 7.1:** There holds

\[
C(\alpha, \epsilon) = O \left( \frac{1}{|\log \alpha|} \left( \frac{1}{(1 - \alpha)^{2\epsilon}} \right)^{2n+m} \right)
\]

(7.13)

and

\[
C_{\text{acc}}(\alpha, \epsilon) = O \left( \left( \frac{1}{(1 - \alpha)\epsilon} \right)^{2n+m} \right).
\]

(7.14)

Furthermore, the complexity of the multigrid successive approximation algorithms presented in this Section is within these bounds.

A comparison of Theorems 6.1 and 7.1 shows that the multigrid algorithm is an improvement over its single-grid counterpart.
7.3 A Comparison With Other Algorithms

We compare our multigrid algorithm with the algorithms reported in [1,2], [16]. The main differences are as follows:

1. The problems solved in these references involve continuous time and lead to an elliptic partial differential equation, while we are dealing with discrete-time problems that lead to an integral equation.

2. The algorithms of [1,2], [16] are based on policy iteration whereas we use successive approximation. The policy iteration algorithm involves a “policy evaluation” step which amounts to solving the linear equation \( T_\mu J_\mu = J_\mu \), where \( \mu \) is a certain policy. It is then suggested that the solution of this equation be carried out using a multigrid algorithm. Whereas an algorithm similar to ours could be suitable for that task, the multigrid algorithm of [1,2], [16] is radically different. Ours proceeds from coarser to finer grids, whereas the algorithm in these references moves repeatedly up and down between different grids. This latter strategy is certainly appropriate for the solution of certain partial differential equations [14], but it is unclear if it could be beneficial for the solution of discounted discrete-time problems.

3. The complexity analysis in [1,2] is carried out only for a specific example. Furthermore, the analysis is based on a heuristic correspondence between policy iteration and Newton’s method, together with an implicit assumption that Newton’s method converges very fast.

There may be practical reasons for choosing policy iteration over successive approximation, even though it seems rather difficult to establish nice bounds on the number of policy iteration steps required for convergence. If policy iteration is employed, our multigrid algorithm may be still used for policy evaluation. The complexity estimates in this case are the same as the ones in Theorem 7.1, except that \( m \) should be set to zero, to reflect the fact that fixing a policy eliminates the burden of choosing an optimal value of the control at each state.

7.4. Lower Bounds and the Optimality of Multigrid Successive Approximation

The following lower bounds on the complexity of the solution of MDPs have been established in [11]:

\[
C(\alpha, \epsilon) = \Omega \left( \left( \frac{1}{(1 - \alpha)^2} \right)^{2n+m} \right),
\]

\[
C_{\text{acc}}(\alpha, \epsilon) = \Omega \left( \left( \frac{1}{(1 - \alpha)} \right)^{2n+m} \right),
\]

and apply to any conceivable algorithm, within our model of computation.

By comparing these lower bounds with the complexity of our algorithm (Theorem 7.1), we note the following:

1. For problems satisfying the accessibility condition, our algorithm is optimal; that is, its complexity is within a constant factor of the lower bound.
2. Without an accessibility condition, our algorithm is always within a factor of $O(1/|\log \alpha|) = O(1/(1 - \alpha))$ of the optimal. In particular, if $\alpha$ is fixed and we concentrate on the dependence on $\epsilon$, our algorithm is again optimal.

Let us also mention that the above lower bounds hold for any problem discretization. Furthermore, these lower bounds also bound the number of oracle queries needed in order to obtain sufficient information to compute an $\epsilon$-approximation of $J^*$ [11]. We then notice that the number of oracle queries in the multigrid algorithm is equal to the lower bounds. This implies that our discretization scheme is optimal, in the sense that no discretization using a smaller number of queries could accomplish the desired goal.
8. COMPUTING $\varepsilon$-OPTIMAL POLICIES

In this section, we consider the computation of an $\varepsilon$-optimal policy, that is, a stationary policy whose expected cost is within $\varepsilon$ of the optimal. The main result of this section is that the upper and lower bounds of Section 7 are applicable to this problem as well; furthermore, computing an $\varepsilon$-optimal policy is "as hard as" computing an $\varepsilon$-optimal cost function (that is, the cost of computing the former is within a constant factor of the cost of computing the latter, and vice versa).

8.1 Definition of $\varepsilon$-Optimal Policies

Given a value of the discretization parameter $h$, we consider the set $\bar{\Pi}_h$ of all policies at grid-level $h$ [see Eq. (4.4)]. These policies are easy to deal with computationally because they are simple functions on $S_h$. Recall that if $\tilde{\mu} \in \bar{\Pi}_h$, we must have $\tilde{\mu}(x) \in \tilde{U}_h(x)$ for all $x \in S$. However, this does not always imply that $\tilde{\mu}(x) \in U(x)$; that is, we have $\tilde{\mu} \notin \Pi$, in general.

To each $\tilde{\mu} \in \bar{\Pi}_h$ we associate the operator $T_\tilde{\mu} : B(S) \mapsto B(S)$ defined by

$$T_\tilde{\mu}J(x) \overset{\text{def}}{=} g(x, \tilde{\mu}(x)) + \alpha \int_S J(y)P(y|x, \tilde{\mu}(x)) \, dy.$$  \hspace{1cm} (8.1)

[Note that, if $\tilde{\mu} \in \Pi$, this definition is consistent with our earlier definition of $T_\tilde{\mu}$; see Eq. (2.12)].

We also associate to $\tilde{\mu}$ the operator $\tilde{T}_\tilde{\mu} : B(S) \mapsto B(S)$ defined by

$$\tilde{T}_\tilde{\mu}J(x) \overset{\text{def}}{=} \tilde{g}_h(x, \tilde{\mu}(x)) + \alpha \int_S J(y)\tilde{P}_h(y|x, \tilde{\mu}(x)) \, dy.$$  \hspace{1cm} (8.2)

Similarly to $T_\mu$, $T_\tilde{\mu}$ and $\tilde{T}_\tilde{\mu}$ are monotone contraction operators and satisfy Eqs. (2.10)-(2.11). Let $J_\mu$ and $\tilde{J}_\tilde{\mu}$ be the fixed points of $T_\mu$ and $\tilde{T}_\tilde{\mu}$, respectively. Note that $J_\mu$ (respectively, $\tilde{J}_\tilde{\mu}$) can be interpreted as the expected cost functions associated with stationary policy $\tilde{\mu}$ for the original MDP (respectively, for the discretized MDP).

Definition 8.1: Let $\varepsilon > 0$. A function $\tilde{\mu} : S \mapsto C$ is called an $\varepsilon$-optimal policy if there exists some $h > 0$ such that $\tilde{\mu} \in \bar{\Pi}_h$, $\|J_\mu - J^*\|_\infty \leq \varepsilon$, and $\|\tilde{J}_\tilde{\mu} - J^*\|_\infty \leq \varepsilon$.

We now proceed to analyze the complexity of computing an $\varepsilon$-optimal policy.

8.2 Upper Bounds for Computing $\varepsilon$-optimal Policies

We will show that computing an $\varepsilon$-optimal policy is "no harder than" (within a constant factor in cost of) computing an $\varepsilon$-optimal cost function; thus, the upper bounds of Theorem 7.1 [Eqs. (7.13) and (7.14)] apply to the computation of an $\varepsilon$-optimal policy as well. To show this, we use the well known fact that the policy used in the final iteration of successive approximation algorithm is basically an $\varepsilon$-optimal policy. The proof of this result depends on the following lemma:

Lemma 8.1: Let $\tilde{J}$ be an element of $B(S)$. Suppose that $\tilde{\mu} \in \bar{\Pi}_h$ is a policy that attains the minimum in the formula for $T_h\tilde{J}$, that is,

$$T_h\tilde{J}(x) = \tilde{g}_h(x, \tilde{\mu}(x)) + \alpha \int_S \tilde{J}(y)\tilde{P}_h(y|x, \tilde{\mu}(x)) \, dy, \quad \forall x \in S.$$
(Equivalently, $T_h \hat{J} = \hat{T}_h \hat{J}$.) Then for all $h \in (0, 1/2K]$ there hold:

a) $\|J_h^* - \hat{J}_\mu\|_{\text{TV}} \leq \frac{\alpha}{1 - \alpha} \|T_h \hat{J} - \hat{J}\|_{\text{TV}}$.

b) $\|\hat{J}_\mu - J_\mu\|_{\text{TV}} \leq \frac{1}{1 - \alpha} (K_1 + \alpha K_2 \|J_\mu\|_Q) h$, where $K_1$ and $K_2$ are the constants of Theorem 3.1.

Proof:

a) Since $T_h \hat{J} = \hat{T}_h \hat{J}$, we have

$$
\|J_h^* - \hat{J}_\mu\|_{\text{TV}} \leq \|J_h^* - T_h \hat{J}\|_{\text{TV}} + \|\hat{T}_h \hat{J} - \hat{J}_\mu\|_{\text{TV}} \\
\leq \frac{\alpha}{1 - \alpha} \|T_h \hat{J} - \hat{J}\|_{\text{TV}} + \frac{\alpha}{1 - \alpha} \|\hat{T}_h \hat{J} - \hat{J}_\mu\|_{\text{TV}} \\
= 2 \frac{\alpha}{1 - \alpha} \|T_h \hat{J} - \hat{J}\|_{\text{TV}},
$$

where we have used Eq. (5.2) to obtain the second inequality. For any scalar $c$, we have $T_h(\hat{J} + c) = \hat{T}_h(\hat{J} + c)$, that is, $\hat{\mu}$ is still a minimizing policy if the function $\hat{J}$ is shifted by any constant. So,

$$
\|J_h^* - \hat{J}_\mu\|_{\text{TV}} \leq 2 \frac{\alpha}{1 - \alpha} \|T_h(\hat{J} + c) - (\hat{J} + c)\|_{\text{TV}} = 2 \frac{\alpha}{1 - \alpha} \|T_h \hat{J} - \hat{J} - (1 - \alpha) c\|_{\text{TV}}.
$$

By letting $c = \frac{1}{2(1 - \alpha)} \left[ \sup_x (T_h \hat{J} - \hat{J})(x) + \inf_x (T_h \hat{J} - \hat{J})(x) \right]$, we obtain the desired result.

b) It is clear from the definition of $\hat{T}_h$ and $T_h$ that

$$
\|\hat{T}_h J - T_h J\|_{\text{TV}} \leq (K + \frac{\alpha}{2} K_P \|J\|_Q) h, \quad \forall J \in B(S),
$$

where $K$ is the constant of Assumptions A.1–A.4 and $K_P$ is the constant of Lemma 3.1. It follows from the proof of Theorem 3.1 that $K \leq K_1$ and $\frac{1}{2} K_P \leq K_2$. So, using Lemma 3.2 we obtain

$$
\|J_h^* - \hat{J}_\mu\|_{\text{TV}} \leq \frac{1}{1 - \alpha} (K_1 + \alpha K_2 \|J_\mu\|_Q) h,
$$

as required.

q.e.d.

To use Lemma 8.1, suppose that we compute an $\varepsilon$-optimal cost function for the general case, using the multigrid successive approximation algorithm of Subsection 7.1. Let $\hat{J} = T_{h_f}^{(h_f)} \hat{J}_f$, so that $T_{h_f} \hat{J}$ corresponds to the last successive approximation iteration [cf. Eq. (7.6)]. Let $\hat{\mu}$ be a policy that attains the minimum in $T_{h_f} \hat{J}$. Then by Lemma 8.1(a),

$$
\|J_{h_f}^* - \hat{J}_\mu\|_{\text{TV}} \leq \frac{\alpha}{1 - \alpha} \|T_{h_f} \hat{J} - \hat{J}\|_Q \leq \varepsilon,
$$

where the last inequality follows from Eq. (7.6). Furthermore, since $\|J_\mu\|_Q \leq \frac{2K}{1 - \alpha}$ [cf. Eq. (3.24)] we see from Eqs. (3.27) and (7.5) that

$$
\frac{1}{1 - \alpha} (K_1 + \alpha K_2 \|J_\mu\|_Q) h_f \leq \frac{K'}{(1 - \alpha)^2} h_f \leq \frac{\varepsilon}{2}.
$$
By Lemma 8.1(b),
\[ \|J_{\tilde{\mu}} - J_\mu\|_\infty \leq \frac{\epsilon}{2}. \] (8.5)
Lastly, the choice of \( h_f \) [cf. Eq. (7.5)] ensures that the discretization error
\[ \|J^* - J_{h_f}^*\|_\infty \leq \frac{\epsilon}{2}. \] (8.6)
Using the triangle inequality and Eqs. (8.3), (8.5), (8.6), we conclude that
\[ \|J^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq \|J^* - J_{h_f}^*\|_\infty + \|J_{h_f}^* - \tilde{J}_{\tilde{\mu}}\|_\infty \leq \frac{3}{2}\epsilon, \] (8.7)
\[ \|J^* - J_{\tilde{\mu}}\|_\infty \leq \|J^* - \tilde{J}_{\tilde{\mu}}\|_\infty + \|\tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}}\|_\infty \leq 2\epsilon. \] (8.8)
(Thus, Eqs. (8.7)-(8.8) show that \( \tilde{\mu} \) is a \( 2\epsilon \)-optimal policy.) We note that a similar reasoning yields the bounds of Eqs. (8.7) and (8.8) for the special case where the accessibility condition is satisfied.

We conclude that the work needed to compute an \( \epsilon \)-optimal policy is no greater than that of computing an \( \frac{\epsilon}{2} \)-optimal cost function, and the upper bounds of Theorem 7.1 apply to the computation of an \( \epsilon \)-optimal policy.

Let us now consider the problem of computing an \( \epsilon \)-optimal admissible policy, that is, a policy \( \mu \in \Pi \) such that \( \|J_\mu - J^*\|_\infty \leq \epsilon \). This can be done, in principle, by first computing an \( \epsilon \)-optimal policy (for some smaller \( \epsilon \)) and approximating it by an element of \( \Pi \), due to the following lemma:

Lemma 8.2: Let \( J \in B(S) \), \( \mu \in \Pi \), \( \tilde{\mu} \in \tilde{\Pi}_h \), and let \( K \) be the constant of Assumptions A.1–A.4. Then,
\[ \|T_\mu J - T_{\tilde{\mu}} J\|_\infty \leq (K + \alpha K\|J\|_Q)\|\mu - \tilde{\mu}\|_\infty. \]
Furthermore,
\[ \|J_\mu - J_{\tilde{\mu}}\|_\infty \leq \frac{1}{1 - \alpha}(K + \alpha K\|J_{\tilde{\mu}}\|_Q)\|\mu - \tilde{\mu}\|_\infty. \]
Proof: The first part of the lemma follows from the fact that \( |g(x, \mu(x)) - g(x, \tilde{\mu}(x))| \) and \( |P(y|x, \mu(x)) - P(y|x, \tilde{\mu}(x))| \) are both bounded by \( K\|\mu - \tilde{\mu}\|_\infty \); the second part follows from Lemma 3.2. q.e.d.

The computation of an \( \epsilon \)-optimal admissible policy \( \mu \) proceeds as follows. We first choose a discretization parameter \( h \) which is small enough so that the discretization error \( K'h/(1 - \alpha)^2 \) is no greater that \( \frac{\epsilon}{8} \). We use the multigrid successive approximation algorithm to compute an \( \frac{\epsilon}{4} \)-optimal cost function and, according to our earlier discussion, we obtain as a by-product an \( \frac{\epsilon}{2} \)-optimal policy \( \tilde{\mu} \in \Pi_h \); that is, \( \|J^* - J_{\tilde{\mu}}\|_\infty \leq \epsilon/2 \).

We note from Lemma 4.1 that there exists some \( \mu \in \Pi \) such that \( \|\mu - \tilde{\mu}\|_\infty \leq (K + 1)h \); so, by Lemma 8.2, \( \|J_\mu - J_{\tilde{\mu}}\|_\infty \leq \frac{1}{1 - \alpha}(K + \alpha K\|J_{\tilde{\mu}}\|_Q)(K + 1)h. \) It can be seen from the proof of Theorem 3.1 that \( K(K + 1) \) is less than \( K_1 \) and \( K_2 \). Proceeding as in Eq. (8.4), we obtain \( \|J_\mu - J_{\tilde{\mu}}\|_\infty \leq \frac{\epsilon}{8} \). So, by the triangle inequality,
\[ \|J^* - J_\mu\|_\infty \leq \|J^* - J_{\tilde{\mu}}\|_\infty + \|J_{\tilde{\mu}} - J_\mu\|_\infty \leq \frac{\epsilon}{2} + \frac{\epsilon}{8} = \frac{5}{8}\epsilon. \]
Thus, $J_\mu$ is indeed an $\epsilon$-optimal admissible policy, as desired.

If the method in the preceding paragraph is to be used, we must be able, given any $\hat{\mu} \in \hat{\Pi}_h$, to compute an admissible $\mu \in \Pi$ such that $\|\hat{\mu} - \mu\|_\infty$ is smaller than $c_1 h$, for some constant $c_1$. This is, in general, impossible under our model of computation; in fact, it is even impossible, in general, to represent an element of $\Pi$ using a finite data structure. On the other hand, for problems that arise in practice, the sets $U(x)$ often have a simple structure and this task is feasible. In those cases, the computation of an $\epsilon$-optimal admissible policy is no harder than the computation of an $\epsilon$-optimal cost function.

### 8.3 Lower Bounds for Computing $\epsilon$-Optimal Policies

We observe that an $\epsilon$-optimal policy, by definition, determines the optimal cost function $J^*$ to within $\epsilon$; so, the lower bounds of Subsection 7.4 [Eqs. (7.15)-(7.16)] apply to the computation of an $\epsilon$-optimal policy as well. (See [11] for more details.) It remains to argue that computing an $\epsilon$-optimal policy is "no easier than" computing an $\epsilon$-optimal cost function (that is, the cost of computing latter is within a constant factor of the cost of computing the former).

For the special case where an accessibility condition is imposed, the upper bound for computing an $\epsilon$-optimal cost function is within a constant factor of the lower bound [cf. Eqs. (7.14) and (7.16)]. We conclude that computing an $\epsilon$-optimal policy is no easier than computing an $\epsilon$-optimal cost function. Thus, we have shown for problems satisfying an accessibility condition that computing an $\epsilon$-optimal policy is as hard as computing an $\epsilon$-optimal cost function.

We now consider the general case. For $\alpha$ fixed and we concentrate on the dependence on $\epsilon$, the upper bound for computing an $\epsilon$-optimal cost function is within a constant factor of the lower bound [cf. Eqs. (7.13) and (7.15)]. Arguing as in the preceding paragraph, we conclude that, with respect to the dependence on $\epsilon$, computing an $\epsilon$-optimal policy is as hard as computing an $\epsilon$-optimal cost function. But because of the "gap" of $O\left(\frac{1}{1-\alpha}\right)$ between the upper and lower bounds, we cannot draw the same conclusion for the dependence on $\alpha$; a different argument is needed.

The basic idea of the argument is as follows. We will show that if an $\frac{\alpha}{2}$-optimal policy is available, then an $\epsilon$-optimal cost function can be quickly computed (with complexity better than the lower bound). Thus, an algorithm can first compute an $\frac{\alpha}{2}$-optimal policy, then use the policy to compute an $\epsilon$-optimal cost function with total computational cost within some constant factor of the cost of computing the policy. It follows that computing an $\epsilon$-optimal policy is no easier than computing an $\epsilon$-optimal cost function.

To use the method described in the preceding paragraph, additional assumptions are required. First we define

$$H(\alpha, \epsilon) = \left\{ h \in (0, 1) \mid h \geq \frac{1}{8K'}(1 - \alpha)^2 \epsilon \right\}, \quad \alpha, \epsilon \in (0, 1),$$

where $K'$ is the constant of Corollary 3.1. It is clear from the discussion in Subsection 8.2 that, for any discount factor $\alpha < 1$ and accuracy parameter $\epsilon > 0$, there exists some $h \in H(\alpha, \epsilon)$ such that
\( \Pi_h \) contains an \( \epsilon \)-optimal policy. We next introduce the assumptions:

**C.1** The dimension of the control space \( m \) is at least 1.

**C.2** Given any discount factor \( \alpha < 1 \) and accuracy parameter \( \epsilon > 0 \), if \( \tilde{\mu} \) is an \( \epsilon \)-optimal policy, then \( \tilde{\mu} \in \Pi_h \) for some \( h \in H(\alpha, \epsilon) \).

Note that Assumption C.1 excludes problems with finite control space. And we “need” Assumption C.2 to ensure that the policy under consideration is not unnecessarily complicated and can be used to quickly compute an \( \epsilon \)-optimal cost function.

For the remainder of the discussion, Assumptions C.1–C.2 will be in effect. Let \( \tilde{\mu} \) be an \( \frac{\epsilon}{2} \)-optimal policy and \( J^0(x) = 0 \) for all \( x \in S \). From the successive approximation error bounds [cf. Eq. (5.5)], if

\[
J^t = \tilde{T}_\mu^t J^0 + \frac{\alpha}{2(1 - \alpha)} \left[ \min_x \{ (\tilde{T}_\mu^t J^0 - \tilde{T}_\mu^{t-1} J^0)(x) \} + \max_x \{ (\tilde{T}_\mu^t J^0 - \tilde{T}_\mu^{t-1} J^0)(x) \} \right],
\]

then

\[
\| \tilde{J}_\mu - J^t \|_\infty \leq \frac{\alpha^t}{2(1 - \alpha)} \| \tilde{T}_\mu J^0 - J^0 \|_\infty \leq \frac{\alpha^t}{1 - \alpha} K,
\]

where we have used the fact that \( \| \tilde{T}_\mu J^0 \|_\infty \leq 2K \).

We now apply \( \tilde{T}_\mu \) on \( J^0 \) for \( t \) times, where \( t \) is the smallest integer such that \( \frac{\alpha^t}{1 - \alpha} K \leq \frac{\epsilon}{2} \). This ensures that \( \| \tilde{J}_\mu - J^t \|_\infty \leq \frac{\epsilon}{2} \), and by the triangle inequality,

\[
\| J^* - J^t \|_\infty \leq \| J^* - \tilde{J}_\mu \|_\infty + \| \tilde{J}_\mu - J^t \|_\infty \leq \epsilon.
\]

Thus, \( J^t \) is an \( \epsilon \)-optimal cost function, as desired.

To bound the complexity of computing \( J^t \), it is seen (cf. Subsection 6.1) that

\[
t = O \left( \frac{\log \left( \frac{1}{(1 - \alpha)\epsilon} \right)}{1 - \alpha} \right).
\]

And by Lemma 5.1, the cost of an iteration of \( \tilde{T}_\mu \) is \( O \left( h^{-(n+m)} \right) \), which by Assumption C.2 is \( O \left( \frac{1}{(1 - \alpha)^2 \epsilon} \right) \). Thus, using the \( \frac{\epsilon}{2} \)-optimal policy, we can compute an \( \epsilon \)-optimal cost function with cost

\[
O \left( \frac{\log \left( \frac{1}{(1 - \alpha)\epsilon} \right)}{1 - \alpha} \left( \frac{1}{(1 - \alpha)^2 \epsilon} \right)^{n+m} \right),
\]

which is less than the lower bound \( \Omega \left( \left( \frac{1}{(1 - \alpha)^2 \epsilon} \right)^{2n+m} \right) \). This completes the proof that computing an \( \epsilon \)-optimal policy is no easier than computing an \( \epsilon \)-optimal cost function. Hence, we have shown for problems not assumed to satisfy an accessibility condition that computing an \( \epsilon \)-optimal policy is as hard as computing an \( \epsilon \)-optimal cost function.

Lastly, it is seen from the preceding argument that Assumption C.2 can be relaxed, with \( H(\alpha, \epsilon) \) replaced by

\[
H'(\alpha, \epsilon) \overset{\text{def}}{=} \left\{ h \in (0, 1) \mid h \geq \frac{1}{8K^2} (1 - \alpha)^{2+2i} \epsilon^i, \text{ where } i = \frac{n - 1}{n + m} \right\}, \quad \alpha, \epsilon \in (0, 1).
\]
9. EXTENSIONS.

We discuss here certain extensions of our results and provide some suggestions for future research. We will only present the main ideas and the reader is referred to [10] for more details.

9.1. Piecewise Lipschitz Continuous Dynamics

Assumption A.2 requires \( P(y|x, u) \) to be Lipschitz continuous. This assumption is unnecessarily restrictive, and rules out many interesting examples. In fact our results remain valid if \( |g(x, u)| \leq K \) for all \( x \in S, u \in C \) and Assumption A.2 is replaced by the following:

**Assumption B.2:** There exists a constant \( K \geq 1 \) such that:

(i) \( \int_S |P(y|x, u) - P(y|x', u')|dy \leq K(\|x - x'\|_\infty + \|u - u'\|_\infty) \), for all \( x, x' \in S \) and \( u, u' \in C \).

(ii) For every \( x \in S \) and \( u \in C \), \( P(y|x, u) \) is a "piecewise Lipschitz continuous" function of \( y \).

By \( P \) being "piecewise Lipschitz continuous", we mean that we can partition the state space \( S \) into a finite collection of disjoint subsets \( U_i \) such that \( P(., x, u) \) is Lipschitz continuous, with Lipschitz constant \( K \), on each set \( U_i \). Furthermore, to rule out pathological cases, we require that the sets \( U_i \) have "piecewise smooth" boundary.

With Assumption A.2 replaced by Assumption B.2, it can be shown that the discretizations of Section 3 again satisfy

\[
\int_S |P(y|x, u) - \tilde{P}_h(y|x, u)|dy \leq K_P h, \quad \forall h \in (0, h_0],
\]

(cf. Lemma 3.1) and this property is the key to the discretization error bounds of Theorem 3.1. Furthermore, any accessibility condition in the continuous problem is again inherited by the discretized problem (cf. Theorem 4.2). As a consequence, all subsequent results, as well as the complexity analysis, remain valid.

9.2. The case where \( P \) is not a probability measure

Suppose that \( |g(x, u)| \leq K \) for all \( x \in S, u \in C \). We can relax Assumption A.4 by assuming instead that there exists some constant \( K \geq 1 \) such that for all \( x, y \in S \) and \( u \in C \) we have:

(a) \( \int_S P(y|x, u) dy \leq 1 \),

(b) \( P(y|x, u) \in [0, K] \).

Such an assumption can be used to model those MDPs in which the system has some nonzero probability of entering a zero-cost absorbing state.

In an even more general class of problems, we can assume that:

\( a' \) \( \int_S |P(y|x, u)| dy \leq 1 \),

\( b' \) \( |P(y|x, u)| \in [0, K] \).

A convenient discretization rule for such problems is to let

\[
\tilde{P}_h(y|x, \bar{u}) \overset{\text{def}}{=} P(\bar{y}|\bar{x}, \bar{u}), \quad \text{if} \quad \int_S |P(\bar{y}|\bar{x}, \bar{u})|dy \leq 1,
\]
and
\[ \tilde{P}_h(y|x, \bar{u}) \overset{\text{def}}{=} \frac{P(\bar{\sigma}_y|\bar{\sigma}_x, \bar{u})}{\int_S |P(\bar{\sigma}_y|\bar{\sigma}_x, \bar{u})| dy}, \]
on \text{otherwise.}

Since \( P \) is allowed to be negative, it is clear that \( T \) and \( T_h \) are now no longer monotone operators. However, they are still contraction operators, with contraction factor \( \alpha \), and the proof of Theorem 3.1 (discretization error bounds) remains valid provided that \( \| \cdot \|_Q \) is replaced by \( \| \cdot \|_\infty \). Using the algorithm of Section 7 (with some minor modifications) it can be shown that the complexity of the multigrid successive approximation algorithm is
\[ O \left( \frac{1}{|\log \alpha|} \left( \frac{1}{(1-\alpha)^2} \right)^{2n+m} \right), \]
exactly as in the case of MDPs satisfying Assumptions A.1–A.4. Unlike the case where \( P \) corresponds to a probability measure, we cannot improve this complexity estimate by imposing an accessibility condition on \( P \). In fact, the lower bound for the case where \( P \) is a nonnegative subprobability measure and satisfies an accessibility condition is shown in [11] to be
\[ \Omega \left( \left( \frac{1}{(1-\alpha)^2} \right)^{2n+m} \right). \tag{9.1} \]

This may seem counterintuitive, given the fact that the case of a nonnegative subprobability measure can be always reduced to the case of a probability measure, by introducing an additional absorbing state to which all of the “missing” probability is channeled. The catch is that the accessibility condition is destroyed in the course of this state augmentation.

### 9.3 Fredholm Equations of the Second Kind

A Fredholm equation of the second kind is an equation of the form
\[ g(x) + \int_S G(y, x)J(y) dy = J(y), \]
where \( S \) is a bounded subset of \( \mathbb{R}^n \), \( g \) and \( G \) are given functions, and \( J \) is the unknown.

The numerical solution of this equation has been well studied (see e.g. [14], [22], [26] and the references therein). Let us assume that \( G \) is a bounded function and that \( \int_S |G(y, x)| dy \leq \alpha \) for all \( x \in S \), where \( \alpha \in (0, 1) \). If we let \( P(y|x) = G(y, x)/\alpha \), it is clear that we are dealing with the problem discussed in Section 9.2, except that the the control variable \( u \) is absent. (Thus, \( m = 0 \).) It follows that (under Lipschitz continuity assumptions) our multigrid algorithm can be used to compute an \( \varepsilon \)-approximation of the solution and has complexity
\[ O \left( \frac{1}{|\log \alpha|} \left( \frac{1}{(1-\alpha)^2} \right)^{2n} \right). \]
Furthermore, the lower bound of Eq. (9.1) becomes

$$\Omega \left( \left[ \frac{1}{(1 - \alpha)^2 \epsilon} \right]^{2n} \right),$$

and therefore our algorithm is optimal as far as the dependence on $\epsilon$ is concerned.

Multigrid algorithms for Fredholm's equation can also be found in [14] and [22], and they are different in the following respects. First, the algorithms in these references are more general because they do not require a contraction assumption. Furthermore, these algorithms perform computations on fine grids and then use certain coarse-grid corrections. This is in contrast to our method that only proceeds from coarse to fine grids. According to our results, for the problems we are considering, our method has optimal dependence on the accuracy parameter $\epsilon$ and close to optimal dependence on $\alpha$. (Note that $\alpha$ can be viewed as a measure of ill-conditioning of the problem.) It is unclear whether the algorithms in [14] have any similar optimality properties.

9.4. Different Norms

Let us consider the $L_p$-norm on $B(S)$ defined by

$$\|J\|_p \overset{\text{def}}{=} \left[ \int_S |J(y)|^p dy \right]^{1/p}, \quad p \in [1, \infty).$$

Since the volume of $S$ is bounded by 1, it is easily shown that $\|J\|_p \leq \|J\|_\infty$ for any $J \in B(S)$ and any $p \in [1, \infty)$. For this reason, the function $J$ returned by our algorithms automatically satisfies $\|J - J^*\|_p \leq \epsilon$.

It also turns out [11] that the lower bounds on the computational complexity of the problem do not change when $L_p$-norms are used to measure the error $J - J^*$. It follows such a different choice of norms does not affect the optimality properties of our algorithms.

9.5. Average Cost Problems

Our results can be extended to the case of average cost Markov Decision Problems. In particular, under an accessibility condition optimal algorithms can be obtained. On the other hand, without an accessibility condition, average cost problems are, in general, ill-posed and have infinite computational complexity. It is an interesting research problem to find conditions that are weaker than accessibility and that guarantee well-posedness.

9.6. Another Formulation of Discrete-Time Stochastic Control Problems

In an alternative formulation of discrete-time stochastic control, we are given a dynamical equation of the form

$$x_{k+1} = f(x_k, u_k, w_k),$$

39
where $k$ denotes the time index, $x_k$ denotes the state, $u_k$ the control, and $w_k$ denotes a noise term with known probability density $Q(w_k|x_k, u_k)$. Even though such problems can be reformulated into our framework, the resulting density $P(\cdot|x_k, u_k)$ is not, in general, Lipschitz continuous. In particular, our results do not apply. An important special case in which our results are inapplicable is the case of deterministic systems in which $P$ corresponds to a singular measure, as opposed to a density. The problem of characterizing the best possible discretization error and the design of optimal (or close to optimal) algorithms for such problems is open to the best of our knowledge.

9.7. Some Practical Issues.

Although our algorithm has excellent theoretical properties, a lot of systematic experimentation is needed to assess its practicality. Furthermore, in a practical implementation, several modifications are worth investigating.

a) Different discretization procedures can be tried in an effort to exploit any additional smoothness in the problem data.

b) Many practical problems involve unbounded state spaces and ways must be found to handle such problems.

c) Whereas our algorithm uses a priori bounds to decide when to change grid-level, one might be able to use information generated by the algorithm and improve performance. In particular, one could try to generate estimates of the degree of smoothness of $J^*$, while the algorithm is running.

d) Finally, the implementation of the "oracle calls" could present several challenges. This is true especially for the oracle calls that provide volume estimates and that generate the sets $\tilde{U}_k(x)$ of admissible controls for the discretized problem.

e) In practice, the running time of successive approximation can be improved by using Gauss-Seidel iterations, and doing a Jacobi iteration only when successive approximation error bounds are needed.

9.8. Conclusions

We have studied the computational requirements of continuous-state Markov Decision Problems and have obtained some fairly definite conclusions, by presenting algorithms with certain optimality properties. There are several problems that remain to be addressed, having to do with alternative formulations (Subsection 9.6), continuous-time formulations, algorithmic implementation issues (Subsection 9.7). We see our work as a contribution to the understanding of the computational issues associated with control theory. Such issues are important because they will ultimately determine the practicality of different faces of control theory.
REFERENCES


