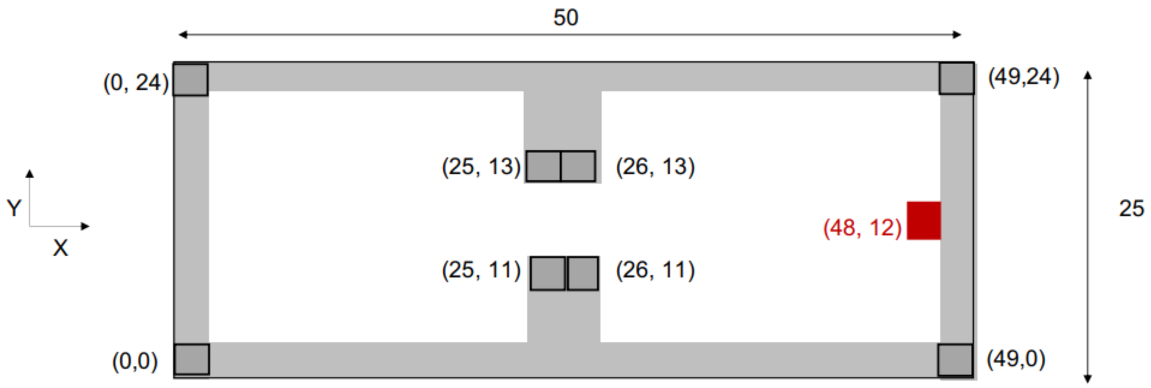
COL 864: **Planning and Estimation for Autonomous Systems** **II Semester 2020-2021**

Assignment 2 Report

Student: Isha Chaudhary

1 MDP



1.1 Part a: Value Iteration

1.1.1 Background

Value Iteration is an iterative approach to solve the Bellman Equation for the State-Value function. It is primarily based on the Gauss-Seidel Method of solving simultaneous, coupled equations.

The State-Value function for a given policy, with time horizon, T, is given by:

$$V_T^\pi(s_t) = E_{s_{\tau:T}}[\sum_{\tau=t}^{T} \gamma^t R(s_\tau, \pi(s_\tau), s_{\tau+1})]$$

Here, γ is the discount factor, and R(s, a, s') is the reward function for the transition to s' from s, by taking action, a. Thus, the Bellman Equation for the State-Value Function, given a policy, is the following:

$$V(s) = E[R_{t+1} + \gamma V(s_{t+1})|s_t = s]$$

or more elaborately:

$$V_T^\pi(s_t) = \sum_{s_{t+1}} p(s_{t+1}|s_t, \pi(s_t))(R(s_t, \pi(s_t), s_{t+1}) + \gamma V_{T-1}^\pi(s_{t+1}))$$

The Value Iteration Algorithm finds the optimal policy by iteratively finding the State-Value function for each of the states and constituting a deterministic policy of actions which maximize the state-value function of a given state, computed from the State-value function of the state reached after action a.

The iterative procedure updates the state-value function as:

$$V(s) = max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$$

The state value function for each of the states is updated till the maximum difference between consecutive state-value functions falls below a pre-decided threshold.

After the state-value function converges, the deterministic policy returned, for a given state, s is:

$$\pi(s) = argmax_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$$

1.1.2 Model

Action Model: Let the intended action be a

$$p(a) = 0.8$$
$$p(A - a) = 0.2/3, \text{ where A is the action space.}$$

Reward Model:

$$R(\text{Collide with a wall}) = -1$$
$$R(\text{reach goal state}) = +100$$
$$R(\text{otherwise}) = 0$$

1.1.3 Implementation Parameters

- The state-values of the grid cells was randomly assigned some value between 0 and 1.
- The state-values of the wall cells was initialized as 0.
- Discount Factor, $\gamma = 0.1$
- Threshold, $\theta = 0.1$
- Maximum number of iterations = 100

1.1.4 Observations

The State-Value function for each of the feasible grid cells is shown in 1.

Here, the blue region indicates the walls in the grid world. All other grid cells can be visited by the agent receiving 0 reward. The grid cell, highlighted with red boundary is the goal state.
The State-Value shown is after 100 iterations of Value Iteration.

The Policy obtained at the end of Value Iteration procedure is shown in 2

1.1.5 Inference

1. The plot for the State-Value function is showing high value for the goal state and other states around it.
2. The high value of the goal state can be attributed to the fact that the goal state is a non-terminal state and a very high reward, +100, is received on staying in the goal state.
3. The corresponding policy map obtained indicates the presence of credit-assignment problem in the given instance. As the reward obtained, for a transition neither into a wall nor into the goal state, is 0, and the discount factor is very low, the value function of the states away from the goal couldn't be updated very much.
4. Low Discount factor value causes less influence of the reward at the goal state, on the states far away from the goal.

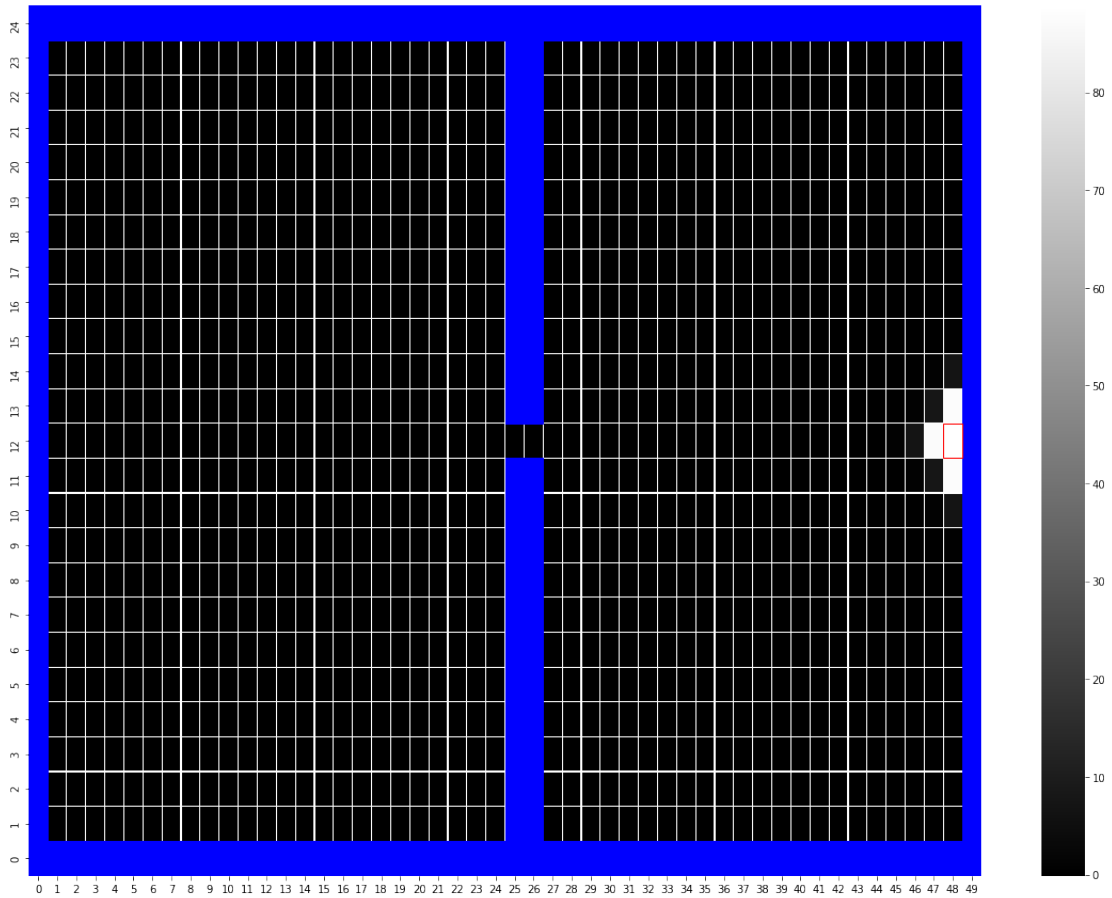


Figure 1: State Value Function

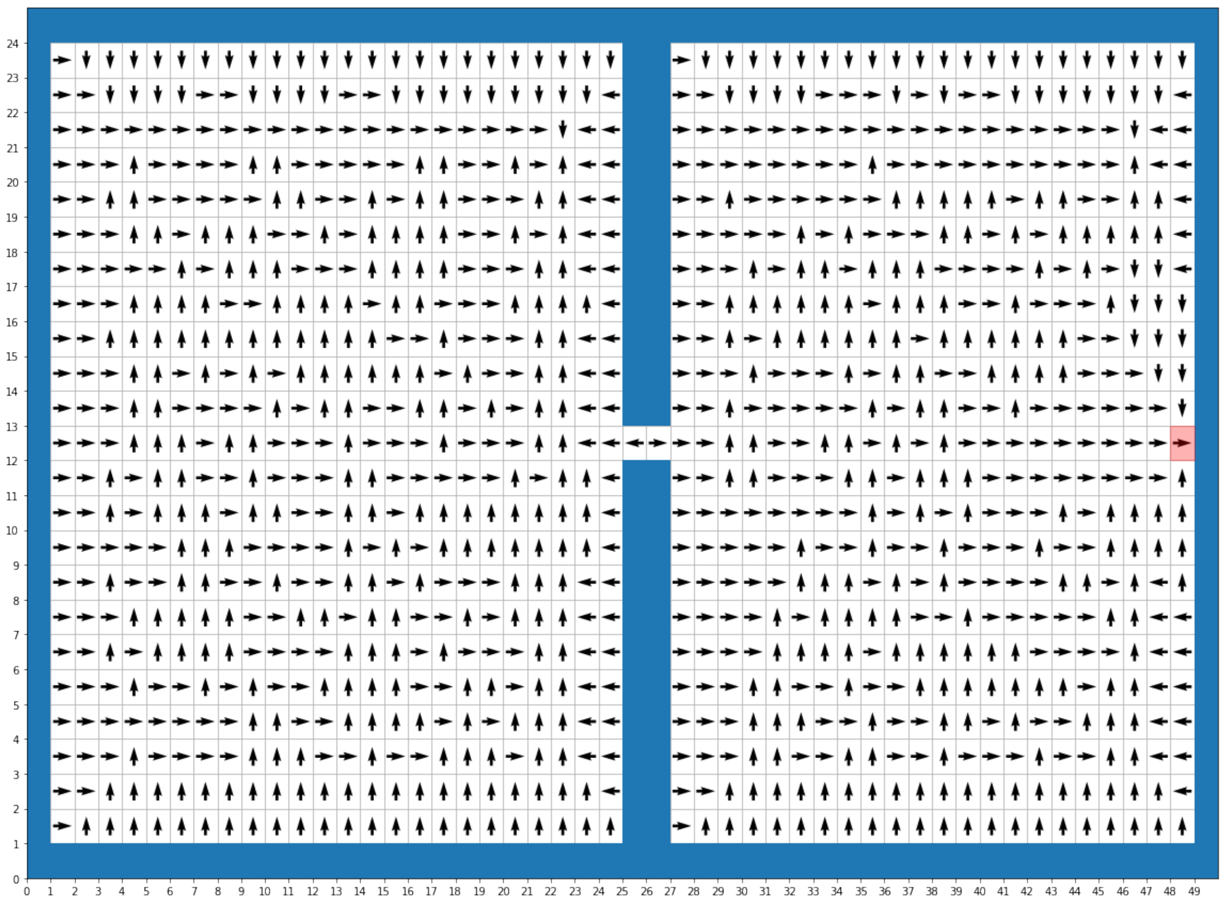


Figure 2: Optimum Policy Map obtained from Value Iteration

1.2 Part b: Varying the Discount factor

The Discount factor is changed to $\gamma = 0.99$.

1.2.1 Observations

Value Function after 20 iterations is given by the image 3

Value Function after 50 iterations is given by the image 4

Value Function after 100 iterations is given by the image 5

In the plots, the blue boundary indicates the walls and the red outlined block is the goal state.

1.2.2 Inference

1. The discount factor used here is very high (= 0.99). This causes the goal state (future state) reward to contribute more to the state-value function of the grid cells away from the goal.

2. As the number of iterations of the algorithm increase, the value function converges more and more and better incorporates the effect of the future states (including the goal state). Hence the state-value of all the grid cells is seen to increase.

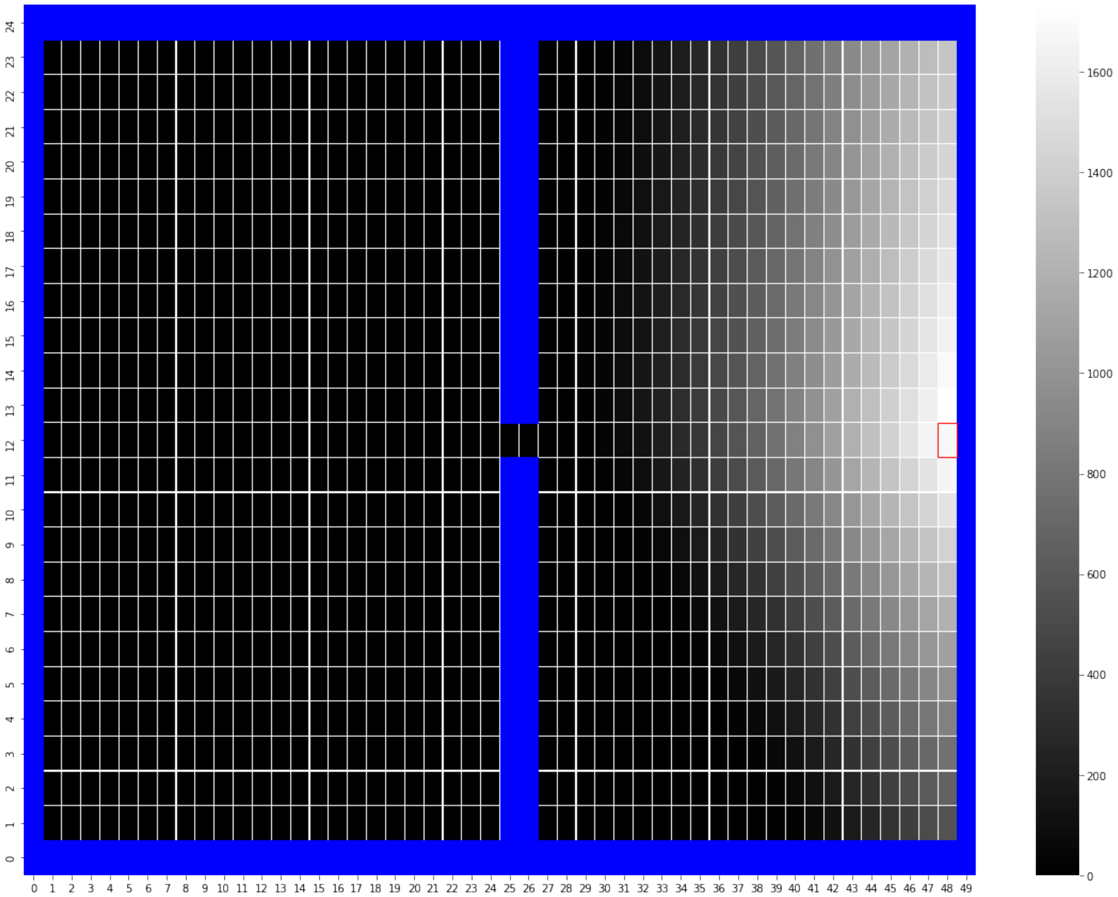


Figure 3: State-Value Function after 20 iterations

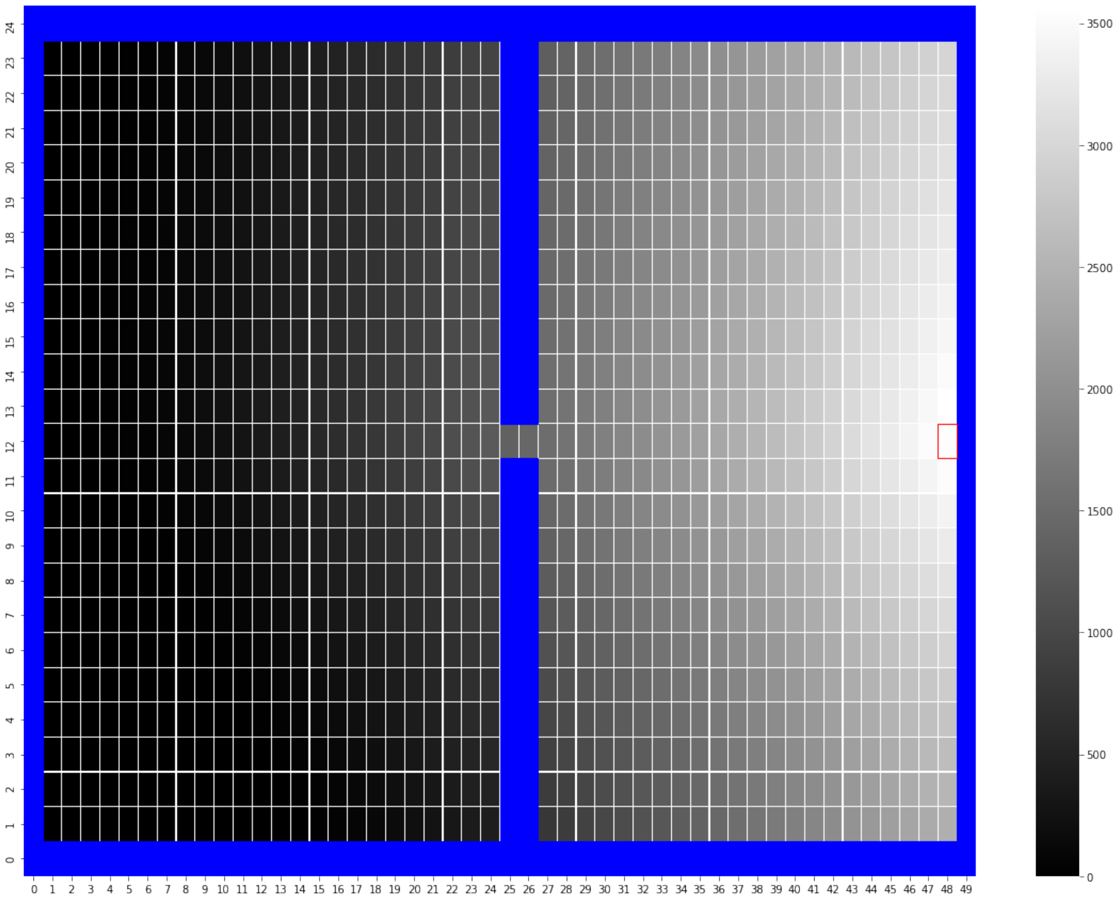


Figure 4: State-Value Function after 50 iterations

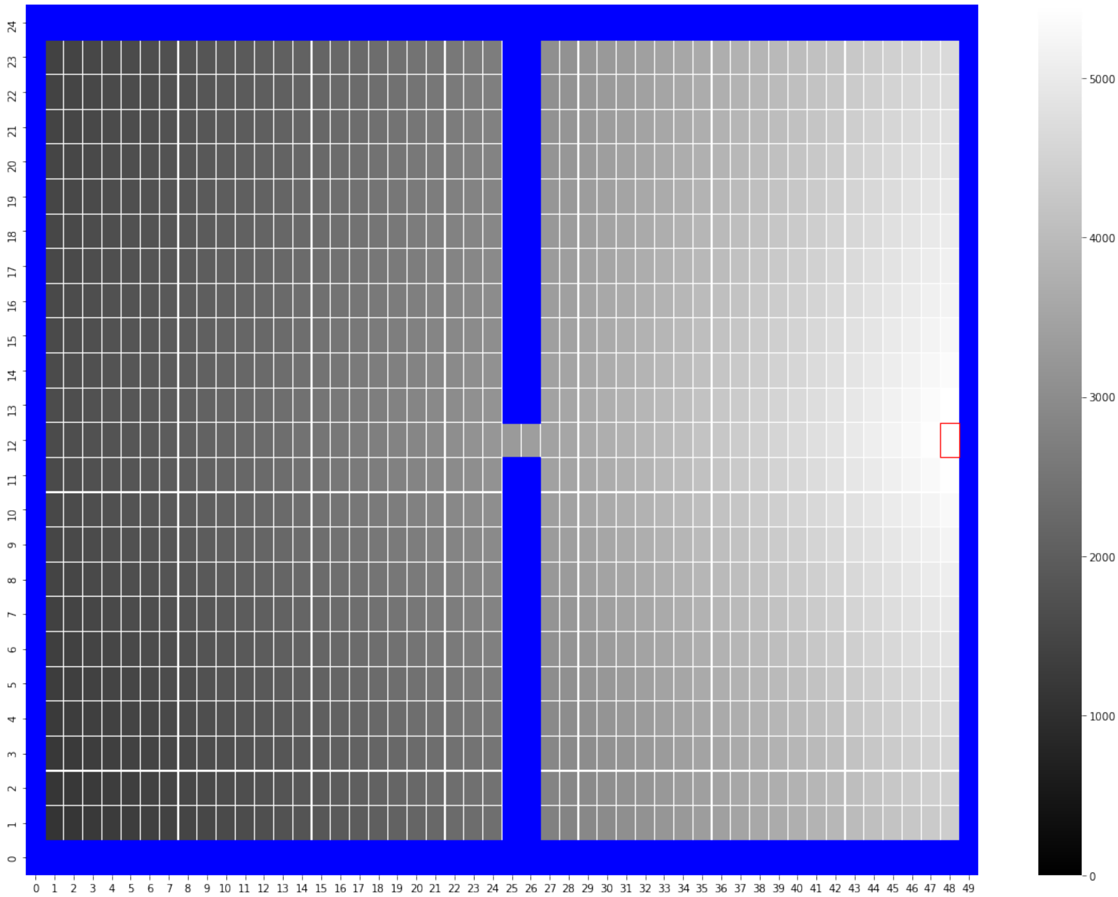


Figure 5: State-Value Function after 100 iterations

1.3 Part c: Policy Simulation

The initial cell in this part is (1, 1).

1.3.1 A Sample Execution of the Policy

A sample execution of the policy obtained after 100 iterations, with $\gamma = 0.99$ is shown in 6

The blue portions indicate the walls and the red cell is the goal.

There are several repeated states and retraced paths in the shown trajectory of the agent, which couldn't be captured.

1.3.2 State-Visitation counts

The state-visitation counts for 200 executions (each with 1000 steps) of the policy generated after 100 value-iteration steps with $\gamma = 0.99$ are visualized in the image 7.

To facilitate better visualization of the number of times a state gets visited, the log plot of the state-visitation counts shown in 8, has been generated. It has been generated by adding 1 to each of the state-visitation counts, to make the log function defined.

The blue patches indicate walls and the red cell is the goal cell. The black cells have 0 visitation count.

1.3.3 Inference

1. The Log-plot of the State-visitation count matrix indicates the regions of the grid where the agent has less tendency to traverse.

2. The plot indicates the agent has maximum tendency to stay at the goal state. This is because the goal state has a high positive reward and is not a terminating state. Thus, the visitation of the goal state keeps on rising till the end of 1000 steps.

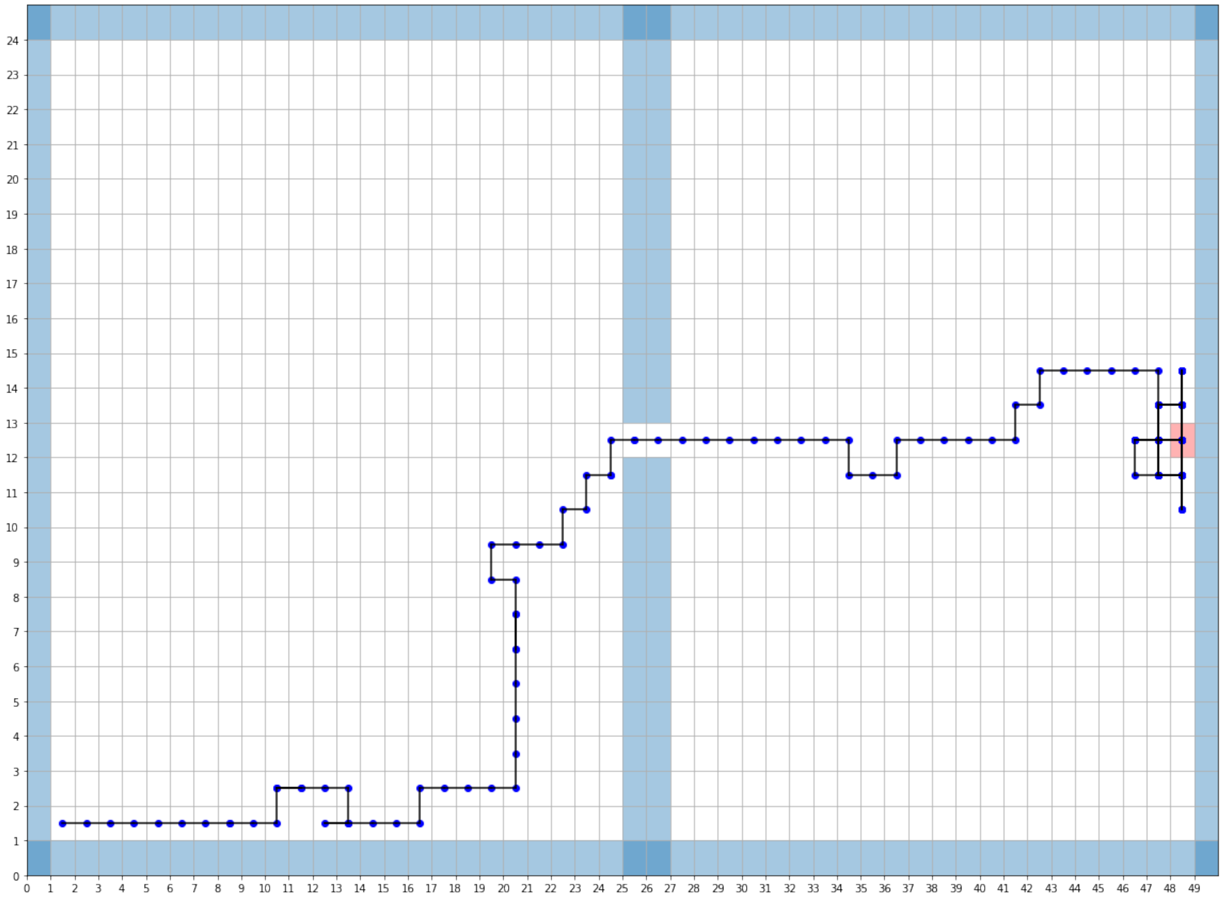


Figure 6: A sample execution of the policy with $\gamma = 0.99$ and 100 iterations (Number of steps = 1000)

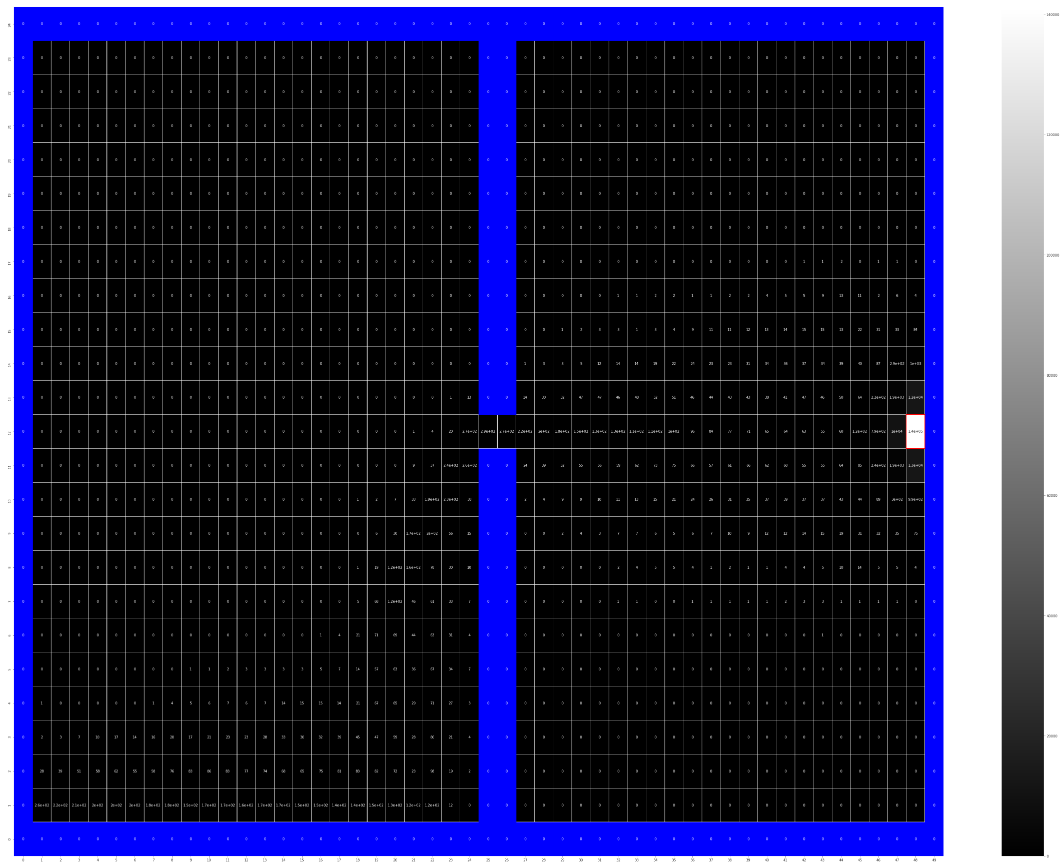


Figure 7: State-Visitation of the various grid cells in 200 executions with 1000 steps each of the policy generated

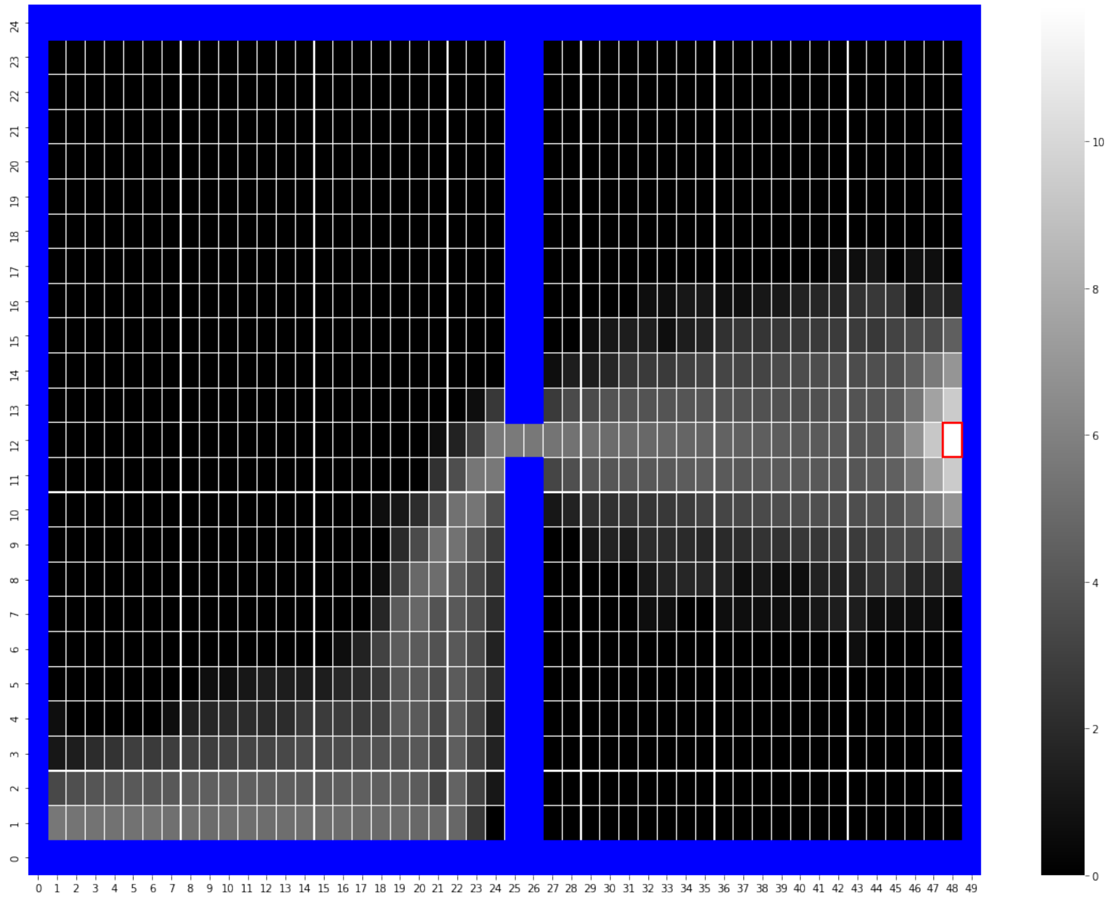


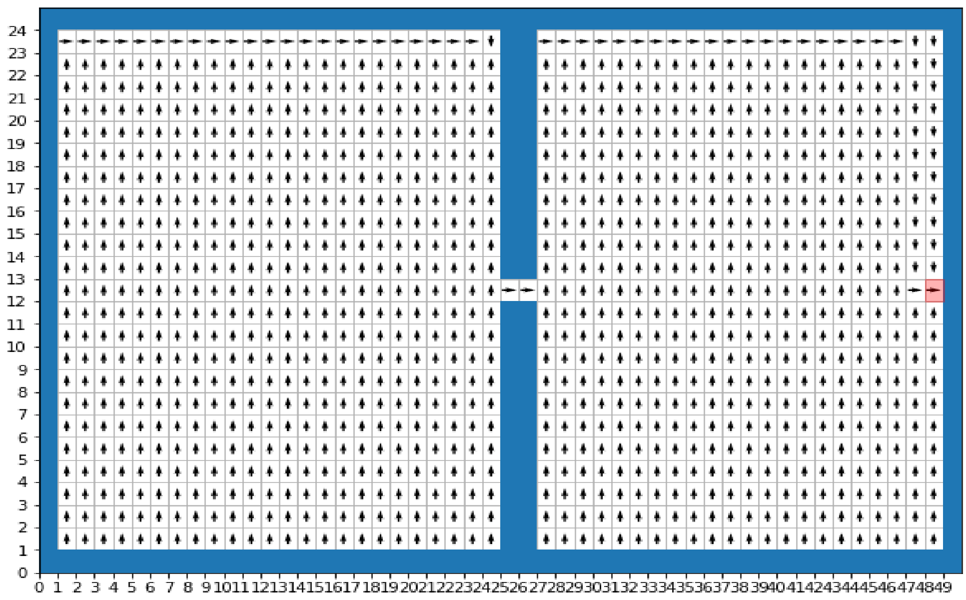
Figure 8: Log plot of state-visitation matrix 7

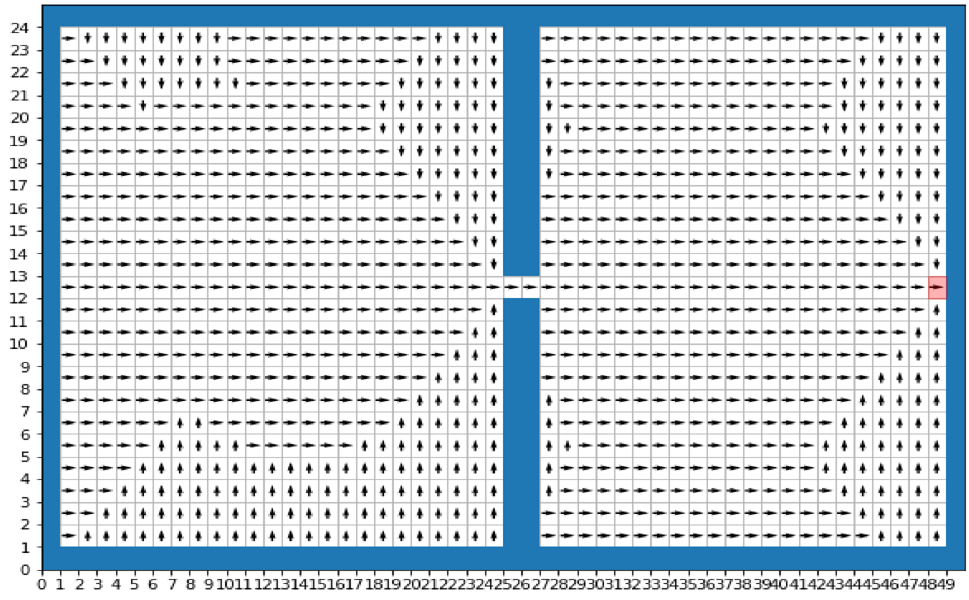
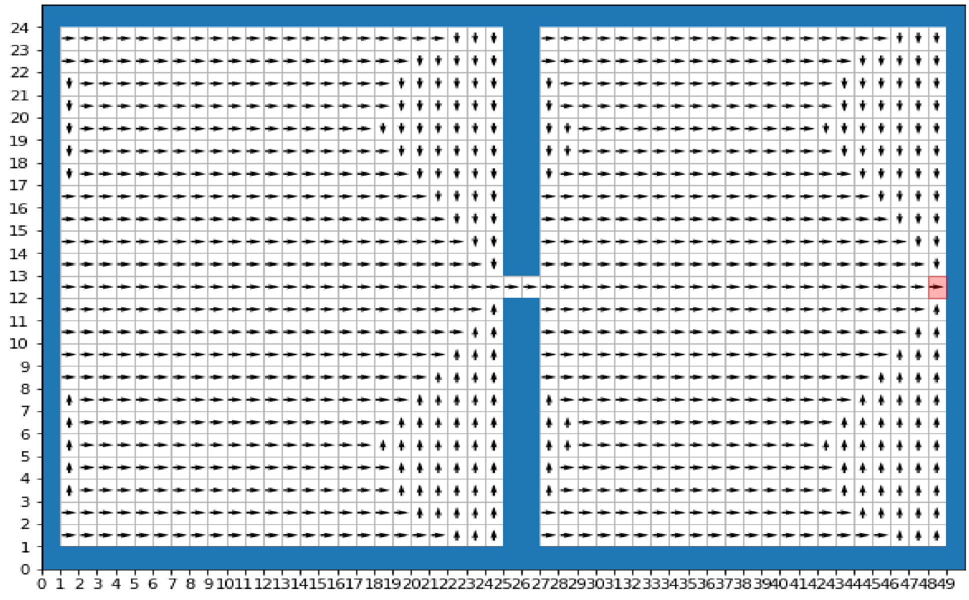
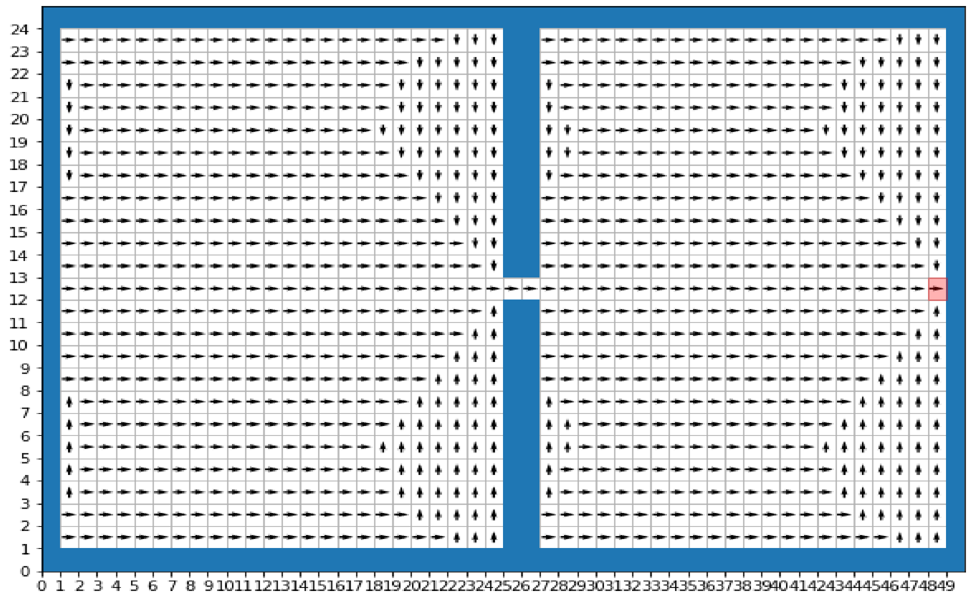
1.4 Part d: Analysing Value Function and Policy convergence

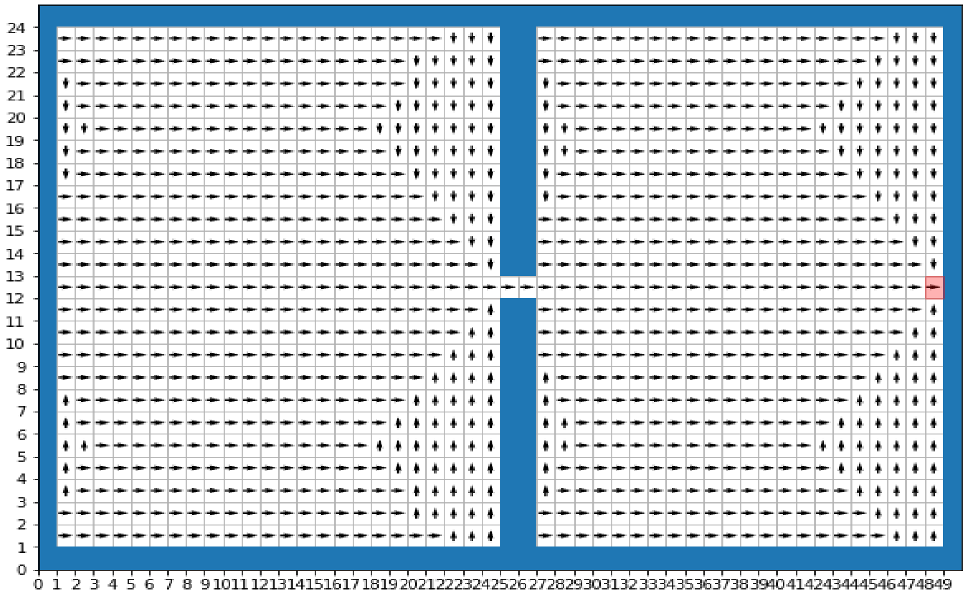
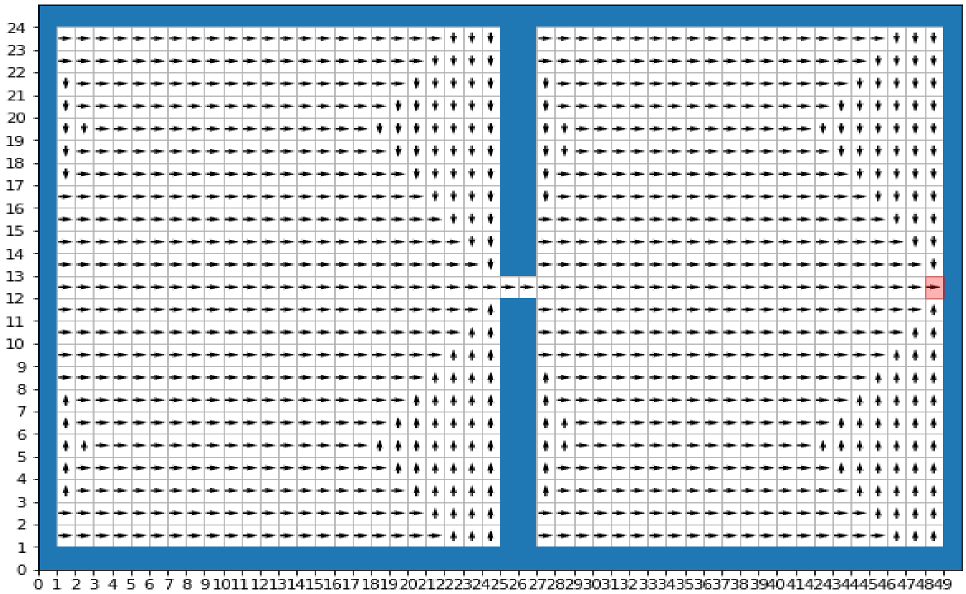
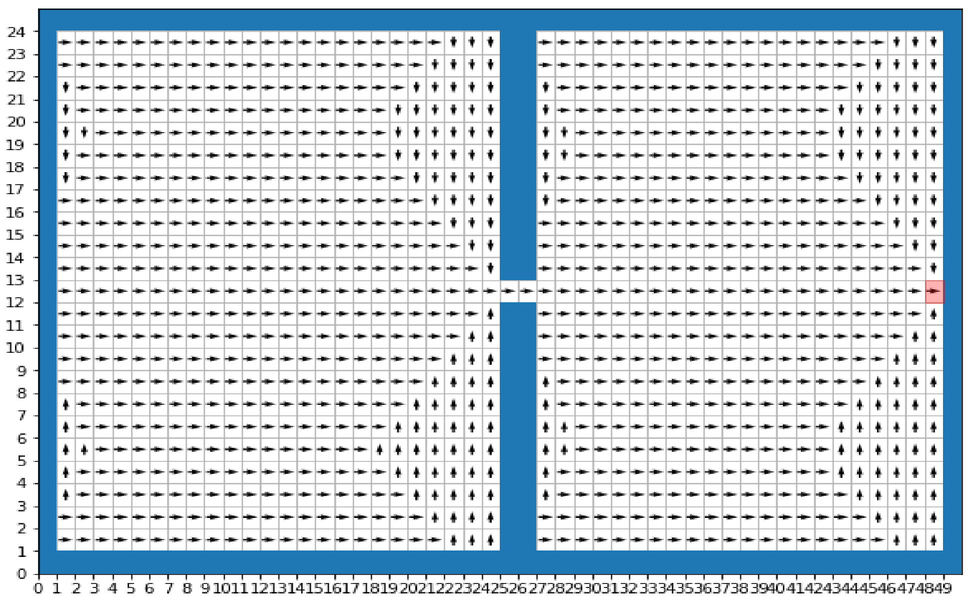
Randomly sampled value iterations are shown in the table below, for the given parameters.

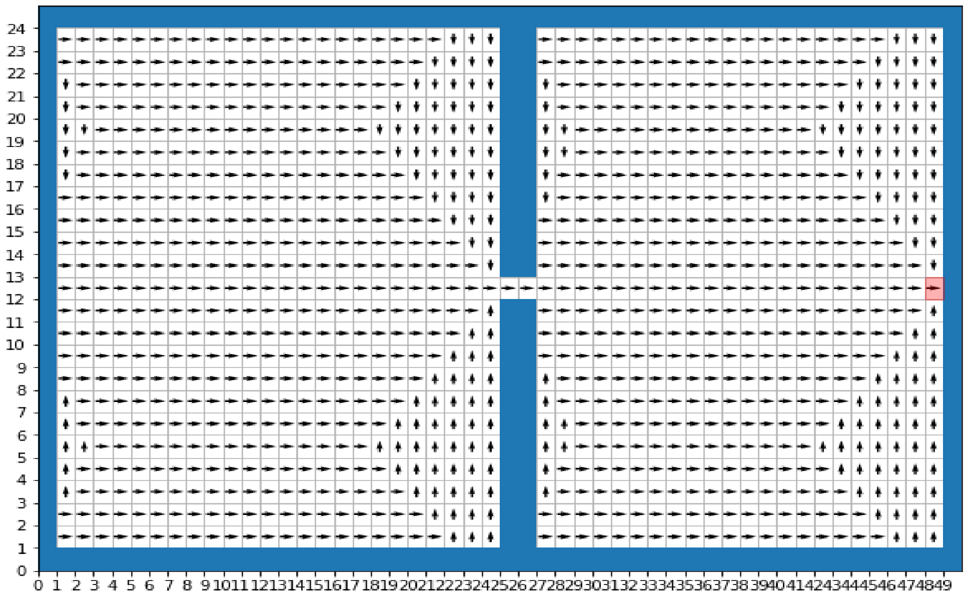
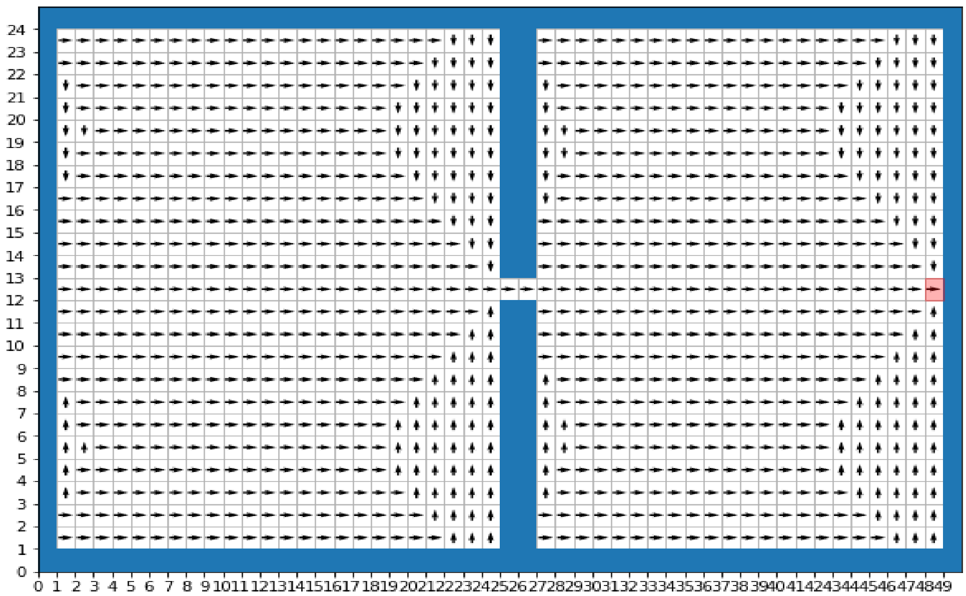
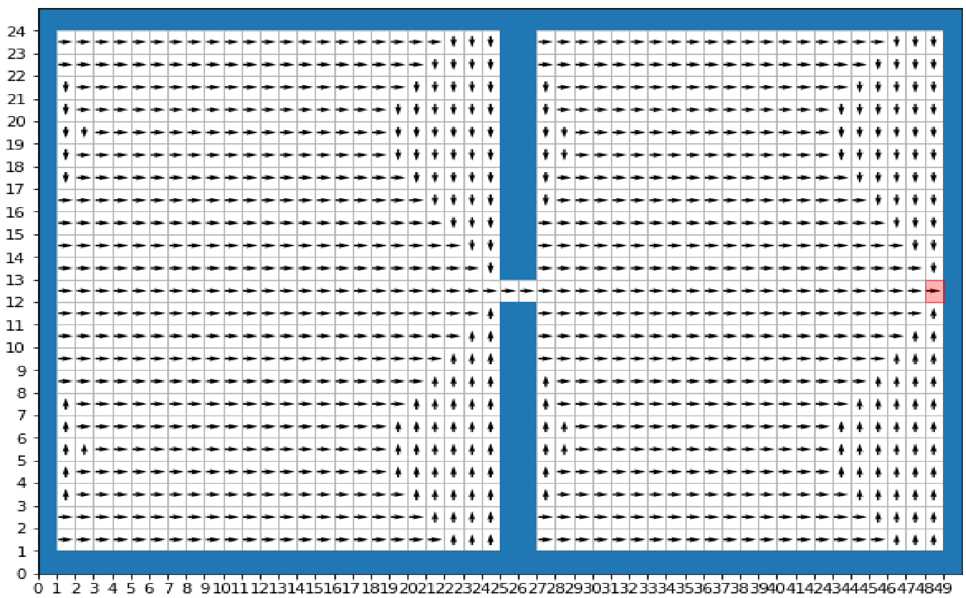
1.4.1 Observations

- $\gamma = 0.99$
- Threshold, $\theta = 0.1$

Iteration	Maxnorm	Policy
1	155.8	

45	56.7	
98	30.2	
101	29.1	

154	15.1	
204	8.1	
271	3.55	

290	2.8	
310	2.2	
370	1.0	

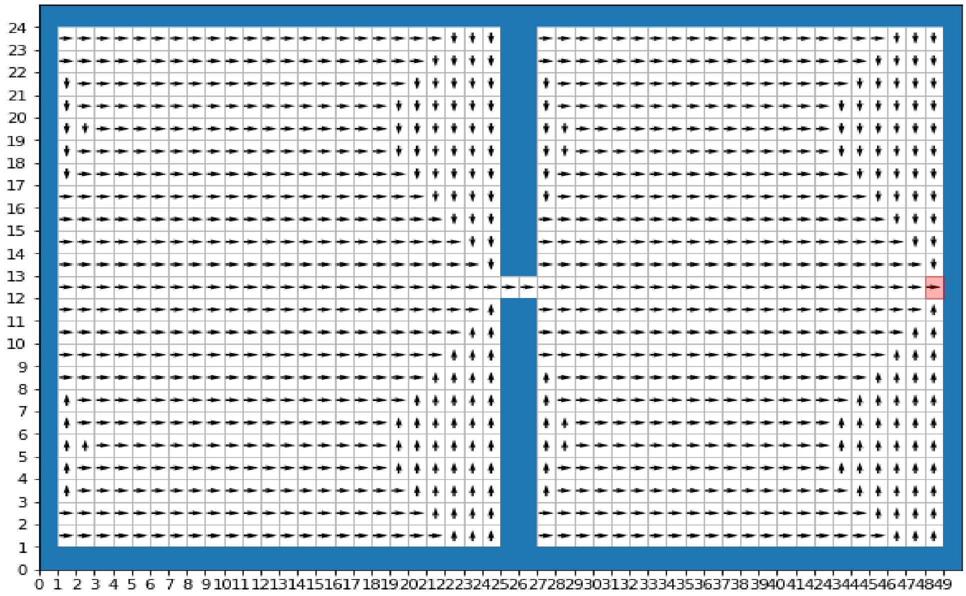
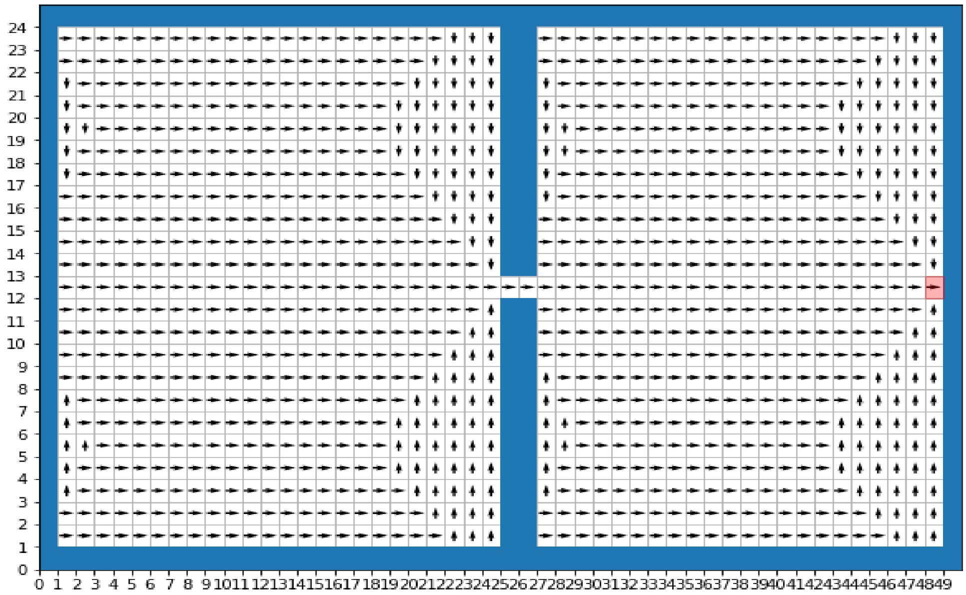
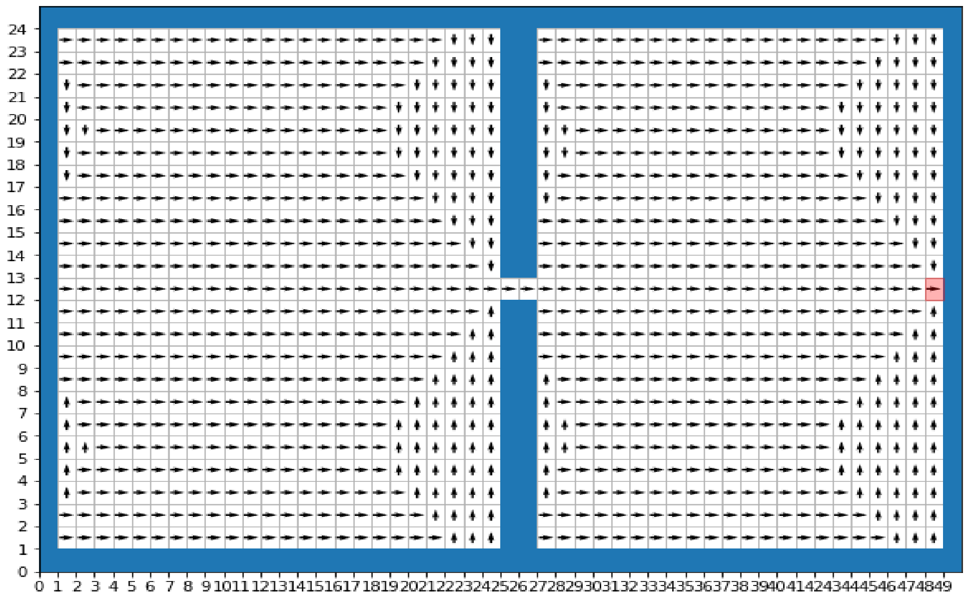
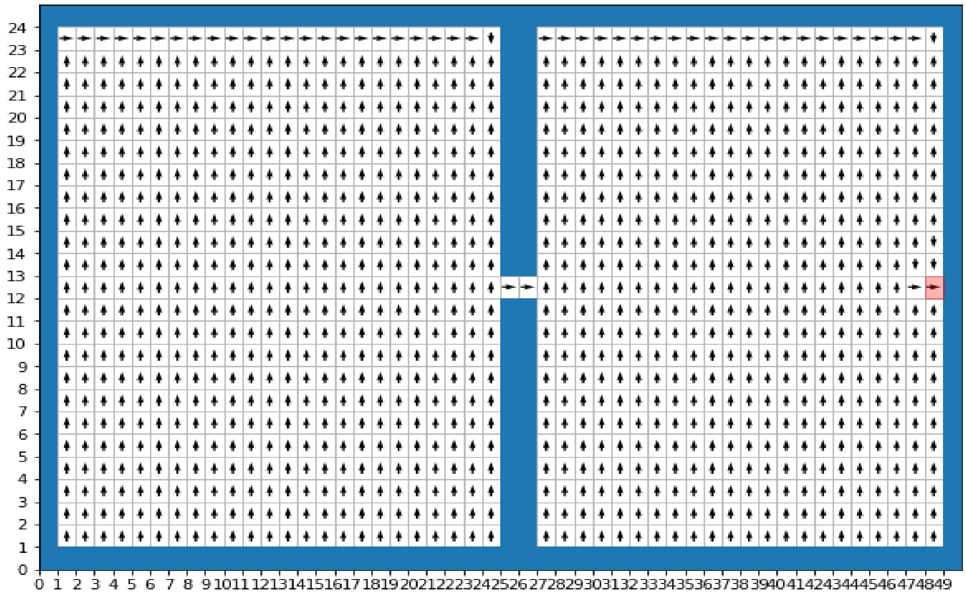
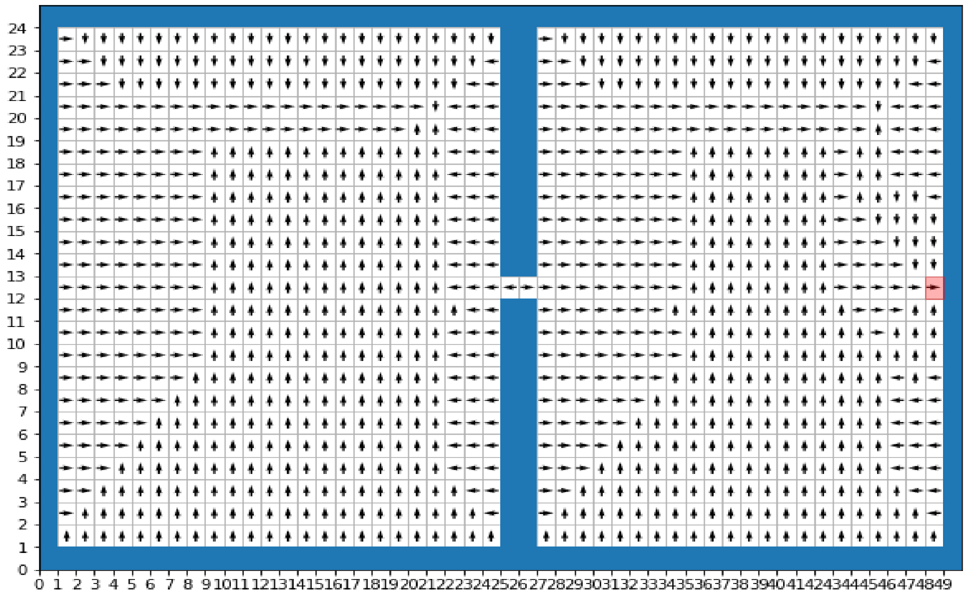
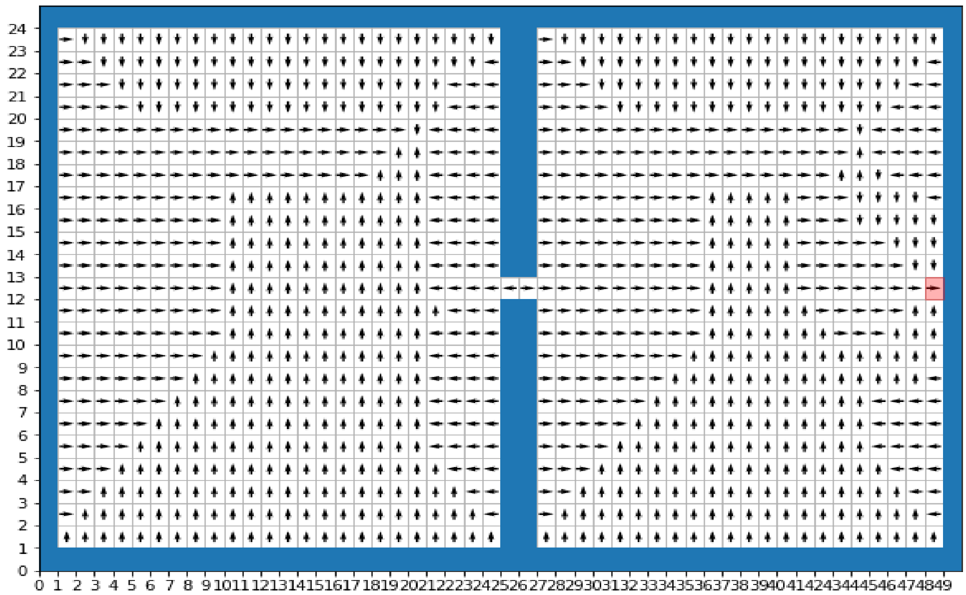
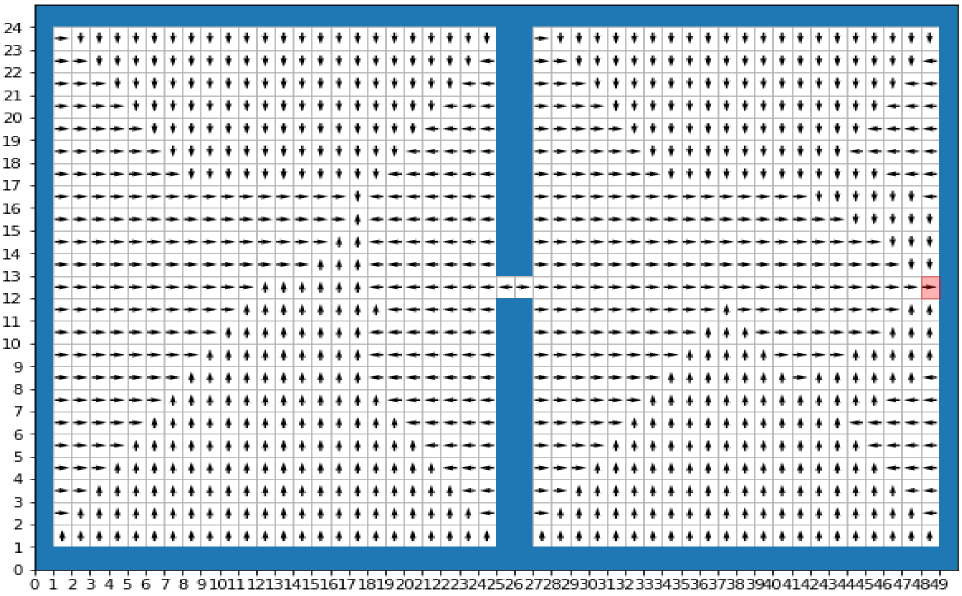
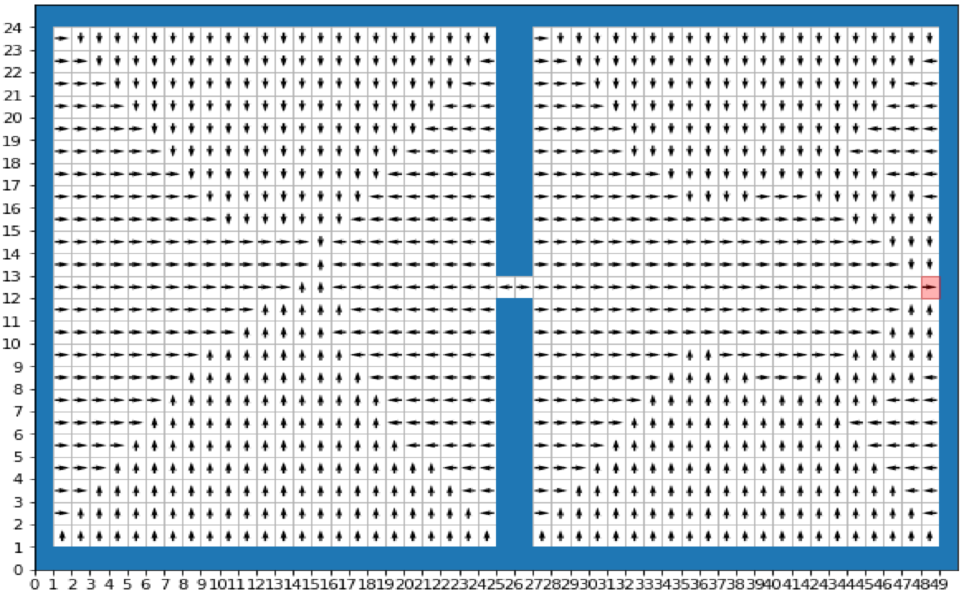
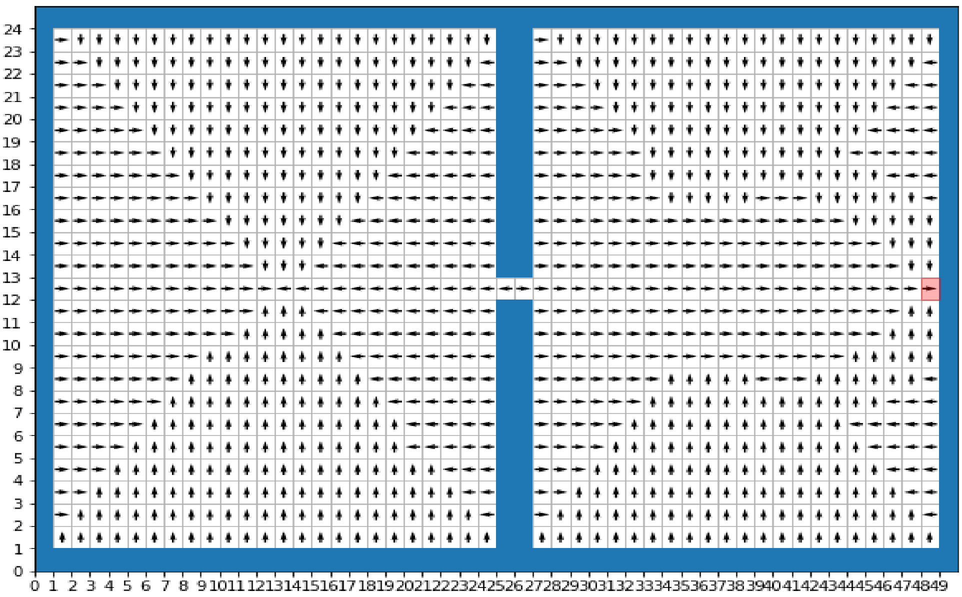
422	0.5	
487	0.24	
560	0.09	

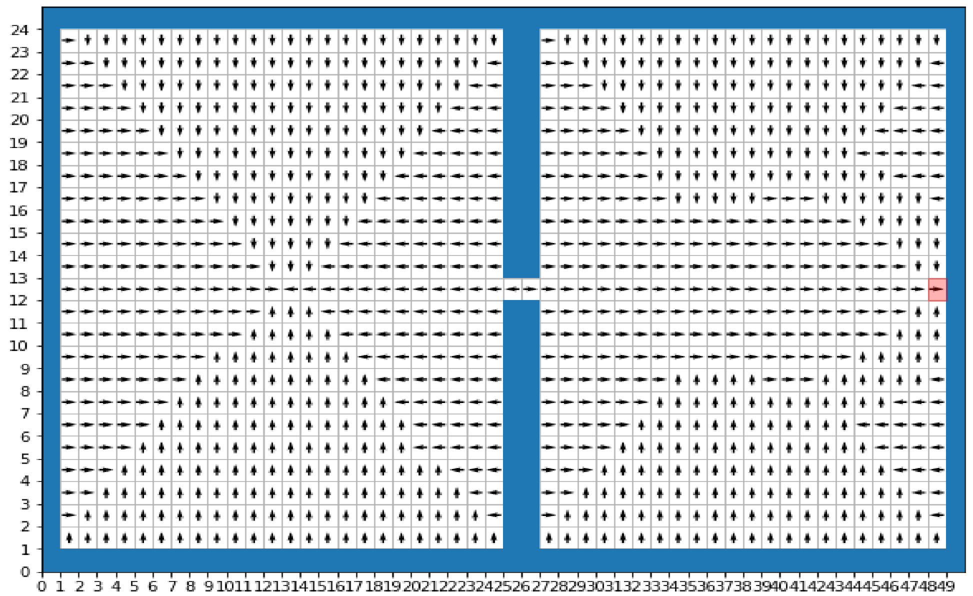
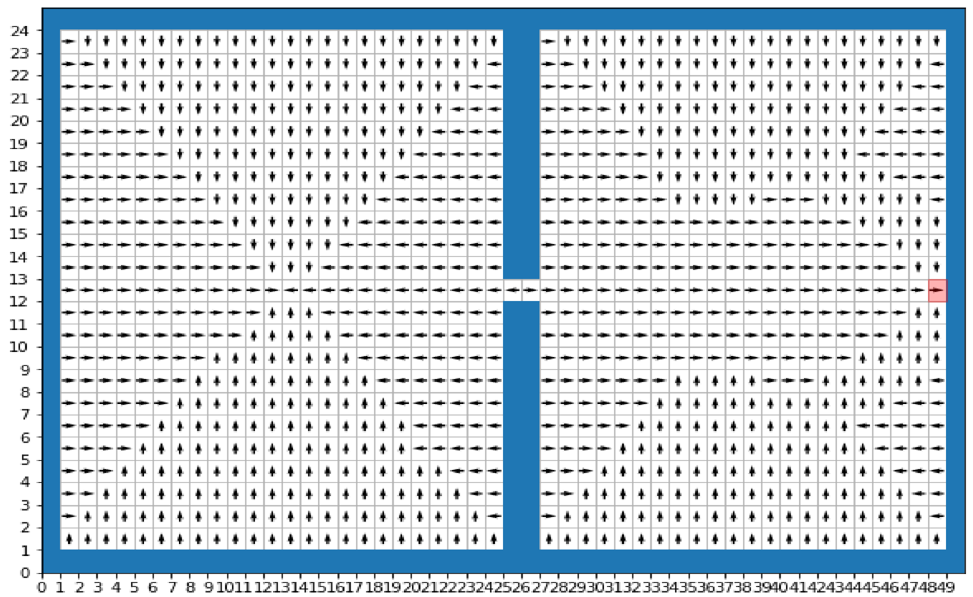
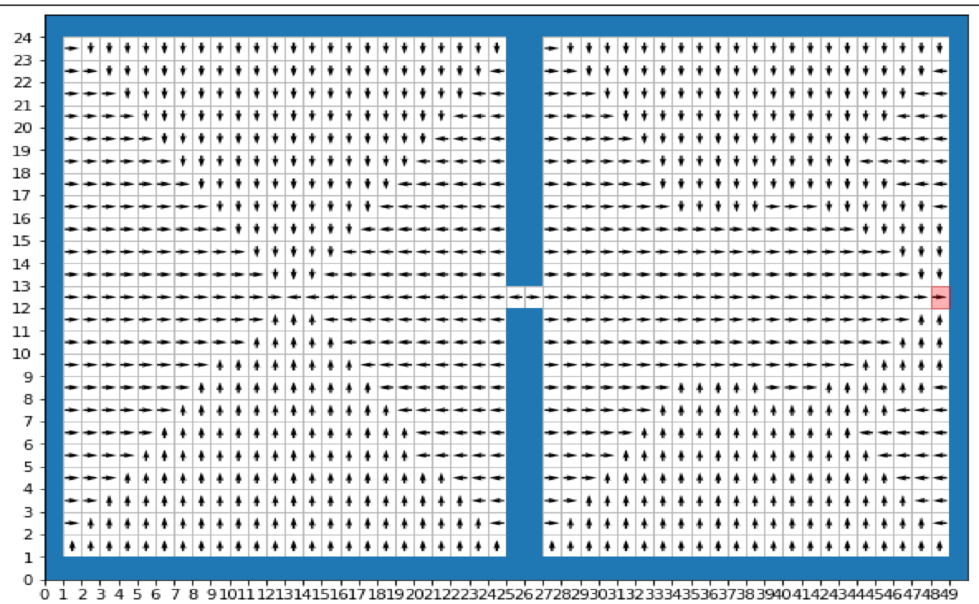
Table 1: Variation of MaxNorm and Policy wrt iteration count for $\gamma = 0.99$

The Policy shown in 1, converges to the final policy after 271 iterations. The maxnorm converges after 560 iterations.

- $\gamma = 0.01$
- Threshold, $\theta = 1e - 50 \approx 0$
- Iterations done $= 100$

Iteration	Maxnorm	Policy
1	79.57	
5	4.47e-07	
7	3.31e-11	

11	1.86e-19	
14	1.21e-25	
18	1.54e-33	

19	6.01e-36	
20	7.52e-37	
21	7.34e-40	

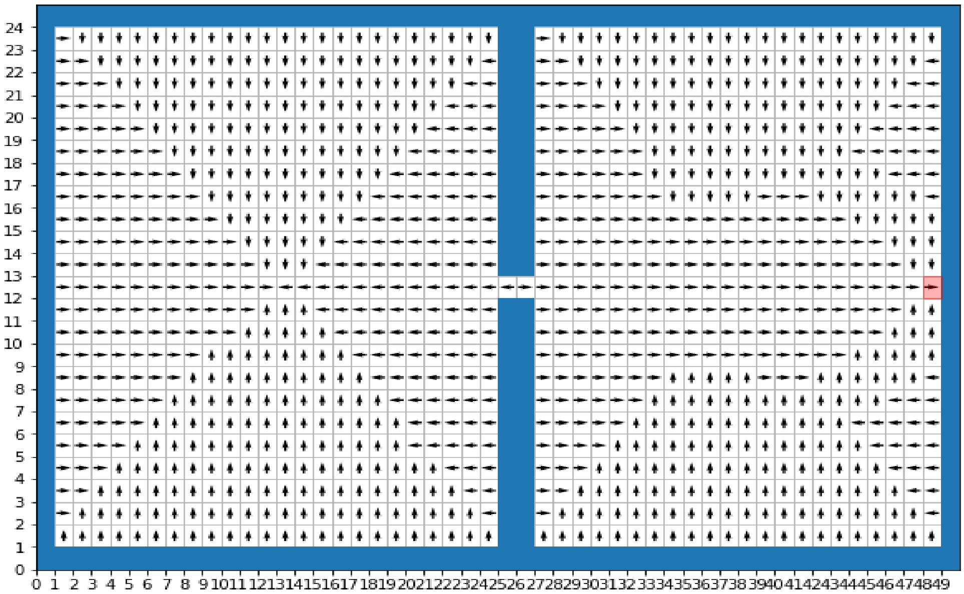
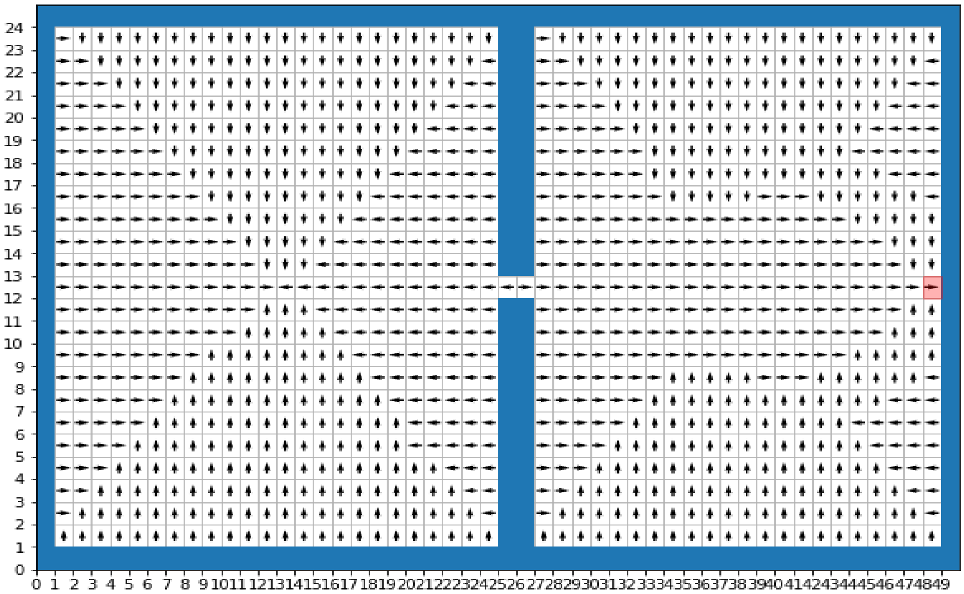
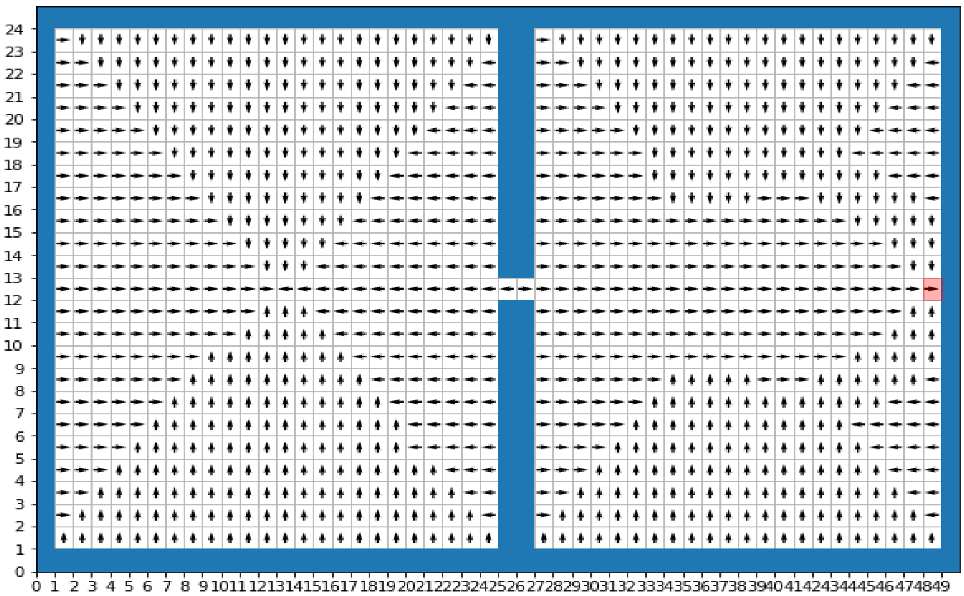
22	2.86e-42	
23	0	
24	0	

Table 2: Variation of MaxNorm and Policy wrt iteration count for $\gamma = 0.01$

The Policy shown in 2 converges after 19 iterations of Value iteration procedure. The maxnorm becomes a very small value, that it gets approximated to 0.

The variation of the maxnorm with the number of iterations for $\gamma = 0.99$ and $\gamma = 0.01$ is shown in 9

1.4.2 Inference

1. For $\gamma = 0.99$, the maxnorm value falls very slowly as is clear from 9. The gradient in maxnorm vs iterations for $\gamma = 0.01$ is quite high. This is because the state-value function values for $\gamma = 0.01$ is very small for most of the states. This is visible from 1 results. So the maxnorm value will fall quickly for smaller values of γ. As the values are small, maxnorm will also be small. For higher values of γ, the state-values are higher due to higher influence of the goal state reward. So the convergence of the state-value function takes more number of iterations for higher values of γ.
2. As the maxnorm for $\gamma = 0.01$ falls very rapidly, its threshold was kept very low and emphasis was given on the number of iterations.
3. The Policy function is seen to converge after 271 iterations for $\gamma = 0.99$. Policy function converges after 19 iterations for $\gamma = 0.01$. Thus, the policy converges much before the convergence of the value function in both the cases, as expected.
4. As the maxnorm for $\gamma = 0.01$ falls much faster, policy and state-value convergence is much faster as compared to those for $\gamma = 0.99$.

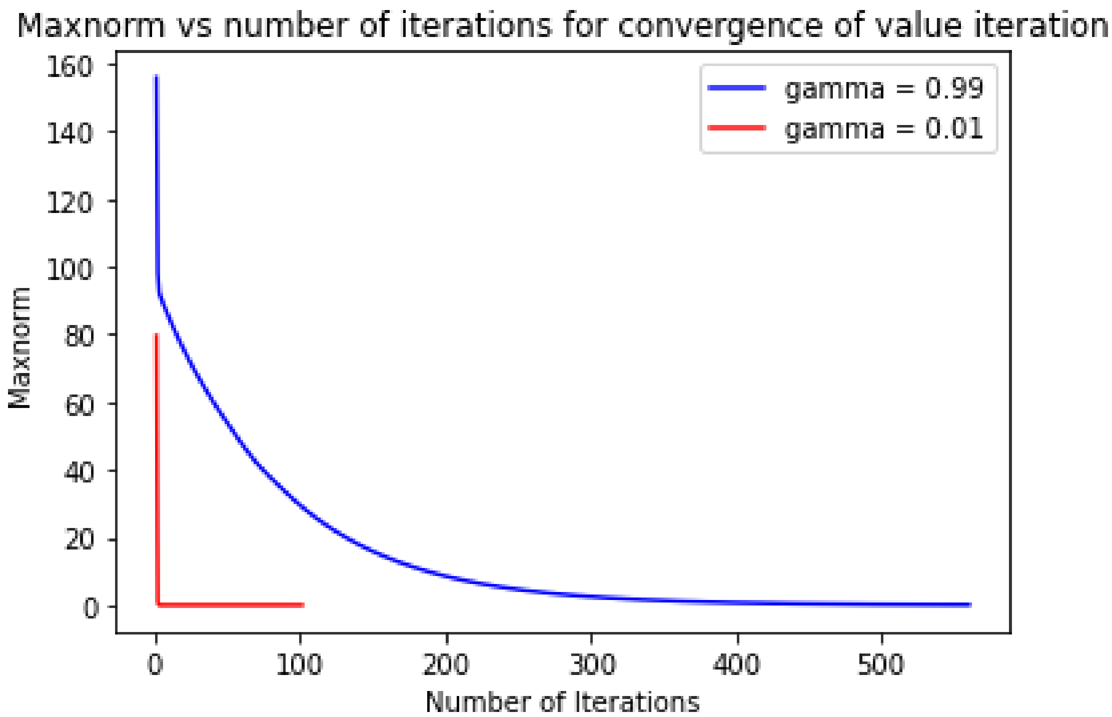


Figure 9: Variation of maxnorm with iterations