Numerical Schemes for the Continuous Q-function of Reinforcement Learning

Stephan Pareigis*

Abstract

We develop a theoretical framework for the problem of learning optimal control. We consider a discounted infinite horizon deterministic control problem in the reinforcement learning context. The main objective is to approximate the optimal value function of a fully continuous problem, using only observed information as state, control, and cost. With results from the numerical treatment of the Bellman equation we formulate regularity and consistency results for the optimal value function. These results help to construct algorithms for the continuous problem. We propose two approximation schemes for the optimal value function which are based on observed data. The implementation of a simple optimal control learning problem shows the effects of the two approximation schemes.

AMS Subject Classification: 49L20, 65N30, 68T05, 93C57

Keywords: learning optimal control, dynamic programming, reinforcement learning, sampled data, approximation of optimal value function

Running Head: Numerical Schemes for the Continuous Q-Function

1 Introduction

In many optimal control problems a solution cannot be obtained by standard numerical methods. This may be due to very large state spaces (as in game playing) or incomplete information such as unknown system dynamics. The mathematical model of the problem may also be either too complicated to handle or too simple to be accurate enough.

*stp@numerik.uni-kiel.de, Lehrstuhl für Praktische Mathematik, Christian-Albrechts-Universität Kiel, D-24098 Kiel, Germany, tel.: +49-431-880 1421

Lately, a promising method called reinforcement learning has been successfully applied to problems in all kinds of application areas ([2], [6]). The optimal value function is learned by letting the real system or a simulation perform the system dynamics and provide the cost. The optimal value function is then approximated using the information provided by the system: state, cost and control. This computation may be off-line as in game playing (the computer gains experience by playing against himself), or on-line as in heavy traffic problems (the computer learns about the dynamics of a data or traffic network by actually controlling it or by controlling a simulation).

A lot of work has been done on reinforcement learning where the underlying system is assumed to be discrete (in time and space), see e.g. [1], [5], [18], [19]. The problems considered here are Markov Decision Problems. Generally three different types for realizing reinforcement learning can be distinguished. In Real Time Dynamic Programming (RTDP) the system dynamic is assumed to be known. The optimal value function is approximated by letting the system perform time steps and applying dynamic programming in each step. In Adaptive RTDP (ARTDP) a parameterized model of the system is stored internally and updated according to the actions of the real system. Q-learning uses no model of the system. Instead, a somewhat augmented value function, called the Q-function, is approximated.

Recently, achievements have been made in generalizing the reinforcement learning concept from Markov Decision Problems to continuous state spaces and time, see e.g. [3], [9], [14]. The problem here seems to be, that the computationally necessary discrete observation of a continuous process (in time and space) produces an additional error, apart from the approximation error that comes from the function approximation. A rigorous formulation of this error is important for the construction of learning algorithms, especially for local error estimation as a stopping criterion for learning or for adaptive grid refinement.

Our approach is similar to the numerical treatment of the Bellman equation as in [4], [8] and [13]. We use these numerical methods to generalize the Q-learning concept to work with fully continuous systems.

We first discretize the process in time only (section 2). We define the discrete time data which is used by the learning agent to control the (discrete time) system. A semi-continuous version of the Q-function is defined, and a regularity and consistency result is presented.

In section 3 we introduce a state space discretization as used in numerical treatment of the Bellman equation (see [10]). This way the trial space for the approximation of the semi-continuous Q-function may be defined.

Two algorithms are proposed in section 4 using this kind of function approximation in Q-learning. Theoretical results are obtained, concerning consistency and error estimation.

Section 5 compares the two algorithms in a numerical experiment.

A crucial point is the choice of control during the learning period. We will not discuss this question here and assume, that the control is performed by a separate device (another program, a PID controller, a person etc). The numerical experiments in section 5 use a certainty equivalence controller with random jumps.

2 Formulation of the learning problem

Let $G \subset \mathbb{R}^n$ be a bounded state space and $A \subset \mathbb{R}^m$ a compact set of actions. We shall assume that the process to be controlled is described by the differential equation

$$\begin{aligned} \dot{\varphi}(\tau) &= f(\varphi(\tau), a(\tau)), \\ \varphi(0) &= x \in G. \end{aligned} \tag{1}$$

We will denote a solution of this equation with

$$\varphi_{x,a(.)} : \mathbb{R}_+ \to G.$$

We want to find a control function $a(.) \in \mathcal{A} = \{b : [0, \infty[\to A,\ b \text{ measurable } \}$, which minimizes the functional

$$J(x, a(.)) := \int_0^\infty e^{-\rho\tau} g(\varphi_{x,a(.)}(\tau), a(\tau)) d\tau, \tag{2}$$

where $\rho > 0$ is the discount rate and

$$g : G \times A \to \mathbb{R}_+$$

is a cost function. We define the reinforcement $r_t : G \times \mathcal{A} \to \mathrm{I\!R}_{\geq 0}$ for $t \in [0, \infty]$ as

$$r_t(x, a(.)) = \int_0^t e^{-\rho\tau} g(\varphi_{x,a(.)}(\tau), a(\tau)) d\tau. \tag{3}$$

The optimal value function is defined as

$$V(x) := \inf_{a(.)\in\mathcal{A}} J(x, a(.)) = \inf_{a(.)\in\mathcal{A}} r_\infty(x, a(.)). \tag{4}$$

In the following we will assume some regularity conditions on f and g.

Proposition 2.1 *Let f and g be Lipschitz-continuous*

$$\begin{aligned} |f(x_1, a) - f(x_2, a)| &\leq L_f |x_1 - x_2| \\ |g(x_1, a) - g(x_2, a)| &\leq L_g |x_1 - x_2| \end{aligned} \tag{5}$$

for constants $L_f, L_g > 0$ independent of $a \in A$. Then the Gronwall-Lemma gives the following estimate for any to points $x_1, x_2 \in G$, $\tau > 0$ and $a(.) \in \mathcal{A}$

$$|g(\varphi_{x_1,a(.)}(\tau), a(\tau)) - g(\varphi_{x_2,a(.)}(\tau), a(\tau))| \leq L_g |x_1 - x_2| e^{L_f \tau}.$$

This also implies the Lipschitz-continuity of $r_t(.,.)$ in the first component

$$|r_t(x_1, a) - r_t(x_2, a)| \leq L_g |x_1 - x_2| \frac{e^{(L_f - \rho)t} - 1}{L_f - \rho} =: L_r |x_1 - x_2|.$$

□

We shall also assume, that f and g are bounded

$$\begin{aligned} f : G \times A \to \mathrm{I\!R}^n &\quad \text{is bounded by} \quad ||f(x, a)|| \leq M_f \quad \forall x, a, \\ g : G \times A \to \mathrm{I\!R} &\quad \text{is bounded by} \quad 0 \leq g(x, a) \leq M_g \quad \forall x, a \end{aligned} \tag{6}$$

It is known that under the above assumptions V is the unique viscosity solution of the continuous time, continuous state Bellman Equation

$$\inf_{a\in A}\{DV(x) f(x, a) - \rho V(x) + g(x, a)\} = 0, \quad x \in G. \tag{7}$$

For viscosity solutions see [11] where also further literature is given. Numerical solutions for equation (7) have been studied by various authors, see e.g. [7], [8], [10].

We will assume, that the functions f and g are completely unknown to the controlling agent (only their existence is assumed, however). The discount factor ρ shall be known. The controlling agent therefore has the possibility to control the system and observe the outcome. He shall control in time steps of size $h > 0$, using control functions $a \in A_h$, where $A_h \subset \{b : b \in \mathcal{A}|_{[0,h[}\}$ a finite subset. At discrete time step $n \in \mathbb{N}$ he gains the following information

- the current state $y_n \in G$,
- an action $a_n \in A_h$,
- the subsequent state $y_{n+1} := \varphi_{y_n,a_n}(h)$
- the local cost $r_n := r_h(y_n, a_n) = \int_0^h e^{-\rho\tau} g(\varphi_{y_n,a_n}(\tau), a_n(\tau))d\tau$.

Note, that it may be assumed that the cost has the given (local) form. If only the total accumulated, discounted reinforcement

$$\tilde{r}_n = \int_0^{(n+1)h} e^{-\rho\tau} g(\varphi_{x,a}(\tau), a(\tau))d\tau$$

for some starting point $\varphi(0) = x$ was given, then with the knowledge of ρ we could calculate

$$r_n = e^{\rho n h}(\tilde{r}_n - \tilde{r}_{n-1}).$$

We will use the Q-learning approach to reinforcement learning. This means, that neither the system nor the cost function is being identified by the observed data and no model for the system is being used (Q-learning is sometimes defined to be a model-free way of reinforcement learning). Instead, the optimal value function is learned directly via the Q-function. Although being quite memory efficient, Q-learning has the disadvantage of converging very slowly. We use it here, because theoretical investigations are quite convenient. Many faster algorithms can be constructed, using more memory (eligibility traces, models for system or cost etc., see also [1]). Our results extend easily to these cases.

We now define the semi-discrete optimal value function V_h and the Q-function Q_h.

Definition 2.2 and Theorem *Let $B(G,\mathbb{R}) := \{v \in Map(G,\mathbb{R}) : v \text{ bounded } \}$ be the space of bounded functions on G. We define the operator*

$$T_h : B(G,\mathbb{R}) \to B(G,\mathbb{R})$$

$$(T_h v)(x) = \min_{a \in A_h} \{r_h(x,a) + e^{-\rho h} v(\varphi_{x,a}(h))\}. \tag{8}$$

Then there is a unique $V_h \in B(G,\mathbb{R})$ with

$$T_h V_h = V_h \quad \text{ and } \quad \sup_{x \in G} |V_h(x)| \leq \frac{M_g}{\rho}. \tag{9}$$

The Q_h-function is now defined as

$$\begin{aligned} Q_h &: G \times A_h \to \mathbb{R}_+, \\ Q_h(x,a) &:= r_h(x,a) + e^{-\rho h} V_h(\varphi_{x,a}(h)). \end{aligned} \tag{10}$$

We also denote elements of A_h as a. Note, that they are A-valued functions on $[0,h]$.

Proof. We show that T_h is a contraction on $B(G,\mathbb{R})$. Clearly, $B(G,\mathbb{R})$ is a Banach-space with norm

$$||v|| = \sup_{x \in G} |v(x)|.$$

Let $v, w \in B(G,\mathbb{R})$. Then for all $x \in G$ there is a control $a \in A_h$ such that

$$\begin{aligned} (T_h v)(x) - (T_h w)(x) &\leq r_h(x,a) + e^{-\rho h} v(\varphi_{x,a}(h)) - r_h(x,a) - e^{-\rho h} w(\varphi_{x,a}(h)) \\ &\leq \sup_{y \in G} e^{-\rho h}(v(y) - w(y)). \end{aligned}$$

Therefore we have

$$||T_h v - T_h w|| \leq e^{-\rho h} ||v - w||.$$

The boundedness in $||.||$-norm is clear, since g is bounded and we have for an arbitrary control function $a(.)$

$$V_h(x) \leq \int_0^\infty e^{-\rho t} g(\varphi_{x,a(.)}(t), a(t)) dt \leq M_g \int_0^\infty e^{-\rho t} dt = \frac{M_g}{\rho}.$$

□

The following corollary follows easily from the definition.

Corollary 2.3 *From (9) we have immediately*

$$V_h(x) = \min_{a \in A_h} Q_h(x,a). \tag{11}$$

□

The Q_h-function is introduced here, because it allows an iteration for approximation of the value function without using a model of the system and the cost, but only the observed information y_n, a_n, y_{n+1}, r_n. This can be seen when substituting $V_h(\varphi_{x,a}(h))$ in (10) with $\min_{a \in A_h} Q_h(\varphi_{x,a}(h), a)$. Using only the observed information

$$y_n = x, \quad a_n = a, \quad y_{n+1} = \varphi_{x,a}(h), \quad r_n = r_h(x,a),$$

then (10) has the following form

$$Q_h(y_n, a_n) = r_n + e^{-\rho h} \min_{a \in A_h} Q_h(y_{n+1}, a).$$

Note, that for the application of T_h as in (8), the complete knowledge of $r_h(.,.)$ is necessary. In the following Lemma we show that Q_h is itself a fixed point of a contraction.

Lemma 2.4 *Define the space of bounded functions on $G \times A_h$ (the space of Q-functions)*

$$B(G \times A_h) := \{q(.,.) : G \times A_h \to \mathbb{R}, \sup_{x \in G, a \in A_h} q(x,a) < \infty\}.$$

Define the operator

$$P_h : B(G \times A_h) \to B(G \times A_h)$$

$$(P_h q)(x,a) = r_h(x,a) + e^{-\rho h} \min_{b \in A_h} q(\varphi_{x,a}(h), b), \quad x \in G,\ a \in A_h.$$

Then the iteration

$$q_{i+1} = P_h q_i$$

for any starting point $q_0 \in B(G \times A_h)$ converges to a unique fixed point $q_h \in B(G \times A_h)$ and

$$q_h = Q_h.$$

Proof. The proof uses the contraction property of P_h with respect to the norm

$$||q|| = \sup_{x\in G, a\in A_h} q(x,a).$$

It is clear, that $B(G\times A_h)$ is a Banach space. The last assertion follows from the uniqueness of the fixed point, when equation (11) is used in (10). □

The next lemma proves a regularity result for V_h.

Lemma 2.5 *Let $\rho > L_f$. Then $V_h \in C^{0,1}(G)$ and*

$$|V_h|_{0,1} = \sup_{x\neq y} \frac{|V_h(x) - V_h(y)|}{|x-y|} \leq L_V := \frac{L_g}{\rho - L_f}.$$

Proof. For the boundedness in the $|.|_{0,1}$-semi norm we show, that for any $V \in B(G)$ with $|V|_{0,1} \leq L_V$ we also have $|T_hV|_{0,1} \leq L_V$. From the uniqueness of the fixed point of T_h and the closedness of $C^{0,1}(G) \cap B(G) \subset B(G)$ the assertion follows. Let $|V|_{0,1} \leq L_V$ and $x, y \in G$, $x \neq y$. Then

$$\begin{aligned}
|(T_hV)(x) - (T_hV)(y)| &\leq \max_{a\in A_h} \Big\{ \int_0^h e^{-\rho\tau} |g(\varphi_{x,a}(\tau), a(\tau)) - g(\varphi_{y,a}(\tau), a(\tau))| d\tau \\
&\quad + e^{-\rho h} |V(\varphi_{x,a}(h)) - V(\varphi_{y,a}(h))| \Big\} \\
&\leq \int_0^h L_g |x-y| e^{(L_f-\rho)\tau} d\tau + e^{-\rho h} L_V |x-y| e^{L_f h} \\
&= L_g |x-y| \frac{e^{(L_f-\rho)h} - 1}{L_f - \rho} + L_V |x-y| e^{(L_f-\rho)h}.
\end{aligned}$$

Division with $|x-y|$ and substitution of L_V gives

$$|T_hV|_{0,1} \leq \frac{L_g}{\rho - L_f} = L_V.$$

□

The lemma shows, that V_h has the same regularity as the solution of the semi-continuous Bellman-equation if it is discretized as in [8]. Note, that V_h uses exact information for cost and subsequent state (see definition of T_h), while in [8] the local cost and subsequent state are approximated (by trapezoidal rule and Euler-step). Similar estimates as in [8] can be easily shown, i.e. regularity of V_h for $\rho = L_f$ and $\rho < L_f$.

As expected, V_h converges to the viscosity solution of the fully continuous Bellman-equation if $h \to 0$. The proof is held short, since it follows the same ideas as in [7]. For the definition of a viscosity solution we also refer to [7].

Theorem 1 *For every $h \in \mathbb{R}_+$ define*

$$A_h^c = \{a : [0, h[\to A \textit{ constant } \}.$$

Then for the value function V_h with respect to A_h^c we have $V_h \to V$ uniformly in G as $h \to 0$, where V is the viscosity solution of

$$\inf_{a \in A} \{Dv(x)f(x,a) - \rho v(x) + g(x,a)\} = 0, \quad x \in G. \tag{12}$$

Proof. The proof follows essentially the argumentation of the proof of Theorem 2.2 in [7]. We first have from Theorem 2.5 and the Arzelà-Ascoli compactness criterion, that for some subsequence $h_p \to 0$ as $p \to \infty$,

$$V_{h_p} \to V, \qquad \text{locally uniformly in } \mathbb{R}^n.$$

For easier notation we will call this subsequence h and let $h \to 0$. We will now check one inequality in the definition of viscosity solution of (12), namely, let $\phi \in C^1(\mathbb{R}^n)$, and $V - \phi$ take a local maximum in x, then

$$\max_{a \in A} \left\{ \rho V(x) - \sum_{i=1}^{n} \frac{\partial \phi(x)}{\partial x_i} f(x,a) - g(x,a) \right\} \leq 0.$$

Take $\phi \in C^1(\mathbb{R}^n)$ and assume x_0 is a local maximum for $V - \phi$. In some closed ball B centered at x_0 we have

$$x_h \to x_0, \quad h \to 0,$$

where x_h is a maximum point for $V_h - \phi$ on B. Then for any $a \in A_h^c$ the point $\varphi_{x_h,a}(h)$ is in B, provided h is small enough. Therefore we have

$$V_h(x_h) - \phi(x_h) \geq V_h(\varphi_{x_h,a}(h)) - \phi(\varphi_{x_h,a}(h)),$$

for all $a \in A_h^c$. This leads to

$$\begin{aligned} 0 &= \max_{a \in A_h^c} \{V_h(x_h) - e^{-\rho h} V_h(\varphi_{x_h,a}(h)) - \int_0^h e^{-\rho\tau} g(\varphi_{x_h,a}(\tau), a(\tau)) d\tau\} \\ &\geq \max_{a \in A_h^c} \{\phi(x_h) - \phi(\varphi_{x_h,a}(h)) + \\ &\quad +(1 - e^{-\rho h}) V_h(\varphi_{x_h,a}(h)) - \int_0^h e^{-\rho\tau} g(\varphi_{x_h,a}(\tau), a(\tau)) d\tau\}. \end{aligned}$$

Since $\phi \in C^1(\mathbb{R}^n)$ for some $0 \leq \zeta_h^a \leq h$ we have

$$0 \geq \max_{a \in A_h^c} \left\{ -\sum_{i=1}^n \frac{\partial \phi}{\partial x_i}(\varphi_{x_h,a}(\zeta_h^a)) + \frac{1 - e^{-\rho h}}{h} V_h(\varphi_{x_h,a}(h)) - \frac{1}{h} \int_0^h e^{-\rho\tau} g(\varphi_{x_h,a}(\tau), a(\tau)) d\tau \right\}.$$

Passing to the limit $h \to 0$ gives the assertion. Equally we show the other inequality in the definition of the viscosity solution. With the uniqueness of the viscosity solution we have $V_h \to V$ independent of the subsequence and the proof is completed. □

3 State space discretization

According to [10] we now discretize the state space with simplices $\{S_j\}$. Define the discretization parameter

$$k = \max_j \{\text{diam } S_j\}.$$

We denote the vertices of the simplices as $\{x_j\}_{j=0}^N$. We also define

$$G^k := \{x_0, \ldots, x_N\}, \quad \text{ and } X := (x_0, \ldots, x_N),$$

the set and the tuple of all vertices.

The trial space for the optimal value function will be the set $\mathcal{V}^k$ of all continuous, piecewise affine functions $w : G \to \mathbb{R}$

$$\mathcal{V}^k := \{w \in C(G) | \nabla w(x) = c_j \text{ in } S_j\},$$

for constants c_j. The approximate Q-functions will be taken from the space

$$\mathcal{W}_h^k := \{q(.,.) : G \times A_h \to \mathbb{R}, \forall a \in A_h : q(., a) \in \mathcal{V}^k \text{ and } \sup_{x \in G, a \in A_h} q(x,a) < \infty\}.$$

Both $\mathcal{V}^k$ and $\mathcal{W}_h^k$ are Banach-spaces with respect to the norms

$$||w|| := \sup_{x_i \in G^k} |w(x_i)| \text{ and } ||q|| := \sup_{x_i \in G^k, a \in A} |q(x_i, a)|.$$

Similarly to the discrete time case it is also shown, that the operator

$$P_h^k : \mathcal{W}_h^k \to \mathcal{W}_h^k$$

$$P_h^k(q)(x,a) = r_h(x,a) + e^{-\rho h} \min_{b \in A_h} q(\varphi_{x,b}(h), b), \quad x \in G^k, a \in A_h$$

has a unique fixed point $Q_h^k \in \mathcal{W}_h^k$. P_h^k updates the function q in all vertices $x \in G^k$ and $a \in A_h$ simultaneously. This corresponds to a Jacobi-type iteration. We define the fully-discrete optimal value function

$$V_h^k \in \mathcal{V}^k, \quad V_h^k(x) = \min_{a \in A_h} Q_h^k(x,a). \tag{13}$$

We will now show, that for $k \to 0$ we have

$$||V_h^k - V_h|| \to 0,$$

and we also give an estimate for $||V_h^k - V_h||$.

Theorem 2 *Let $\rho > L_f$, $h \in]0, \frac{1}{\rho}[$. Then it holds that*

$$\sup_{x \in G, a \in A_h} |V_h^k(x,a) - V_h(x,a)| \leq \frac{L_g}{C\rho(\rho - L_f)} \frac{k}{h}, \tag{14}$$

for some constant C.

Proof. Let $x \in G$ with representation $x = \sum_{i=1}^N \mu_i x_i$ in barycentric coordinates. Since $V_h^k \in \mathcal{V}^k$, we have $V_h^k(x) = \sum_{i=1}^N \mu_i V_h^k(x_i)$. Then

$$\left|V_h^k(x) - V_h(x)\right| \leq \sum_{i=1}^N \mu_i |V_h^k(x_i) - V_h(x_i)| + \sum_{i=1}^N \mu_i |V_h(x_i) - V_h(x)|. \tag{15}$$

From (13) we have at the vertices x_i for some control $a \in A_h$

$$\begin{aligned} |V_h^k(x_i) - V_h(x_i)| &\leq e^{-\rho h} |V_h^k(\varphi_{x_i,a}(h) - V_h(\varphi_{x_i,a}(h))| \\ &\leq e^{-\rho h} \sup_{y \in G} |V_h^k(y) - V_h(y)|. \end{aligned}$$

The second expression in (15) may be estimated with the Lipschitz-continuity of V_h. This gives

$$|V_h(x_j) - V_h(x)| \leq L_V k.$$

Together we have

$$\sup_{x \in G} \left| V_h^k(x) - V_h(x) \right| \leq \frac{L_V k}{1 - e^{-\rho h}}.$$

Bounding $e^{-\rho h}$ from above by $e^{-\rho h} \leq h\rho(\frac{1}{e} - 1) + 1$, for $h \in]0, \frac{1}{\rho}[$, and taking as Lipschitz-constant $L_V = \frac{L_g}{\rho - L_f}$, the assertion follows with $C = 1 - \frac{1}{e}$. □

The methods of proof as developed in [12] can be applied here and yield the following theorem.

Theorem 3 *Let $\rho > L_f$, $h \in]0, \frac{1}{\rho}[$. Then*

$$||V - V_h^k|| \leq C(\sqrt{h} + \frac{k}{\sqrt{h}}),$$

where C is a constant independent of k and h.

□

4 Algorithms for learning

We now turn our attention to the actual real time learning algorithm. The problem here is, that information at the vertices $\{x_i\}$ is not always available as demanded by the operator P_h^k defined in the last section. The update-algorithm itself cannot choose the position of the sample points and will therefore have to work with information at arbitrary points in G. We give two possible solutions for updating the ν^{th}-iterate q^ν of the Q-function Q_h^k in points $\{x_i\}$, when information is given in arbitrary points.

Definition 4.1 *We call $y \in G$ a* sample point *at time τ, if the following information is available.*

$$\begin{array}{lcl} y \in G & : & \textit{state at time } \tau, \\ a \in A_h & : & \textit{action at time } \tau, \\ z \in G & = & \varphi_{y,a}(h), \ \textit{state at time } \tau + h, \\ r \in \mathbb{R}_+ & = & \int_0^h e^{-\rho\tau} g(\varphi_{y,a}(\tau), a) d\tau, \ \textit{local cost} . \end{array}$$

Let $G^k = \{x_1, \dots, x_N\}$ and $X = (x_1, \dots, x_N)^T$, as before. For any $x \in G$ let $\Lambda(x) = (\lambda_1(x), \dots, \lambda_N(x))^T$ be the barycentric coordinates of x with respect to X, i.e.

$$x = \sum_{i=1}^{N} \lambda_i(x) x_i = \Lambda(x)^T X.$$

We will also write for functions $q \in \mathcal{W}_h^k$

$$q(a) := (q(x_1, a), \dots, q(x_n, a))^T. \tag{16}$$

We then get for any $x \in G$, $a \in A_h$

$$q(x, a) = \Lambda(x)^T q(a).$$

We will also write

$$\min(q) := (\min_b q(x_1, b), \dots, \min_b q(x_N, b))^T,$$

i.e. if q^ν is the current iterate of the Q-function, then $\min(q^\nu)$ is the current iterate of the optimal value function.

4.1 Algorithm with information in vertices

Lets assume, that y is a sample point and coincides with a vertex x_i. In every time step we then have the following algorithm.

```
Q − VERTEX(y, a, z, r)
    determine $i \in \{1, \dots, N\} : y = x_i$
    calculate $\Lambda(z)$
    $v_{tmp} := \min_{b \in A_h} \Lambda(z)^T q^\nu(b)$
    update $q^{\nu+1}(x_i, a) := r + e^{-\rho h} v_{tmp}$                (17)
```

This update corresponds to an operator

$$P^{\mathtt{ve}}_{y,a} : \mathcal{W}_h^k \to \mathcal{W}_h^k,$$

$$(P^{\mathtt{ve}}_{y,a}q)(x,b) = r + e^{-\rho h}\Lambda^T(z)\min(q), \quad \text{if} \quad x = y \text{ and } b = a$$
$$(P^{\mathtt{ve}}_{y,a}q)(x,b) = q(x,b), \quad \text{if} \quad x \neq y \text{ and } b \neq a.$$

$P^{\mathtt{ve}}_{y,a}$ is only defined if $y = x_i$ for some $i = 1, \ldots, N$. It only updates the function q in the sample point y and control a. Since cost r and subsequent state z depend on starting state y and control a, the operator $P^{\mathtt{ve}}_{y,a}$ does not depend explicitly on r and z (this is different in a stochastic setting, where the cost and the subsequent state depend stochastically on y and a).

Note, that $P^{\mathtt{ve}}_{y,a}$ is not a strict contraction with respect to the *sup*-norm. (contraction number 1), since it acts only upon the vertex $x_i = y$. The following definition defines a contraction

$$P^{\mathtt{ve}} := \Pi_{y\in G^k}\Pi_{a\in A_h}P^{\mathtt{ve}}_{y,a} = \ldots \circ P^{\mathtt{ve}}_{x_i a} \circ \ldots,$$

where every combination of $x_i \in G^k$ and $a \in A_h$ appears exactly once. This corresponds to a Gauß-Seidel-type iteration. We do not show, that the operator $P^{\mathtt{ve}}$, which depends on the order of the operators $P^{\mathtt{ve}}_{y,a}$, is a contraction. It is easily seen, that the fixed point of $P^{\mathtt{ve}}$ is the same as of P^k_h.

4.2 Algorithm with information in arbitrary points

Kaczmarz-Algorithm. Now we want to generalize $P^{\mathtt{ve}}_{y,a}$ for arbitrary $y \in G$. Given a sample point $y \in G$ with coordinates $\Lambda(y)$ and $a \in A_h$. The coordinates of z shall be $\Lambda(z)$. We then define the operator (in vector notation as in (16))

$$(P^{\mathtt{ka}}_{y,a}q)(a) = q(a) + [r + e^{-\rho h}\Lambda^T(z)\min(q) - \Lambda^T(y)q(a)]\frac{\Lambda(y)}{\Lambda^T(y)\Lambda(y)}, \tag{18}$$

The value $r + e^{-\rho h}\Lambda^T(z)\min(q)$ may be called the update value. The difference from the old value $\Lambda^T(y)q(a)$ to the update value is weighted with the barycentric coordinates $\Lambda(y)$, and added to the values of $q(a)$ in the vertices of the simplex which contains the sample point. The Kaczmarz-update has the property, that the new iterate $q(a)$ assumes the update value at the position y. It holds that

$$\begin{aligned}(P^{\mathtt{ka}}_{y,a}q)(y,a) &= \Lambda^T(y)((P^{\mathtt{ka}}_{y,a}q)(a)) \\ &= \Lambda^T(y)q(a) + [r + e^{-\rho h}\Lambda^T(z)\min(q) - \Lambda^T(y)q(a)]\frac{\Lambda^T(y)\Lambda(y)}{\Lambda^T(y)\Lambda(y)} \quad (19)\end{aligned}$$

$$= \quad r + e^{-\rho h}\Lambda^T(z)\min(q).$$

The Kaczmarz-algorithm and the algorithm with information in vertices are identical, if the sample point y is a vertex. If $y = x_i$ for some $x_i \in G^k$, then $\Lambda(y) = e_i$, the i-th unit vector, and we have

$$(P^{\mathbf{ka}}_{y,a}q)(a) = r + e^{-\rho h}\Lambda^T(z)\min(q) = (P^{\mathbf{ve}}_{y,a}q)(a). \tag{20}$$

In every time step the Kaczmarz-algorithm takes the following form.

```
Q − KACZMARZ(y, a, z, r)
    calculate $\Lambda(y)$
    calculate $\Lambda(z)$
    $v_{tmp} := \Lambda^T(z)\min(q)$
    update $q^{\nu+1}(a) := q^{\nu}(a) + (r + e^{-\rho h}v_{tmp} - \Lambda^T(y)q^{\nu}(a))\frac{\Lambda(y)}{\Lambda^T(y)\Lambda(y)}$        (21)
```

Since $\Lambda(y)$ is the vector of barycentric coordinates of y with respect to X, it has only $n+1$ non-zero entries (number of vertices of a simplex), where n is the dimension of the state space $G \subset \mathrm{I\!R}^n$. The positions in which $\Lambda(y)$ is zero, do not contribute to the new value $q^{\nu+1}$.

Lemma 4.2 *The Kaczmarz-algorithm can generate an approximation with arbitrarily high* $||.||$*-norm, although the function to approximate is bounded.*

Proof. Consider the interval $[0, 1]$, and let $u : [0, 1] \to \mathrm{I\!R}$ be the function to approximate. We assume that we have only two vertices at $\xi_0 = 0$ and $\xi_1 = 1$. The approximating function shall be $q(\xi) = q_0 + q_1\xi$ on $[0, 1]$. For any two (distinct) data points $\zeta_0, \zeta_1 \in [0, 1]$ it holds, that with the Kaczmarz-algorithm applied alternately at ζ_0 and ζ_1, the sequence of iterates

$$q^{\nu}(\xi) = q_0^{\nu} + q_1^{\nu}\xi, \quad \nu = 0, 0.5, 1, 1.5, 2\ldots$$

with

$$q_0^{\nu+0.5} \quad = \quad q_0^{\nu} + (u(\zeta_0) - q^{\nu}(\zeta_0))\frac{1-\zeta_0}{(1-\zeta_0)^2\zeta_0^2},$$

$$\begin{aligned} q_1^{\nu+0.5} &= q_1^\nu + (u(\zeta_0) - q^\nu(\zeta_0))\frac{\zeta_0}{(1-\zeta_0)^2\zeta_0^2}, \\ q_0^{\nu+1} &= q_0^{\nu+0.5} + (u(\zeta_1) - q^{\nu+0.5}(\zeta_1))\frac{1-\zeta_1}{(1-\zeta_1)^2\zeta_1^2}, \\ q_1^{\nu+1} &= q_1^{\nu+0.5} + (u(\zeta_1) - q^{\nu+0.5}(\zeta_1))\frac{\zeta_1}{(1-\zeta_1)^2\zeta_1^2} \end{aligned}$$

will converge to a $q(.)$ such that $q(\zeta_0) = u(\zeta_0)$ and $q(\zeta_1) = u(\zeta_1)$.

If now u is chosen, such that $\lim_{\xi \to 0} \frac{d}{dx} u(\xi) = \infty$, then the Kaczmarz-algorithm applied alternately at $\zeta_0 = 0$ and $\zeta_1 \in]0, 1]$ will converge to some approximation $q(.)$ (depending on ζ_1). It then holds, that $q(1) \to \infty$ for $\zeta_1 \to 0$. □

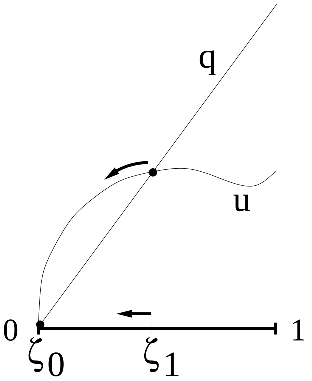

Figure 1: The Kaczmarz-algorithm has the property, that it may produce an unbounded sequence of approximations (function q in the figure), already if the function u to approximate is bounded. Kaczmarz-iteration applied alternately at two points ζ_0, ζ_1 will let the limit q connect the data-points $u(\zeta_0)$ and $u(\zeta_1)$.

Despite this property, the Kaczmarz-algorithm showed good numerical results. This may be due to the fact, that the pathological situation from the example will rarely occur. In practical situations, the sample points are distributed evenly in G. However, the described property may still give problems, if Kaczmarz-algorithm is applied close to singular points of the system.

Kronecker-Algorithm. The Kronecker-algorithm gives another solution to the problem of distributing a value inside a simplex to its vertices. It simply updates only in the closest vertex.

Again let $y \in G$ be a sample point, $a \in A_h$. Let $\Lambda(z)$ be the barycentric coordinates of $z = \varphi_{y,a}(h)$. Define $E(y) = (e_1(y), \ldots, e_N(y))^T$, where $e_j(y) = 1$ if

$$\min_k |x_k - y| = |x_j - y|,$$

and 0 otherwise. Let $r = r_h(y,a)$. We then define the operator

$$(P^{\mathbf{kr}}_{y,a}q)(a) = q(a) + E(y)[r + e^{-\rho h}\Lambda^T(z)\min(q) - \Lambda^T(y)q(a)], \tag{22}$$

and $(P^{\mathbf{kr}}_{y,a}q)(b) = q(b)$ if $a \neq b$.

The algorithm now takes the following form.

```
Q − KRONECKER(y, a, z, r)
        calculate $E(y)$
        calculate $\Lambda(z)$
        $v_{tmp} := \Lambda^T(z)\min_{b\in A_h} q^\nu(b)$
        update $q^{\nu+1}(a) := q^\nu(a) + (r + e^{-\rho h}v_{tmp} - q^\nu(a))E(y)$
                                                                        (23)
```

Concerning convergence, the Kronecker-algorithm is similar to the algorithm with information in vertices.

Lemma 4.3 *Let $Y = (y_1, \ldots, y_n)$, where $y_1, \ldots, y_n \in G$ is a set of sample points, such that for every vertex x_j, $j \in \{1, \ldots, N\}$ there is a unique y_k such that*

$$|x_j - y_k| = \min_i |x_i - y_k|.$$

For easier notation, the corresponding points and vertices shall have the same index, such that $|x_j - y_j| = \min_i |x_i - y_j|$. We will also assume, that for every y_j and every $a \in A_h$ we have costs $r_h(y_j, a)$. Let

$$P^{\mathbf{kr}}_j := \Pi_{a\in A_h} P^{\mathbf{kr}}_{y_j,a},$$

i.e. $P^{\mathbf{kr}}_j$ updates a Q-function in all controls $a \in A_h$ in the point y_j. Then the iteration operator

$$P^{\mathbf{kr}}_Y := P^{\mathbf{kr}}_1 \circ \ldots \circ P^{\mathbf{kr}}_N \tag{24}$$

is a contraction and for any $q^0 \in \mathcal{W}^k_h$ we have convergence of q^ν, $\nu \to \infty$, where

$$q^{\nu+1} = P^{\mathbf{kr}}_Y q^\nu. \tag{25}$$

Proof. It is easily seen, that by the assumption of unique correspondence of y_j to x_j, the operator $P^{\mathbf{kr}}_Y$ is a contraction with respect to the $||.||$-norm in $\mathcal{W}^k_h$. From the fixed-point theorem we therefore have convergence of $q^{\nu+1} = P^{\mathbf{kr}}_Y q^\nu$. □

Generalizations of Lemma 4.3 may be formulated, e.g. the following Corollary.

Corollary 4.4 *Lemma 4.3 holds also, if for every vertex x_j, $j = 1, \ldots, N$ there is a set of sample points $\{y_j^a\}_{a \in A_h}$ with*

$$|x_j - y_j^a| = \min_i |x_i - y_j^a|, \qquad \textit{for all } a \in A_h.$$

The operator $P_j^{\mathbf{kr}}$ will then have to be defined as

$$P_j^{\mathbf{kr}} := \Pi_{a \in A_h} P_{y_j^a}^{\mathbf{kr}}.$$

The fixed point $q_Y = P_Y^{\mathbf{kr}} q_Y$ clearly depends on the set of sample points $\{y_j\}_{j=1}^N$. The difference between q_Y and Q_h^k is, that they are fixed points for different sets of sample points. With the given notation we actually have

$$Q_h^k = q_X, \quad X = (x_1, \ldots, x_n).$$

We are interested in estimating the value $|| \min(q_Y) - \min(Q_h^k)||$ for some fixed point $q_Y = P_Y^{\mathbf{kr}} q_Y$ with a given set of sample points $\{y_j\}_{j=1}^N$ satisfying the conditions of Lemma 4.3.

Theorem 4 *Let $Y = (y_1, \ldots, y_N)^T$ be the tuple of points $\{y_i\}$ satisfying the conditions of Lemma 4.3. Let $V_Y = \min(q_Y)$, where $q_Y = P_Y^{\mathbf{kr}}(q)$ and let V_h^k be as defined before. Let V_h^k be Lipschitz-continuous with Lipschitz-constant $L_V > 0$. If $\rho > L_f$, then the following estimate holds*

$$\sup_{z \in G} |V_Y(z) - V_h^k(z)| \leq C(k + \frac{k}{h}) \tag{26}$$

for a constant C.

Proof. We introduce the following notation for costs and subsequent states. Define the column vector $R_Y(a)$ for every $a \in A_h$, $i = 1, \ldots, N$

$$[R_Y(a)]_i := \int_0^h e^{-\rho\tau} g(\varphi_{y_i,a}(\tau), a) d\tau,$$

and the $N \times N$-matrix $\Lambda_Y(a)$ holding the barycentric coordinates of the $\varphi_{y_i,a}$ such that

$$[\Lambda_Y(a) X]_i := \varphi_{y_i,a}(h) \tag{27}$$

and similarly define $R_X(a)$ and $\Lambda_X(a)$.

For V_Y and V_h^k we have

$$\begin{aligned} V_Y(x_i) &= \left[\min_{a\in A_h}\{R_Y(a) + e^{-\rho h}\Lambda_Y(a)V_Y\}\right]_i, \\ V_h^k(x_i) &= \left[\min_{a\in A_h}\{R_X(a) + e^{-\rho h}\Lambda_X(a)V_h^k\}\right]_i, \quad i = 1,\ldots,N. \end{aligned}$$

The points y_i have the property $|y_i - x_i| \le k$ for all $i = 1,\ldots,N$. The difference of local costs $R_Y(a)$ and $R_X(b)$ for minimizing controls $a, b \in A_h$ may be estimated by

$$\begin{aligned} |[R_X(a) - R_Y(b)]_i| &\le \int_0^h e^{-\rho\tau}|g(\varphi_{x_i,a}(\tau), a) - g(\varphi_{x_i,a}(\tau), b)|d\tau \\ &\le \int_0^h e^{(L_f-\rho)\tau} L_g|x_i - y_i|d\tau \\ &\le L_g kh. \end{aligned} \tag{28}$$

To estimate $|[\Lambda_Y(a)V_Y - \Lambda_X(b)V_h^k]_i|$, we take

$$|\varphi_{x_i,a}(h) - \varphi_{y_i,b}(h)| \le |x_i - y_i|e^{L_f h} \le ke^{L_f h} =: \varepsilon,$$

$i = 1,\ldots,N$, and the Lipschitz-continuity of V_h^k $(v, w \in G)$, to get

$$\begin{aligned} &|V_Y(\varphi_{y_i,a}(h)) - V_h^k(\varphi_{x_i,a}(h))| \\ &\le \max_{v\in G}\max_{|v-w|\le\varepsilon}|V_Y(v) - V_h^k(w)| \\ &\le \max_{v\in G}\max_{|v-w|\le\varepsilon}|V_Y(v) - V_h^k(v)| + |V_h^k(v) - V_h^k(w)| \\ &\le \max_{v\in G}|V_Y(v) - V_h^k(v)| + L_V\varepsilon. \end{aligned} \tag{29}$$

Together with (28) and (29) we get

$$\max_{v\in G}|V_Y(v) - V_h^k(v)| \le L_g kh + e^{-\rho h}(L_V k e^{(L_f-\rho)h} + \max_{w\in G}|V_Y(v) - V_h^k(v)|),$$

or

$$\max_{v\in G}|V_Y(v) - V_h^k(v)| \le \frac{k(L_g h + L_V e^{(L_f-\rho)h})}{1 - e^{-\rho h}}.$$

We can estimate $\frac{1}{1-e^{-\rho h}} \le \frac{1}{h\rho c}$ with $c = (1 - \frac{1}{e})$ for $h \in]0, \frac{1}{\rho}[$, and $e^{(L_f-\rho)h} \le 1$.

Together we have

$$||V_Y - V_h^k|| \le k\frac{L_g}{c\rho} + \frac{k}{h}\frac{L_V}{c\rho}.$$

This lets the constant C be equal to $\frac{1}{c\rho}\max\{L_g, L_V\}$.

□

If actual learning is performed in real time, then a fixed set of sample points Y will not be given, of course. The sample points may be located along the trial trajectory and will be distributed rather arbitrarily. In the investigation of properties of the Kronecker-algorithm we want to prove a result, which states that V_Y for a given set of points Y sufficing the assumptions of Lemma 4.3 lies inside a certain region which may be specified.

We first introduce some notation. Define

$$\mathrm{box}(x_i) := \{\xi \in G \,:\, |x_i - \xi| \le |x_j - \xi| \text{ for all } j = 1, \ldots, N\},$$

and for any point $\xi \in G$ we write

$$\mathrm{box}(\xi) := \bigcup_i \mathrm{box}(x_i), \qquad \text{for all the } i \text{ with } \xi \in \mathrm{box}(x_i).$$

For any $y \in G$ define the operator T_y as

$$T_y : \mathcal{V}^k \to \mathcal{V}^k,$$

$$(T_y v)(x_i) = \begin{cases} \min_{b \in A_h} \left\{ r_h(y, b) + e^{-\rho h} v(\varphi_{y,b}(h)) \right\}, & \text{if } x_i \in \mathrm{box}(y) \\ v(x_i), & \text{else .} \end{cases} \tag{30}$$

Theorem 5 *Let V_* and V^* be the fixed points in $\mathcal{V}^k$ of*

$$\begin{aligned} V_*(x_i) &= \inf_{y \in box(x_i)} T_y V_*(x_i) & (31) \\ V^*(x_i) &= \sup_{y \in box(x_i)} T_y V^*(x_i), \quad x_i \in G^k. & (32) \end{aligned}$$

Then for any $v \in \mathcal{V}^k$ with

$$V_* \le v \le V^* \tag{33}$$

we also have for all $y \in G$

$$V_* \le T_y v \le V^*. \tag{34}$$

For a set of points $Y = (y_1, \ldots, y_n)$ let $V_Y = \min(q_Y)$, where $q_Y = P_Y^{\mathbf{kr}}(q_Y)$ is the fixed point as before. Then a simple consequence of (34) is

$$V_* \le V_Y \le V^*. \tag{35}$$

In particular,

$$V_* \leq V_h^k \leq V^*.$$

Proof. It is straight forward to show that (31) and (32) are fixed point equations with respect to the maximum norm in $\mathcal{V}^k$, in particular $V_*, V^* \in \mathcal{V}^k$. Now let V_* be a function sufficing (31). Let $v \in \mathcal{V}^k$ with (33). With the monotonicity of T_y we get

$$V_*(x_i) = \inf_{y \in \mathrm{box}(x_i)} T_y V_*(x_i) \leq \inf_{y \in \mathrm{box}(x_i)} T_y v(x_i) \leq T_y v(x_i), \tag{36}$$

and analogously $T_x v(x) \leq V^*(x)$.

□

We have now established a region, in which a value function V_Y must lie for a set of test points Y as in Lemma 4.3. In the following Theorem we prove a result about the size of this region.

Theorem 6 *Suppose that V^* and V_* are Lipschitz-continuous with some constant L_V (since V^* and V_* are both bounded above by $\frac{M_g}{\rho}$ and by zero below, and $V^*, V_* \in \mathcal{V}^k$ we can say that $L_V \leq \frac{M_g}{k\rho}$). Then the following a priori estimate holds:*

$$||V^* - V_*|| \leq \frac{(L_r + L_V)k}{1 - e^{-\rho h}}. \tag{37}$$

Proof. Since $V^*, V_* \in \mathcal{V}^k$, if is sufficient to estimate the difference for all $x_i \in G^k$. We will write $\varphi_{y,a} = \varphi_{y,a}(h)$. Let $x_i \in G^k$. For some points $y, z \in box(x_i)$ and some control $b \in U$ we have

$$\begin{aligned}
V^*(x_i) - V_*(x_i) &= \max_{v \in box(x_i)} T_v V^*(x_i) - \min_{v \in box(x_i)} T_v V_*(x_i) \\
&= T_y V^*(x_i) - T_z V_*(x_i) \\
&= \min_{a \in U} \{r_h(y,a) + e^{-\rho h} V^*(\varphi_{y,a})\} - \min_{a \in U} \{r_h(z,a) + e^{-\rho h} V_*(\varphi_{z,a})\} \\
&\leq r_h(y,b) - r_h(z,b) + e^{-\rho h}(V^*(\varphi_{y,b}) - V_*(\varphi_{z,b})) \\
&\leq L_r|y - z| + e^{-\rho h}(V^*(\varphi_{y,b}) - V_*(\varphi_{y,b}) + V_*(\varphi_{y,b}) - V_*(\varphi_{z,b})) \\
&\leq L_r k + e^{-\rho h}(V^*(\varphi_{y,b}) - V_*(\varphi_{y,b})) + L_V |y - z| e^{L_f h} \\
&\leq L_r k + e^{-\rho h} \sup_{\zeta \in G}(V^*(\zeta) - V_*(\zeta)) + L_V k e^{L_f h}
\end{aligned}$$

We may now take $\sup_{x\in G}$ on both sides and get

$$||V^* - V_*|| \leq \frac{k(L_r + L_V e^{L_f h})}{1 - e^{-\rho h}}.$$

□

The Lipschitz-constant $L_V = \frac{M_g}{k\rho}$ grows with decreasing grid size k. We show, that under some condition, L_V can be chosen independently of k. This allows to conclude, that $||V^* - V_*|| \to 0$ for $k \to 0$.

Proposition 4.5 *Assume, that for any two vertices x_i, x_j we have an $\alpha \in]0,1]$ with*

$$\alpha k \leq |x_i - x_j|.$$

Also assume, that $\frac{2}{\alpha}e^{(L_f-\rho)h} < 1$. Then V^* and V_* are Lipschitz-continuous with constant

$$L_V = \frac{2}{\alpha} \cdot \frac{L_r}{1 - \frac{2}{\alpha}e^{(L_f-\rho)h}}.$$

Proof. Theorem 5 stated, that V^* and V_* are fixed points in $\mathcal{V}^k$. We will show, that if for some $V \in \mathcal{V}^k$ it holds that $|V|_{0,1} \leq L_V$, then also

$$\left| \inf_{y\in\text{box}(x_i)} T_y V(x_i) \right|_{0,1} \leq L_V$$

(and sup respectively). This shows, that because of uniqueness of V_* (and V^*) and closedness of $C^{0,1}(G)$ the proposition holds. Again, since $V_* \in \mathcal{V}^k$ we may consider two adjacent cells box(x_j) and box(x_i) with centers x_j, x_i. We get

$$\left| \inf_{z\in\text{box}(x_j)} T_z V_*(x_j) - \inf_{y\in\text{box}(x_i)} T_y V_*(x_i) \right| = |T_z V_*(x_j) - T_y V_*(x_i)|,$$

where on the right hand side we assume the existence of a $z \in \overline{\text{box}(x_j)}$ and $y \in \overline{\text{box}(x_i)}$, such that the infima are obtained in z, y, respectively. Furthermore, we may assume the existence of a $b \in A_h$, such that

$$\begin{aligned} |T_z V_*(x_j) - T_y V_*(x_i)| &= |r_h(z,b) + e^{-\rho h} V_*(\varphi_{z,b}) - r_h(y,b) - e^{-\rho h} V_*(\varphi_{y,b})| \\ &= |r_h(z,b) - r_h(y,b) + e^{-\rho h}(V_*(\varphi_{z,b}) - V_*(\varphi_{y,b}))| \\ &\leq |z-y|(L_r + e^{(L_f-\rho)h} L_V). \end{aligned}$$

We now estimate

$$|z-y| \le 2k = \frac{2}{\alpha}\alpha k \le \frac{2}{\alpha}|x_i - x_j|.$$

We may now substitute L_V and with the assumption $\frac{2}{\alpha}e^{(L_f-\rho)h} < 1$ we get

$$\begin{aligned} \left| \inf_{y\in \mathrm{box}(x_i)} T_y V(x_i) \right|_{0,1} &\le \frac{2L_r}{\alpha} + e^{(L_f-\rho)h}\frac{2}{\alpha} \cdot \frac{L_r}{1-\frac{2}{\alpha}e^{(L_f-\rho)h}} \\ &= \frac{2}{\alpha} \cdot \frac{L_r}{1-\frac{2}{\alpha}e^{(L_f-\rho)h}} = L_V. \end{aligned}$$

□

5 Numerical experiments

We performed learning experiments with a linear oscillator with controllable amplitude. The system equation has the following form

$$\dot{y} = f(y,u) := \begin{pmatrix} u & 1 \\ -1 & u \end{pmatrix} (y-v), \quad v = \begin{pmatrix} .375 \\ .375 \end{pmatrix}, \quad y \in G = [0,1]\times[0,1], \quad u \in [-c,c] \tag{38}$$

The stationary point of the uncontrolled system is v. The eigenvalues of the system are $\{u+i, u-i\}$.

At the boundary the system trajectory shall be projected onto the boundary. The right side of (38) therefore takes the following form at the boundary

$$\dot{y}_1 = \begin{cases} \min\{0; uy_1 + y_2\} & \text{falls} \quad y_1 = 1 \\ \max\{0; uy_1 + y_2\} & \text{falls} \quad y_1 = 0 \end{cases} \tag{39}$$

$$\dot{y}_2 = \begin{cases} \min\{0; uy_2 - y_1\} & \text{falls} \quad y_2 = 1 \\ \max\{0; uy_2 - y_1\} & \text{falls} \quad y_2 = 0 \end{cases}. \tag{40}$$

The goal of the optimal control shall be steer the solution along a given trajectory in state space (see figure 2). The reinforcement or cost function is therefore chosen to be

$$g(y) = \mathrm{dist}(L,y)^{\frac{1}{2}}, \tag{41}$$

where L denotes the set of points in the trajectory. We took the square root of the distance, to penalize stronger when close to the given trajectory. This speeded learning up. Note that the restriction of g to a grid is still L-continuous (as a function in $\mathcal{V}^k$). The cost functional takes the form

$$J_\rho(y, u(.)) = \int_0^\infty e^{-\rho\tau} g(\varphi_{y,u(.)}(\tau))d\tau. \tag{42}$$

For the simulation of the system (38) during the learning phase and also for calculation of the optimal value function V we discretized (38) with the implicit mid-point rule. Let $y_n \in G$ be the nth state of the discrete time system. Then we have

$$y_{n+1} = y_n + hf(\frac{1}{2}(y_n + y_{n+1}), u) \tag{43}$$

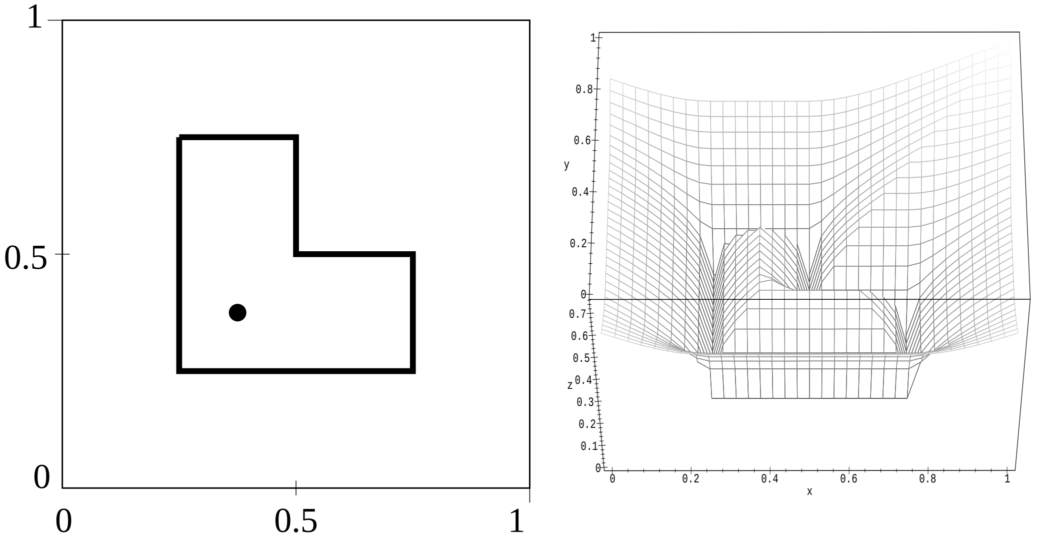

Figure 2: L-form of the given trajectory. The stationary point of the system is at $(.375, .375)$ (depicted as a big dot). The right picture shows the cost function g.

Because of the special form of $f(.,.)$ the implicit mid point rule may be formulated in an explicit form (see [15] for details).

The value of the local cost

$$r_h(y, a) = \int_0^h e^{-\rho\tau} g(\varphi_{y,a}(\tau))d\tau \tag{44}$$

was approximated by the trapezoidal rule

$$r_{appr}(y) = \frac{1}{2}(g(y) + e^{-\rho h} g(y)). \tag{45}$$

The state space was discretized in regular simplices with $k = 2^{-n}$ $n = \{3, 4, 5, 6, 7, 8\}$. The discount rate was chosen as $\rho = 5$. Is is clear that large discount rates give faster learning rates.

The following pictures show the mean distance of the system trajectory from the given L-form in one round, depending on the time steps that have passed. The dotted line shows this value when the optimal value function V_h^k was used for controlling the process. The other line shows this value during the learning process.

The first two of the following pictures show a comparison between Kaczmarz- and Kronecker-iteration. In most experiments (using different k and h) the Kaczmarz-Algorithm seemed to be more stable (the mean is closer to the dotted line). For this reason we subsequently used Kaczmarz-Iteration.

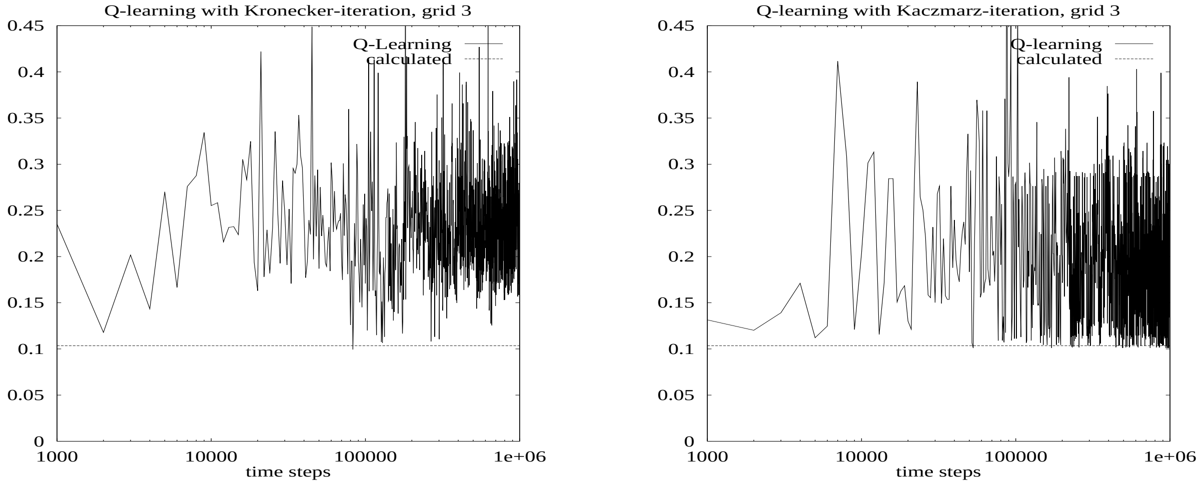

Figure 3: Difference between Kronecker- and Kaczmarz-iteration on a grid with $k = 1/2^n$ and $h = 0.2$.

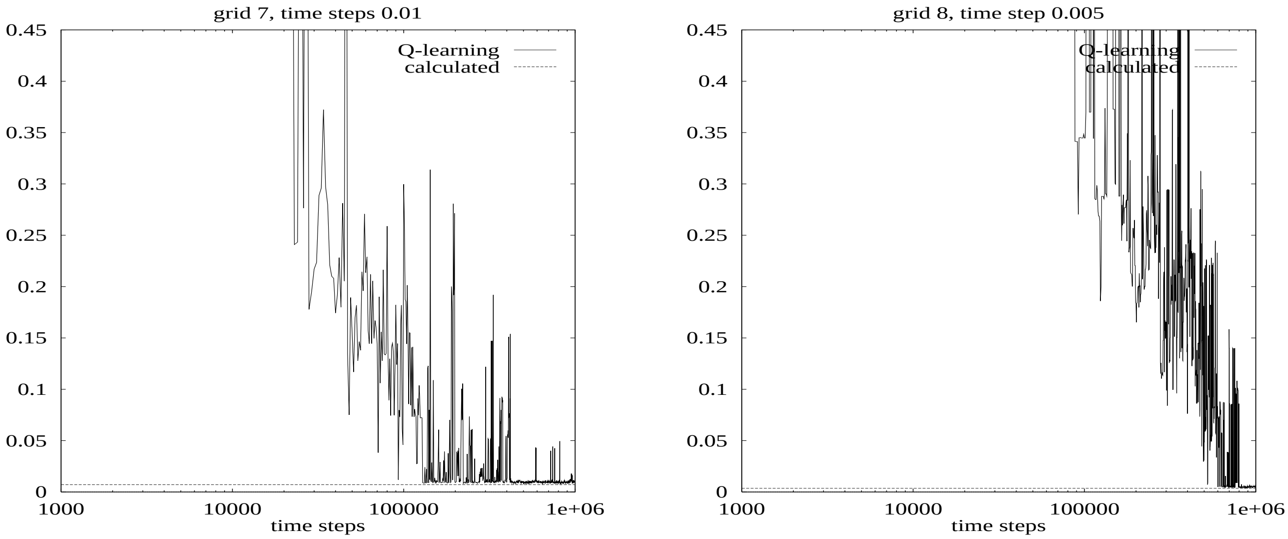

Figure 4: Learning with small time and space discretization approximates the optimal value well. The learning phase increases with decreasing discretization.

The last two pictures show the learning behavior for a very fine discretization. The relationship of time to space discretization was chosen as in Theorem 3. These pictures suggest, that learning could be accelerated, if the value function was approximated on a coarse grid first, and then the grid was refined. A preliminary result of this kind is described in [16].

6 Conclusions

We have investigated theoretically and numerically the behavior of two approximation schemes for Q-learning. The theoretical results may be used to define an error estimate (see [16], [17]) for local grid refinement. Local values of $V^* - V_*$ are used here.

Numerical experiments show, that with correct values for time and space discretization an accurate control may be learned. The specific choice of function approximation (Kaczmarz or Kronecker), however, makes a difference and is crucial.

Since a–priori information about the system and the cost function is not given to the controlling agent, he will have to experiment with different values of the discretization values k and h. Further research is directed towards an automatic detection of the optimal choice of the discretization parameters without a–priori knowledge of the system. Adaptive– and multi–grid algorithms for learning are also in current investigation.

References

[1] A. Barto, S. Bradtke, and S. Singh. Learning to act using real-time dynamic programming. *AI Journal on Computational Theories of Interaction and Agency 72:81-138*, 1995.

[2] A. G. Barto and R. H. Crites. Improving elevator performance using reinforcement learning. In *M.E.H.D.S.Touretzky, M. C. Mozer, editor, Advances in Neural Information Processing Systems 8, MIT Press*, 1996.

[3] S. Bradtke and M. Duff. Reinforcement Learning Methods for Continuous-Time Markov Decision Problems. In *NIPS-94*, 1994.

[4] F. Camilli and M. Falcone. An approximation scheme for the optimal control of diffusion processes. Technical report, Università degli Studi di Roma "'La Sapienza"', 1992.

[5] P. Dayan and T. Sejnowski. TD(λ) converges with probability 1. Technical report, Skripps Institute San Diego, 1992.

[6] T. Dietterich and W.Zhang. A reinforcement-learning approach to job-shop scheduling. In *Proceedings of the 14.th International Joint Coference on Artificial Intelligence*, 1995.

[7] I. C. Dolcetta. On a discrete approximation of the Hamilton-Jacobi equation of dynamic programming. *Appl Math Optim 10:367-377*, 1983.

[8] I. C. Dolcetta and H. Ishii. Approximate solutions of the Bellman equation of deterministic control theory. *Appl Math Optim 11:161-181*, 1984.

[9] K. Doya. Temporal difference learning in continuous time and space. In *NIPS 8*, 1996.

[10] M. Falcone. A numerical approach to the infinite horizon problem of deterministic control theory. *Appl Math Optim 15:1-13*, 1987.

[11] W. H. Fleming and H. M. Soner. *Controlled Markov Processes and Viscosity Solutions*. Springer-Verlag, 1993.

[12] R. Gonzalez and M. Tidball. On the rates of convergence of fully discrete solutions of Hamilton-Jacobi equations. *INRIA, Rapports de Recherche, No 1376, Programme 5*, 1991.

[13] L. Grüne. Numerische Optimale Steuerung und Stabilisierung. Master's thesis, Universität Augsburg, 1993.

[14] A. W. Moore and C. G. Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning, Volume 21*, 1995.

[15] S. Pareigis. *Lernen der Lösung der Bellman-Gleichung durch Beobachtung von kontinuierlichen Prozeßen*. PhD thesis, Universität Kiel, 1996.

[16] S. Pareigis. Adaptive choice of grid and time in reinforcement learning. In *Proceedings of the International Conference on Neural Information Processing Systems, to appear*, 1997.

[17] S. Pareigis and M. Riedmiller. A hybrid grid refinement scheme for reinforcement learning based on local defect correcting methods. Technical report, Lehrstuhl Praktische Mathematik, Universität Kiel, 1997.

[18] R. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning 3:9-44*, 1988.

[19] C. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning 8:279-292*, 1992.