Policy and Q-value Iteration

Gokul Swamy

(slides partially copied from Sanjiban / Wen; all mistakes my own)

Recap: Policy Gradients

$$\nabla_\theta J(\pi_\theta) = \sum_h^H \mathbb{E}_{s_h, a_h \sim \rho_h^{\pi_\theta}}[\nabla_\theta \log \pi_\theta(a_h \mid s_h) A^{\pi_\theta}(s_h, a_h)]$$

State-Action Distribution — *Likelihood Gradient* — *Advantage Function*

$$\theta_{t+1} \leftarrow \pi_{\theta_t} + \eta \nabla_\theta J(\pi_{\theta_t})$$

Policy Gradients = Advantage-Weighted MLE!

🔑 ***Question***: How big of a step (η) can we take?

Outline for Today

1. Recap: Policy Iteration and Q-value Iteration

2. Proving Monotonic Improvement of PI

3. What is *function approximation* in RL?

4. What breaks when we introduce function approximation into Policy and Q-value Iteration?

5. (Next time) The answer to the 🔑 Question!

Outline for Today

1. **Recap: Policy Iteration and Q-value Iteration**

2. Proving Monotonic Improvement of PI

3. What is *function approximation* in RL?

4. What breaks when we introduce function approximation into Policy and Q-value Iteration?

5. (Next time) The answer to the 🔑 Question!

Recap: Q-value Iteration

$\forall s \in \mathcal{S}, a \in \mathcal{A}, Q^\star_H(s, a) = r(s, a)$

for $h = H - 1, \ldots, 1$:

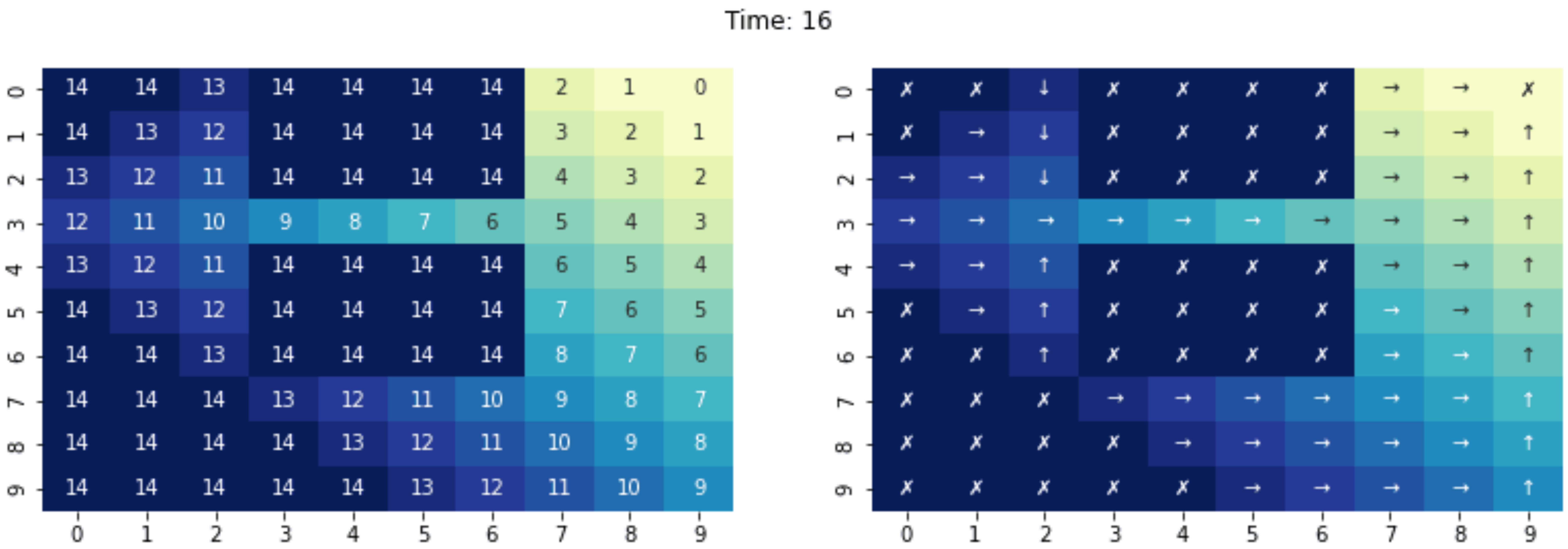
$$Q^\star_h(s, a) = r(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s' \mid s, a) \max_{a' \in \mathcal{A}} Q^\star_{h+1}(s', a')$$

Return $\pi^\star_h(s) = \arg\max_{a \in \mathcal{A}} Q^\star_h(s, a)$

Q: why does this give us $\pi^\star$?

A: proof by backwards induction!

Recap: Q-value Iteration



Recap: Policy Iteration

Initialize with arbitrary π_0, $t = 0$

while true:

$\forall s \in \mathcal{S}, a \in \mathcal{A},$ compute $Q^{\pi_t}(s, a)$ // *Policy Evaluation (how?)*

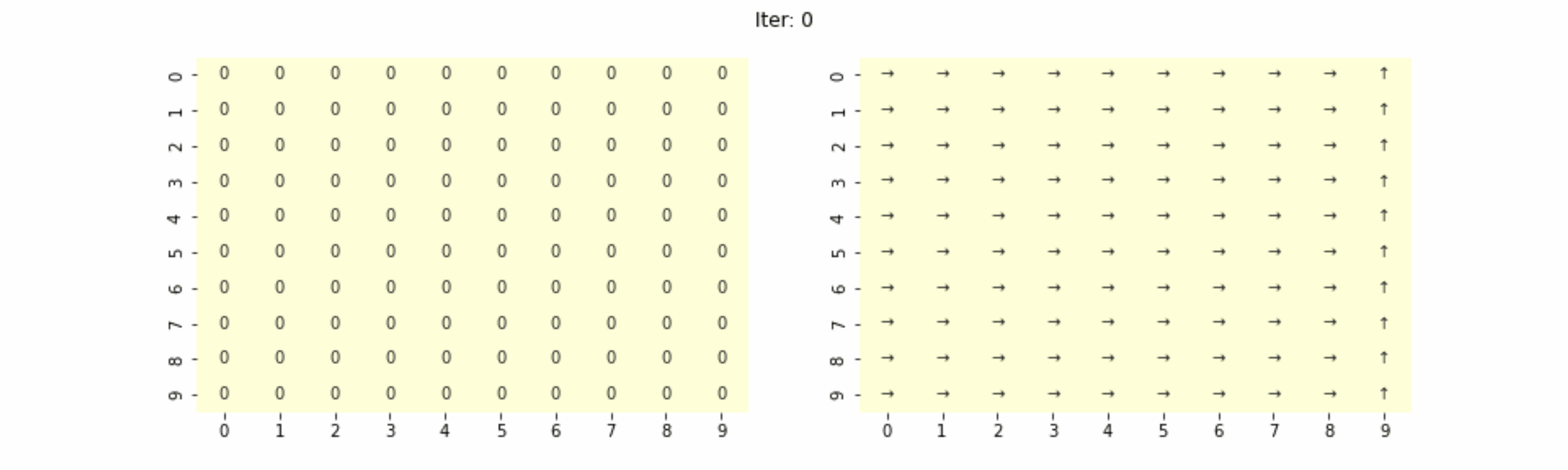
$\pi_{t+1}(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi_t}(s, a)$ // *Policy Improvement*

if $\pi_{t+1} = \pi_t$: **break;**

$t \leftarrow t + 1$

Return π_t

Recap: Policy Iteration



Recap: Policy Evaluation

Option 1: $\forall s \in \mathcal{S}, a \in \mathcal{A}$, roll out π_t repeatedly and compute the average over trajectories of cumulative reward.

Option 2:

$$\forall s \in \mathcal{S}, a \in \mathcal{A}, Q_H^{\pi_t}(s, a) = \mathbb{E}_{a \sim \pi_t(s)}[r(s, a)]$$

for $h = H - 1, \ldots, 1$:

$$Q_h^{\pi_t}(s, a) = \mathbb{E}_{a \sim \pi_t(s)}[r(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s' \mid s, a) Q_{h+1}^{\pi_t}(s', a')]$$

Option 2 *is caches Q-values to save samples!*

Recap: Policy Iteration

Initialize with arbitrary π_0, $t = 0$

while true:

$\forall s \in \mathcal{S}, a \in \mathcal{A},$ compute $Q^{\pi_t}(s, a)$ // *Policy Evaluation (how?)*

$\pi_{t+1}(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi_t}(s, a)$ // *Policy Improvement*

if $\pi_{t+1} = \pi_t$: **break;**

$t \leftarrow t + 1$

Return π_t

Q: why is $J(\pi_{t+1}) \geq J(\pi_t)$ *(monotonic improvement)*?

Outline for Today

1. Recap: Policy Iteration and Q-value Iteration

2. **Proving Monotonic Improvement of PI**

3. What is *function approximation* in RL?

4. What breaks when we introduce function approximation into Policy and Q-value Iteration?

5. (Next time) The answer to the 🔑 Question!

Monotonic Improvement of PI

Recall that defined $\pi_{t+1}(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi_t}(s, a)$

Thus, we know that $\forall s \in \mathcal{S}$,
$\mathbb{E}_{a \sim \pi_{t+1}(s)}[Q^{\pi_t}(s, a)] = \max_{a \in \mathcal{A}} Q^{\pi_t}(s, a) \geq \mathbb{E}_{a \sim \pi_t(s)}[Q^{\pi_t}(s, a)]$

Q: How do we translate these *local* improvements to a guarantee of *global* improvement?

The Performance Difference Lemma (PDL)

$$J(\pi_{t+1}) - J(\pi_t) = \mathbb{E}_{\xi \sim \pi_{t+1}}\left[\sum_h^H \underbrace{\mathbb{E}_{a' \sim \pi_{t+1}(s_h)}[Q^{\pi_t}(s_h, a')] - \mathbb{E}_{a' \sim \pi_t(s_h)}[Q^{\pi_t}(s_h, a')]}_{\geq 0, \forall s \in \mathcal{S}}\right]$$

$$\geq 0$$

[Kakade and Langford, '02]

Proving the PDL

$$J(\pi_{t+1}) - J(\pi_t) = \mathbb{E}_{\xi \sim \pi_{t+1}}\left[\sum_h^H r(s_h, a_h)\right] - \mathbb{E}_{s_0}[V^{\pi_t}(s_0)]$$

$$= \mathbb{E}_{\xi \sim \pi_{t+1}}\left[\sum_h^H r(s_h, a_h) - V^{\pi_t}(s_0)\right]$$

(same start state dist.)

$$= \mathbb{E}_{\xi \sim \pi_{t+1}}\left[\sum_h^H r(s_h, a_h) + V^{\pi_t}(s_{h+1}) - V^{\pi_t}(s_h)\right]$$

(telescopes)

Proving the PDL

$$J(\pi_{t+1}) - J(\pi_t) = \mathbb{E}_{\xi\sim\pi_{t+1}}\left[\sum_h^H r(s_h, a_h) + V^{\pi_t}(s_{h+1}) - V^{\pi_t}(s_h)\right]$$

$$= \mathbb{E}_{\xi\sim\pi_{t+1}}\left[\sum_h^H r(s_h, a_h) + V^{\pi_t}(s_{h+1}) - \mathbb{E}_{a'\sim\pi_t(s_h)}[Q^{\pi_t}(s_h, a')]\right]$$

$$= \mathbb{E}_{\xi\sim\pi_{t+1}}\left[\sum_h^H \mathbb{E}_{a'\sim\pi_{t+1}(s_h)}[Q^{\pi_t}(s_h, a')] - \mathbb{E}_{a'\sim\pi_t(s_h)}[Q^{\pi_t}(s_h, a')]\right]$$

Recap: Policy Iteration

Initialize with arbitrary π_0, $t = 0$

while true:

$\forall s \in \mathcal{S}, a \in \mathcal{A},$ compute $Q^{\pi_t}(s, a)$ // *Policy Evaluation (how?)*

$\pi_{t+1}(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi_t}(s, a)$ // *Policy Improvement*

if $\pi_{t+1} = \pi_t$: **break;**

$t \leftarrow t + 1$

Return π_t

Q: why is $J(\pi_{t+1}) \geq J(\pi_t)$ *(monotonic improvement)*?

A: local improvements + PDL

Outline for Today

1. Recap: Policy Iteration and Q-value Iteration

2. Proving Monotonic Improvement of PI

3. **What is *function approximation* in RL?**

4. What breaks when we introduce function approximation into Policy and Q-value Iteration?

5. (Next time) The answer to the 🔑 Question!

Function Approximation in RL

- For problems with small state / action spaces (i.e. *tabular* problems), we can represent policies, value functions, Q functions as *lookup tables* in memory.
- However, for problems with large or continuous state spaces, we instead need to fit functions to approximate these functions in memory.
- e.g. policies: NN that maps from state to mean / variance of a Gaussian, often called an *actor*
- e.g. Q functions: NN that maps from state / action pairs to a real number, often called a *critic*

Fitted Q-Iteration

$\forall s \in \mathcal{S}, a \in \mathcal{A}, Q^\star_H(s,a) = r(s,a)$

for $h = H - 1, \ldots, 1$:

$$Q^\star_h(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} T(s' \mid s,a) \max_{a' \in \mathcal{A}} Q^\star_{h+1}(s',a')$$

Return $\pi^\star_h(s) = \arg\max_{a \in \mathcal{A}} Q^\star_h(s,a)$

Receive some dataset $\mathcal{D} = \{(s,a,r,s')\}$

Initialize $\hat{Q}_0 \in \mathcal{F}_Q, t \leftarrow 0$

for $t \in 1, \ldots, T$

$$\hat{Q}_{t+1} \leftarrow \arg\min_{Q \in \mathcal{F}_Q} \mathcal{E}_{\mathcal{D}}[(Q(s,a) - (r + \max_{a' \in \mathcal{A}} \hat{Q}_t(s',a')))^2]$$

Return π_T

Fitted Q-Iteration

source of pain 1 *source of pain 2*

$$\hat{Q}_{t+1} \leftarrow \arg\min_{Q \in \mathcal{F}_Q} \mathcal{E}_{\mathcal{D}}[(Q(s,a) - (r + \max_{a' \in \mathcal{A}} \hat{Q}_t(s', a')))^2]$$

"target network"

Approximate Policy Iteration

Initialize with arbitrary π_0, $t = 0$

while true:

$\forall s \in \mathcal{S}, a \in \mathcal{A},$ compute $Q^{\pi_t}(s, a)$

$\pi_{t+1}(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi_t}(s, a)$

if $\pi_{t+1} = \pi_t$**: break;**

$t \leftarrow t + 1$

Return π_t

Initialize with arbitrary π_0, $t = 0$

for $t \in 1, \ldots, T$

Sample

$\mathcal{D}_t = \{(s_h, a_h, \hat{Q} = \sum_{\tau=h}^{H} r(s_\tau, a_\tau)) \sim \pi_t\}$

Fit $\hat{Q}_t \leftarrow \arg\min_{Q \in \mathcal{F}_Q} \mathbb{E}_{\mathcal{D}_t}[(Q(s, a) - \hat{Q})^2]$

$\pi_{t+1}(s) = \arg\max_{a \in \mathcal{A}} \hat{Q}_t(s, a)$

if $\pi_{t+1} = \pi_t$**: break;**

Return π_T

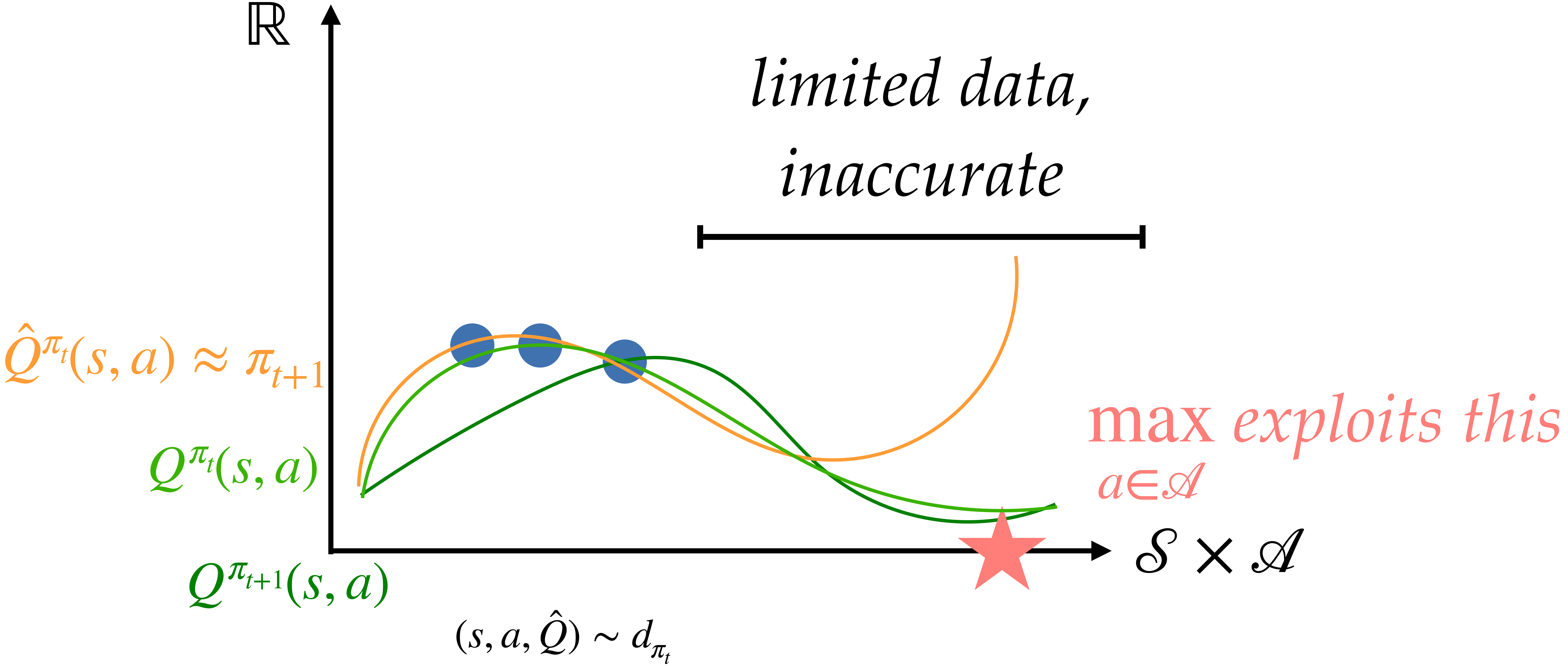
same sources of pain!

Outline for Today

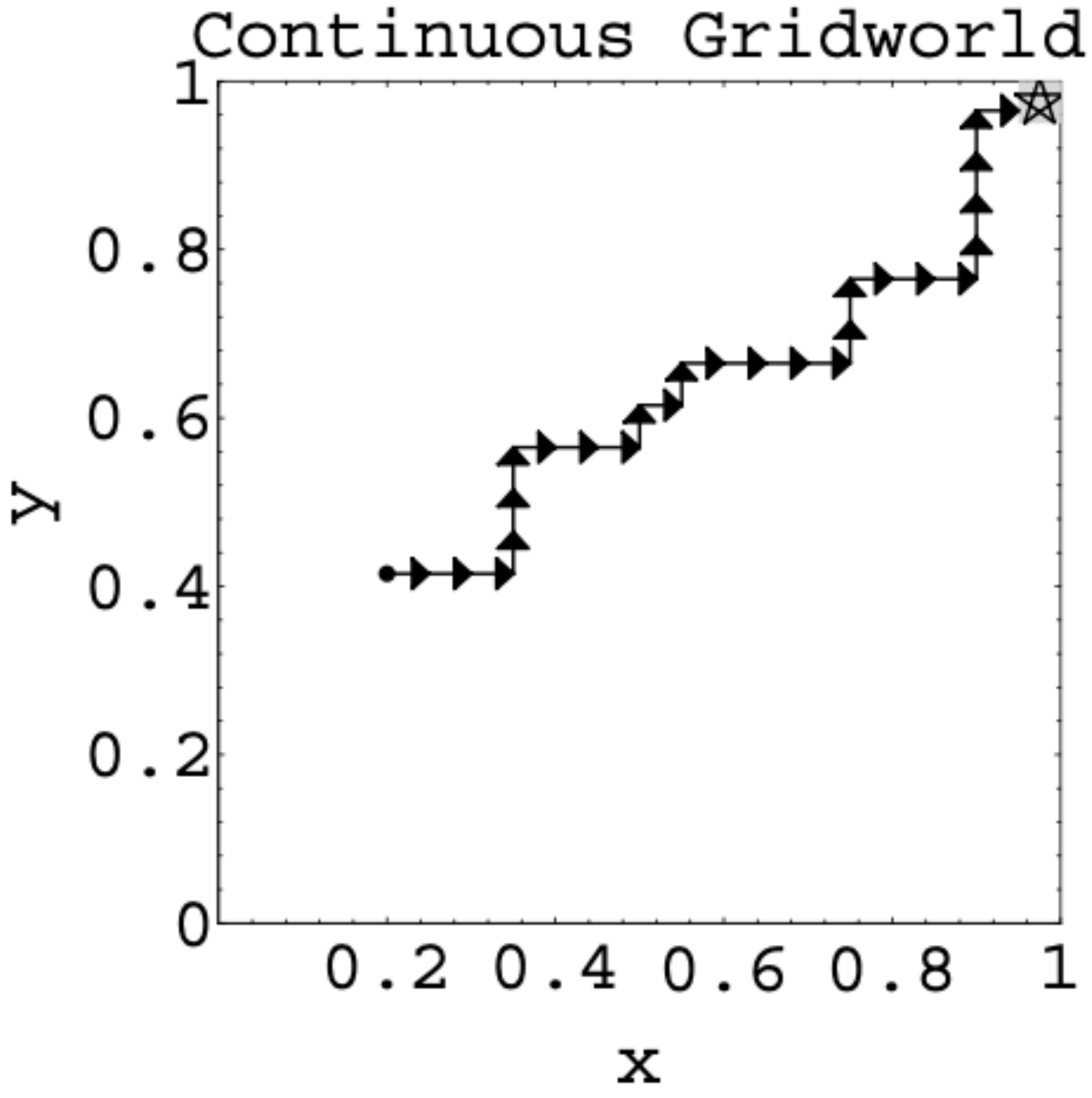
1. Recap: Policy Iteration and Q-value Iteration
2. Proving Monotonic Improvement of PI
3. What is *function approximation* in RL?
4. **What breaks when we introduce function approximation into Policy and Q-value Iteration?**
5. (Next time) The answer to the 🔑 Question!

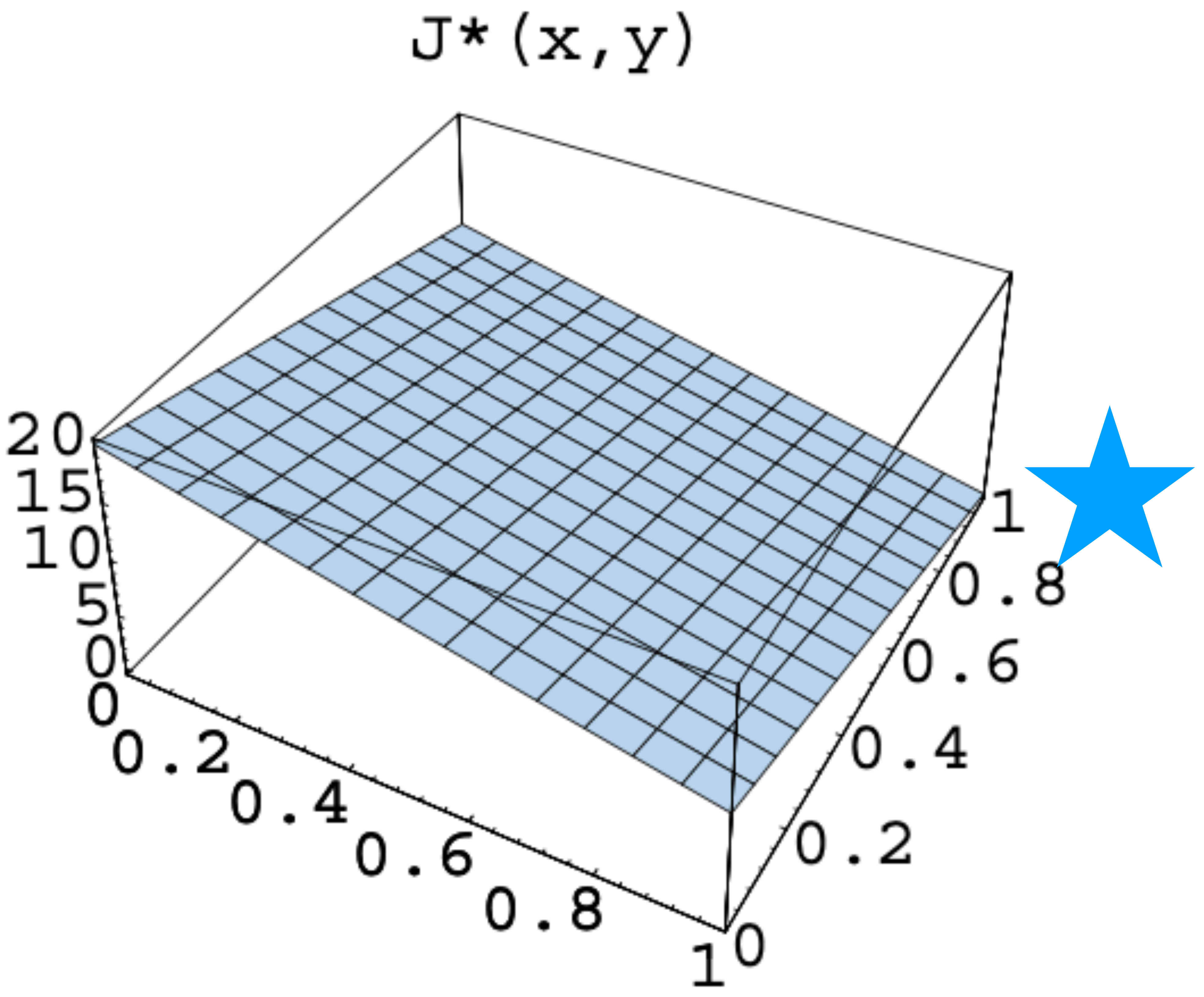
The Perils of Function Approximation in RL:

No More Monotonic Improvement! 😱😱😱

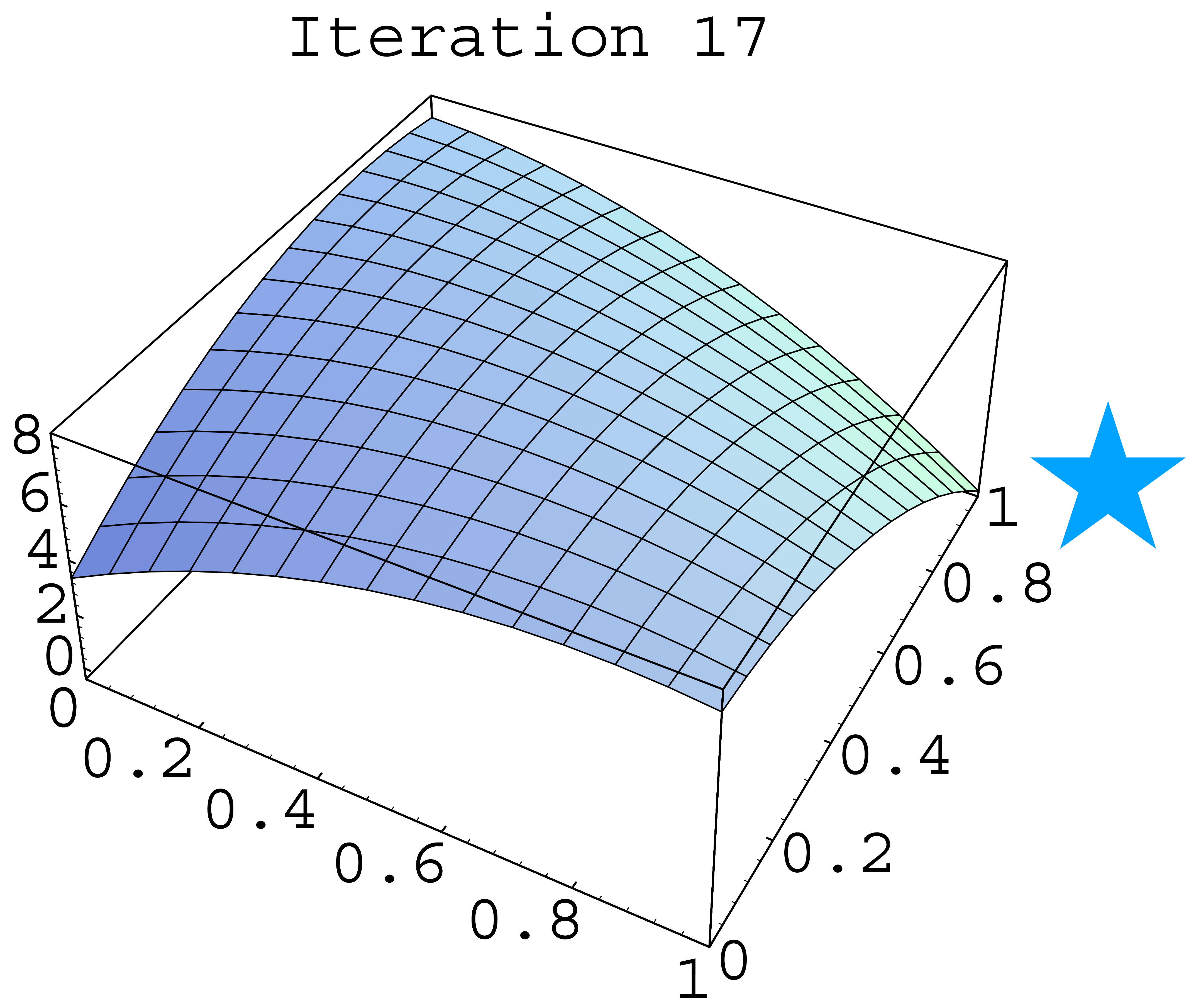


Example 1: Gridworld

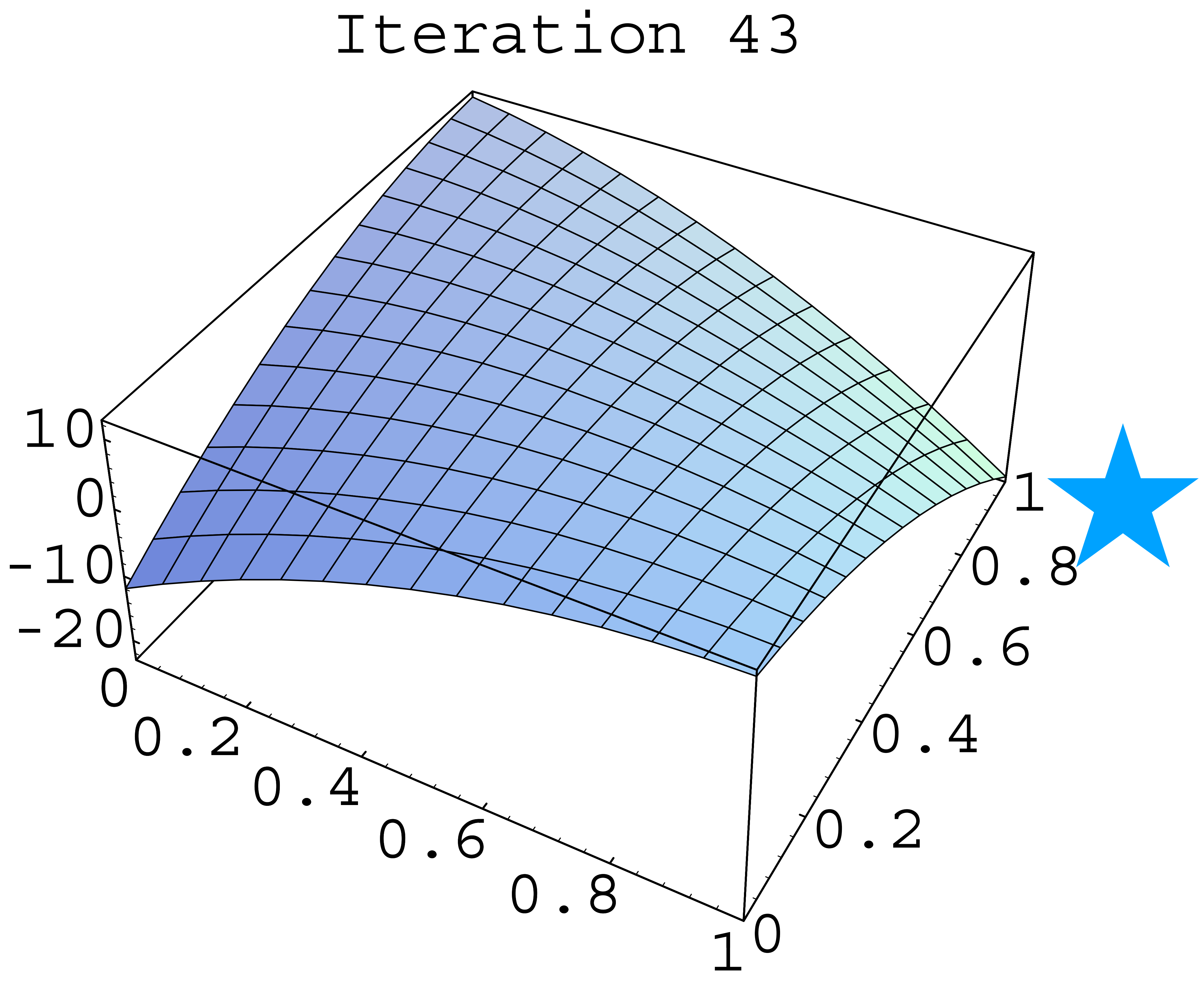




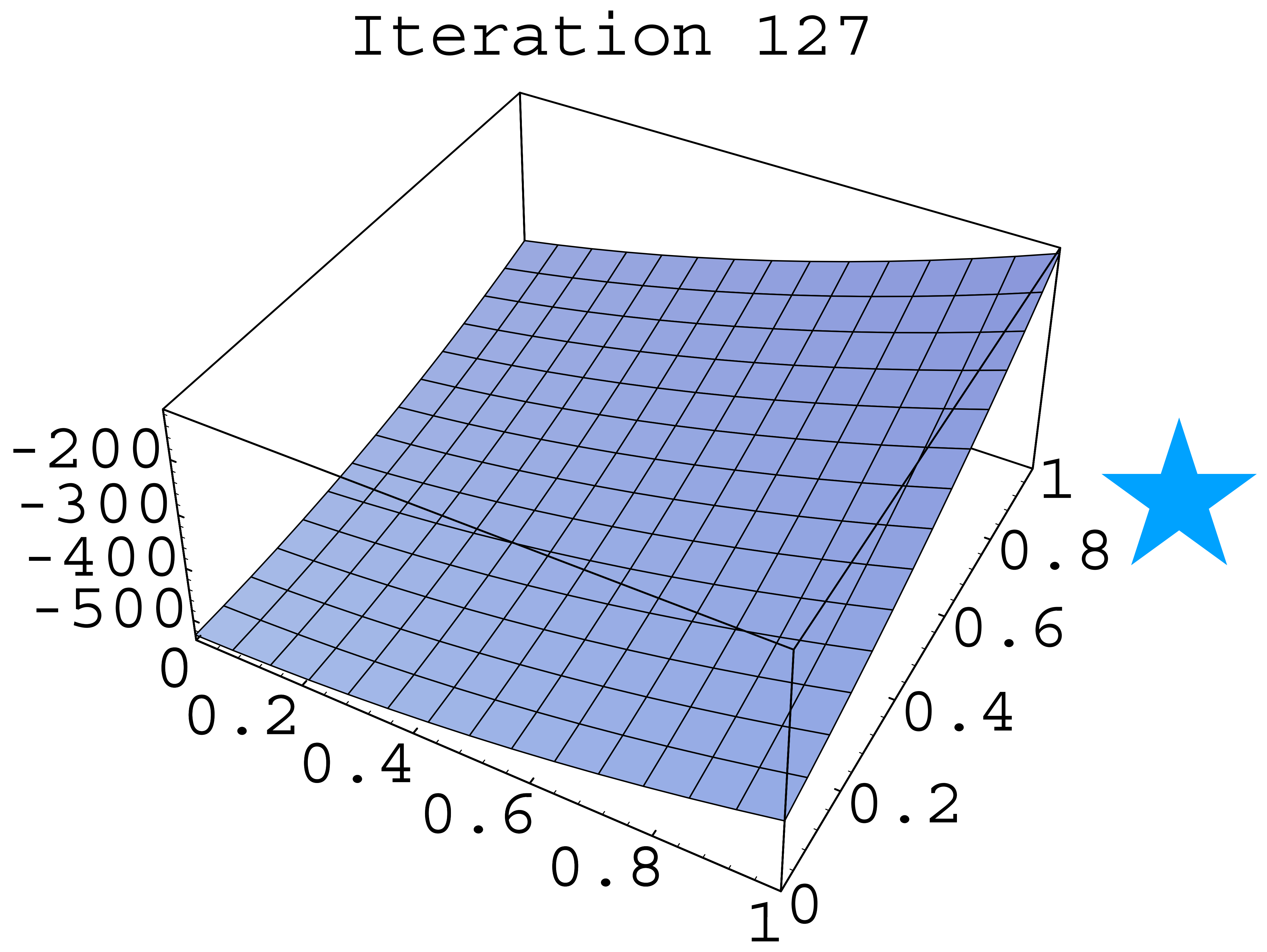
Value Iteration with a Quadratic



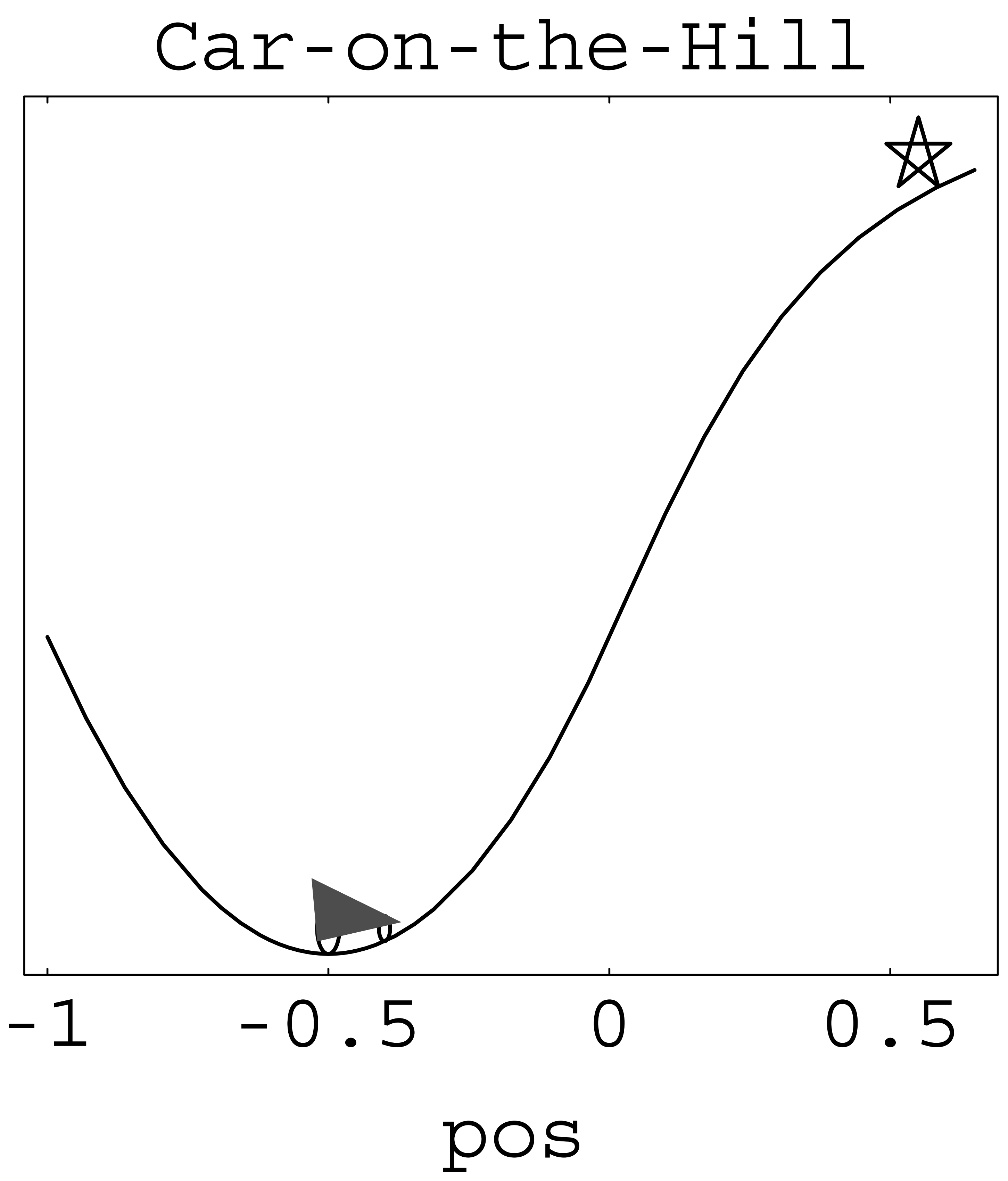
Value Iteration with a Quadratic

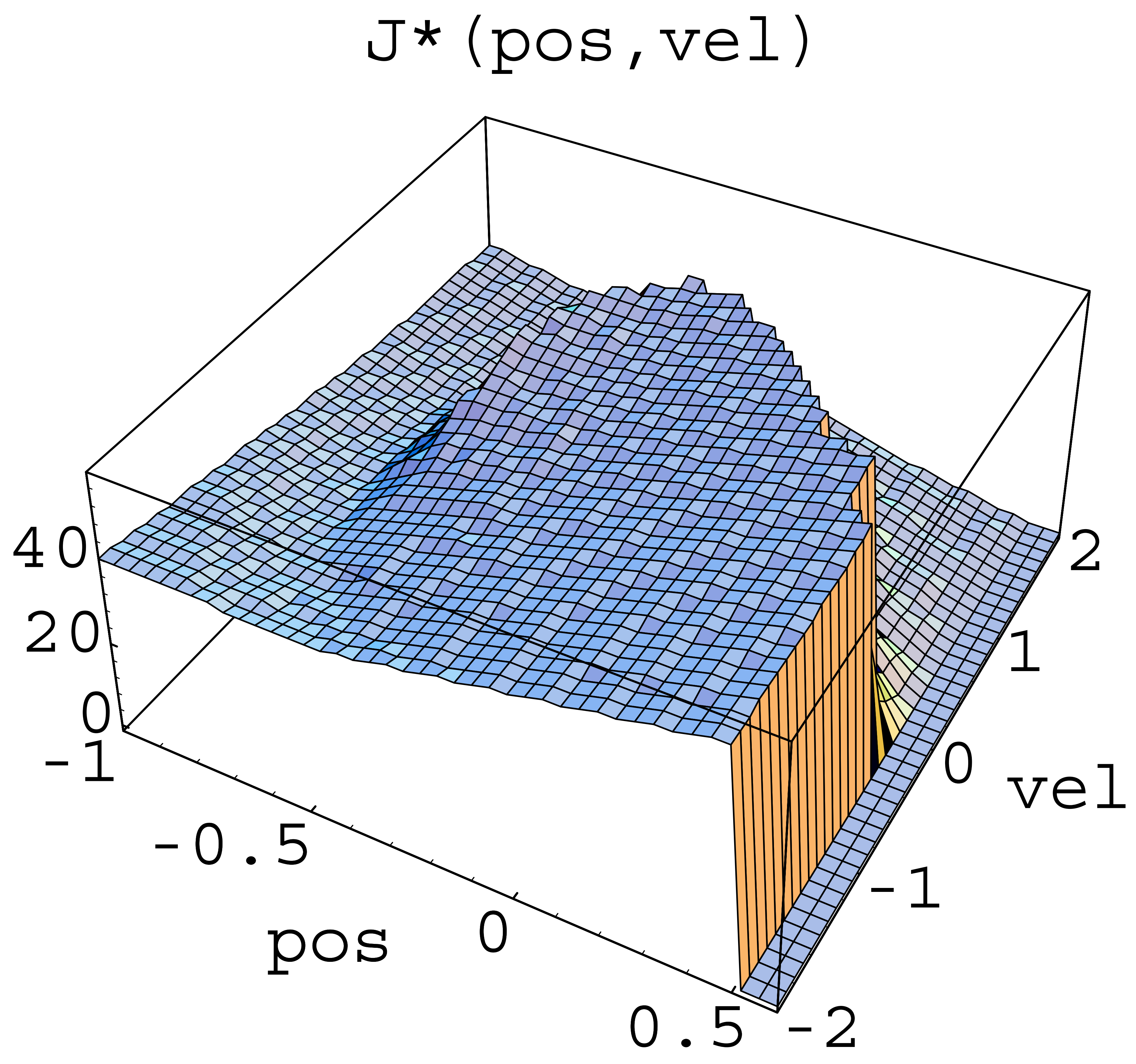


Value Iteration with a Quadratic

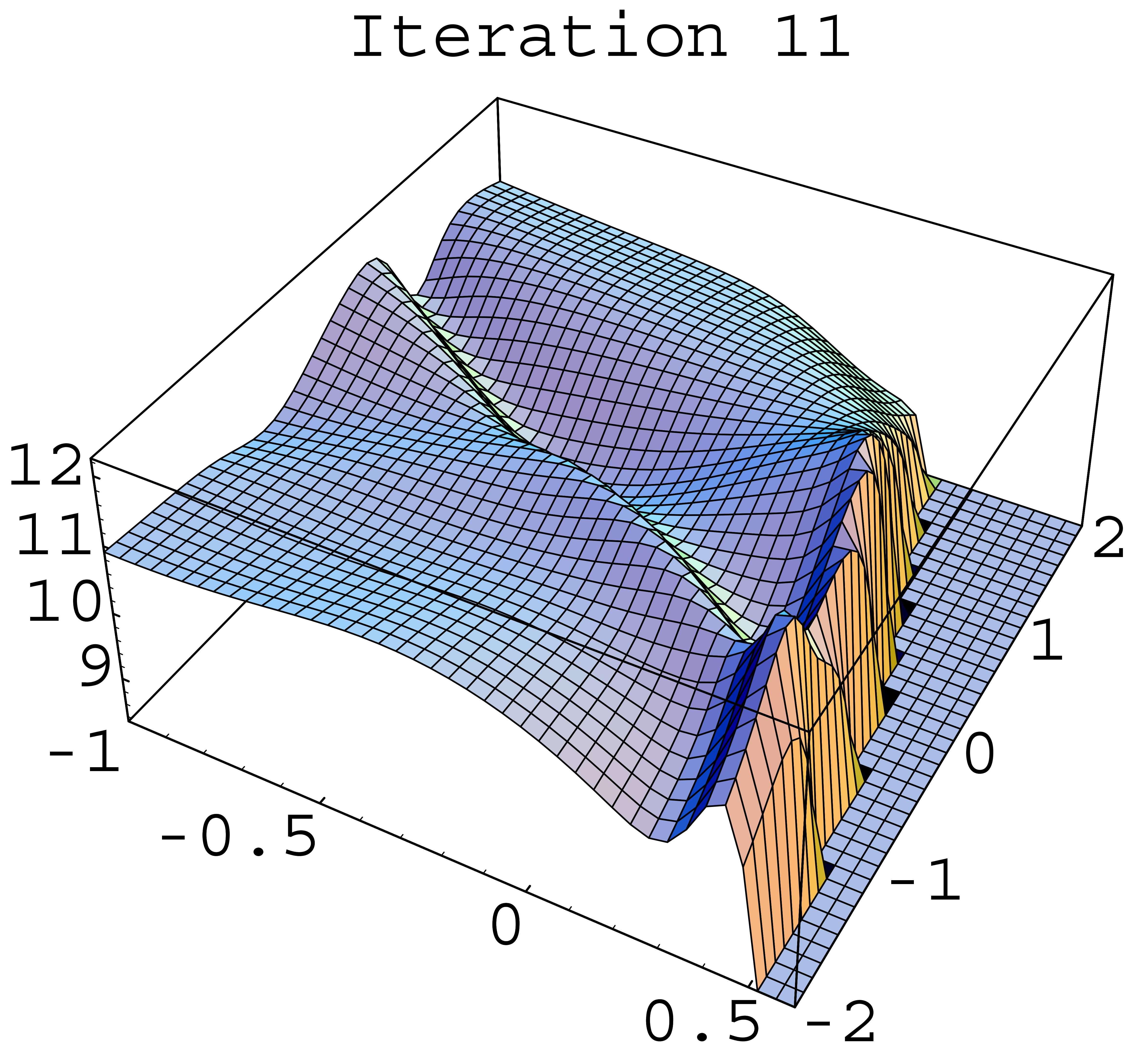


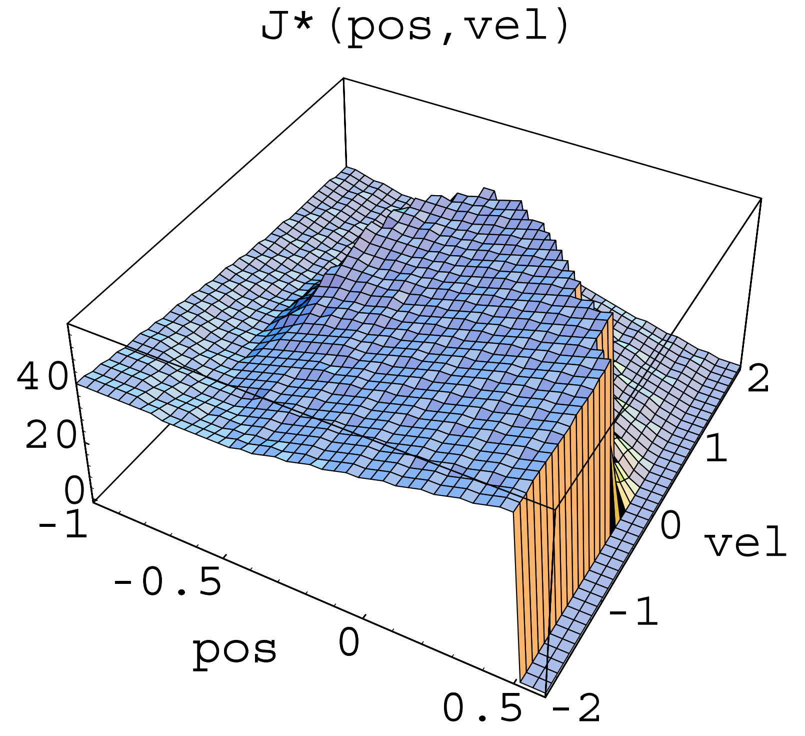
Example 2: Mountain Car



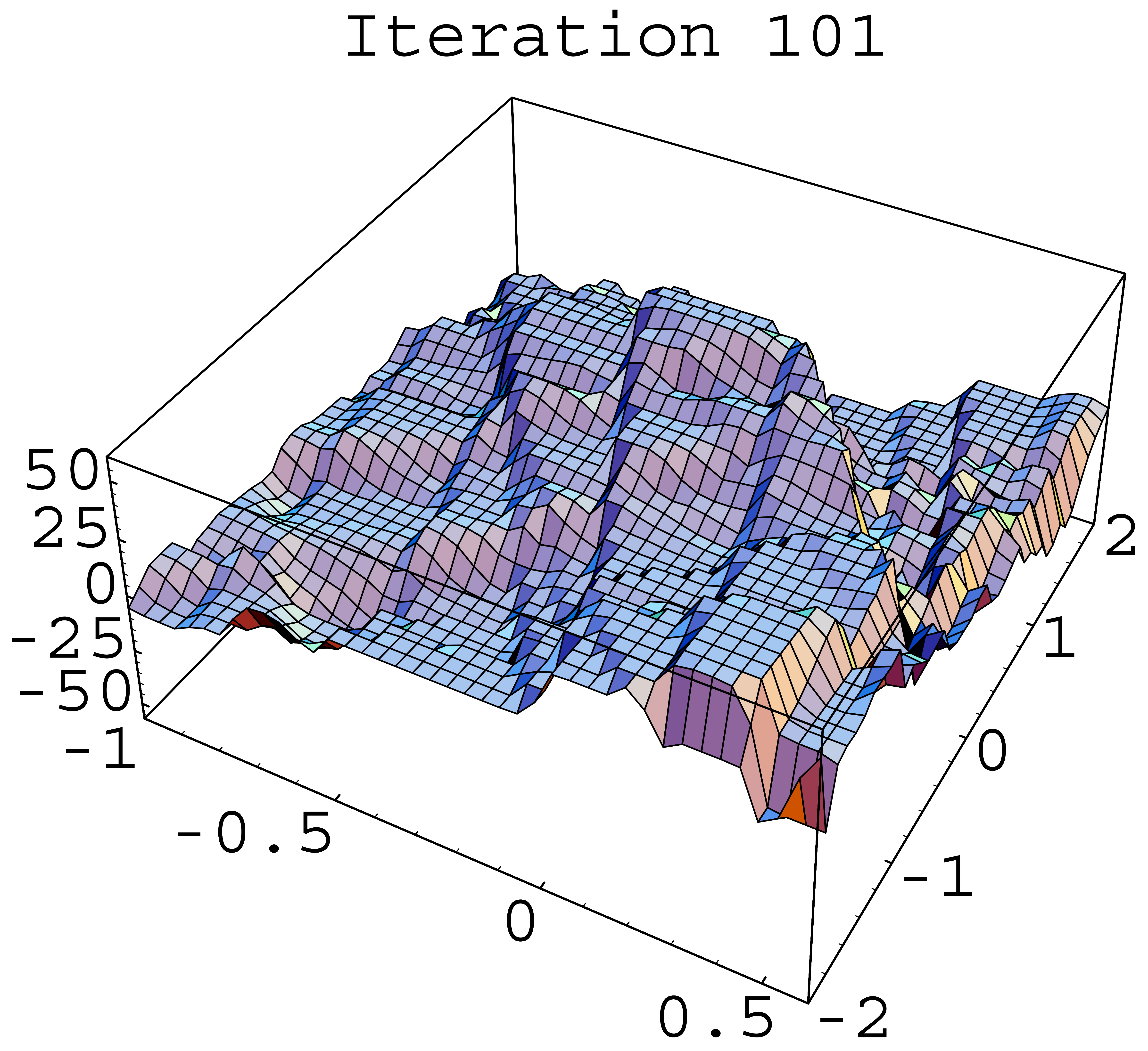


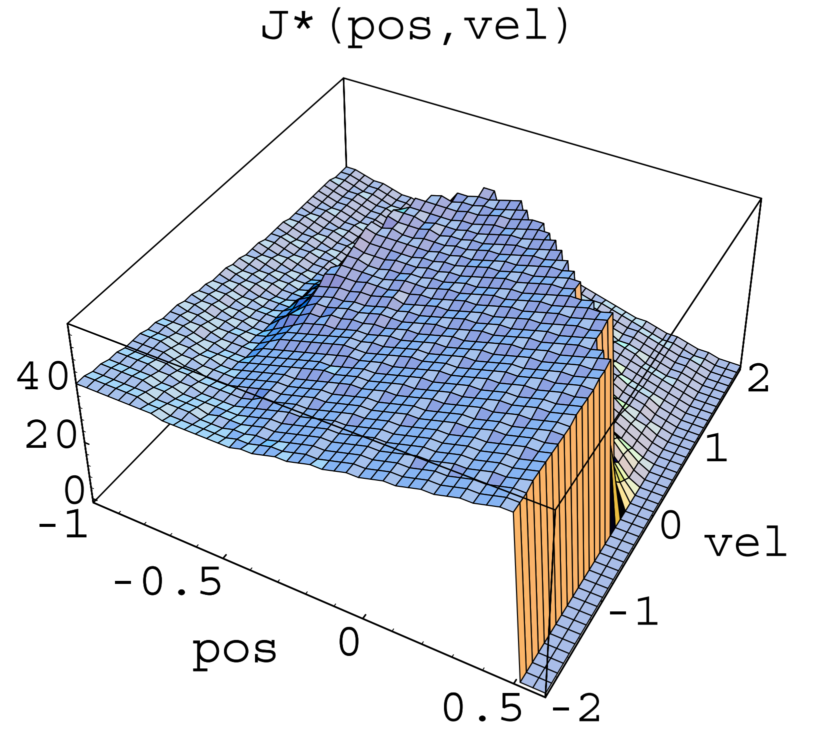
Value Iteration with a 2 Layer NN



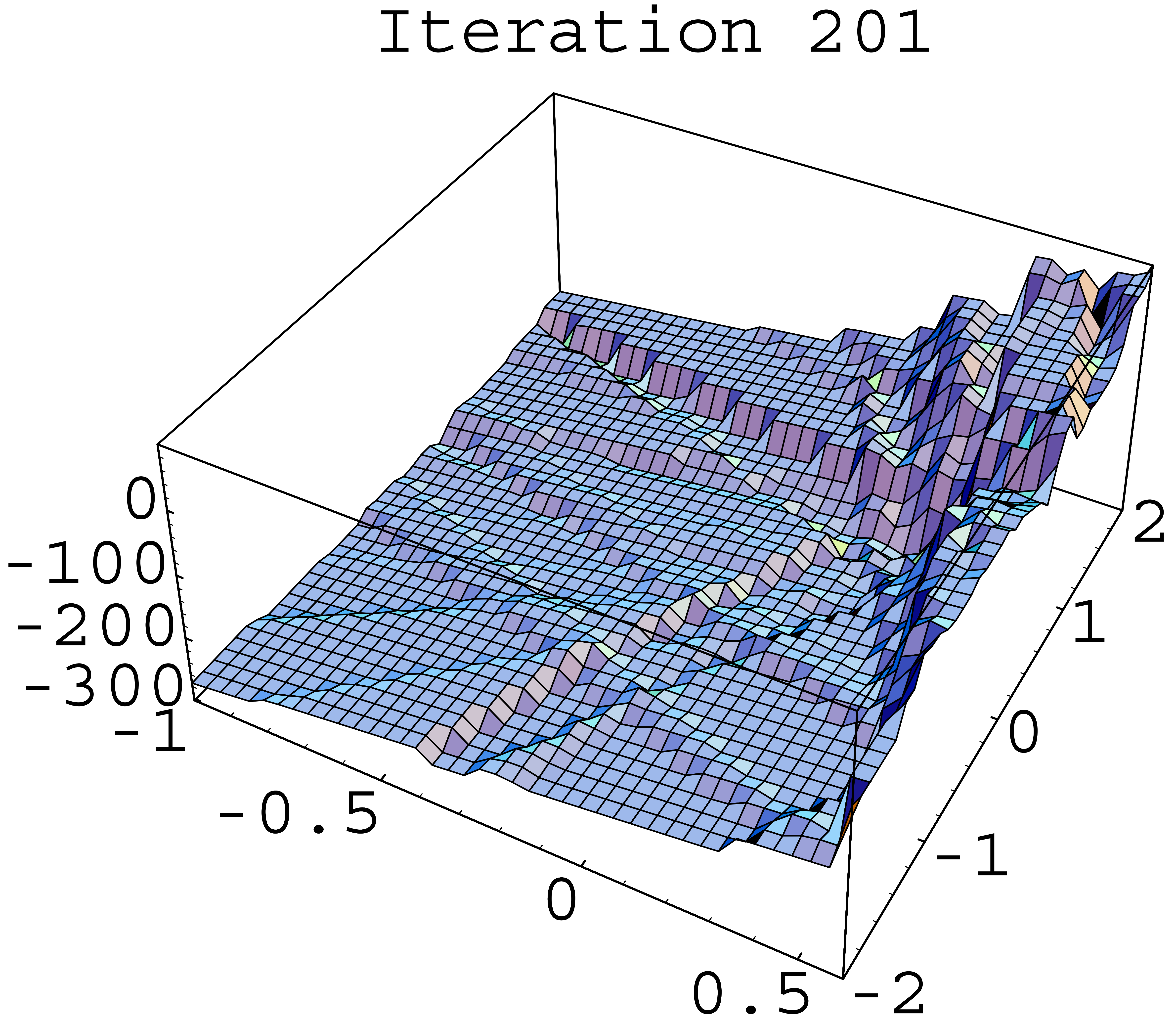


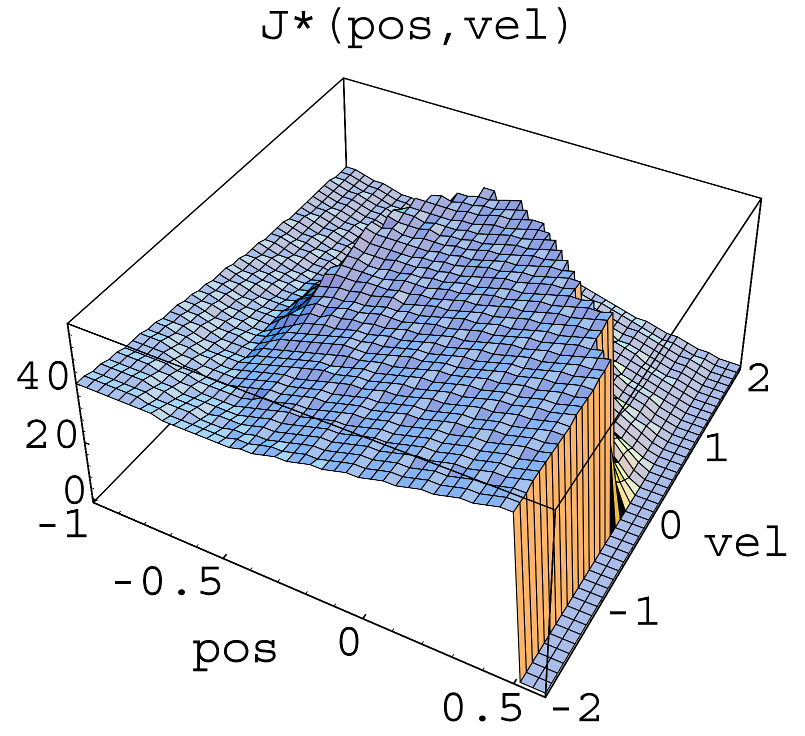
Value Iteration with a 2 Layer NN





Value Iteration with a 2 Layer NN





Summary

1. Policy Iteration and Q-value Iteration have monotonic improvement guarantees in the tabular setting, which we can prove via the PDL.

2. Scaling either method to larger problems requires function approximation for both policies and Q-functions.

3. However, these functions can be overly optimistic outside of their training distribution, leading to poor performance at test time (i.e. *covariate shift*).

4. Stay tuned: ***how do we fix this????***