Name: Rajat Jaiswal **Entry Number:** 2017CS50415

1 Part 1: Solving an MDP

Consider a mobile robot in the grid world domain as shown in figure 1 below.

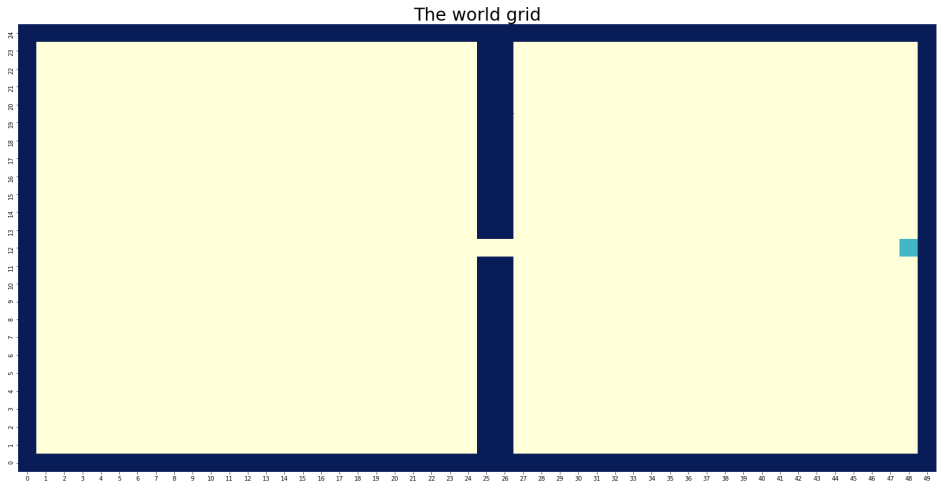


Figure 1: The world grid of the robot with walls and goal state.

(a) Solve for a policy using value iteration for the problem stated. Use a discount factor of $\gamma = 0.1$ and a threshold of $\theta = 0.1$

Solution:

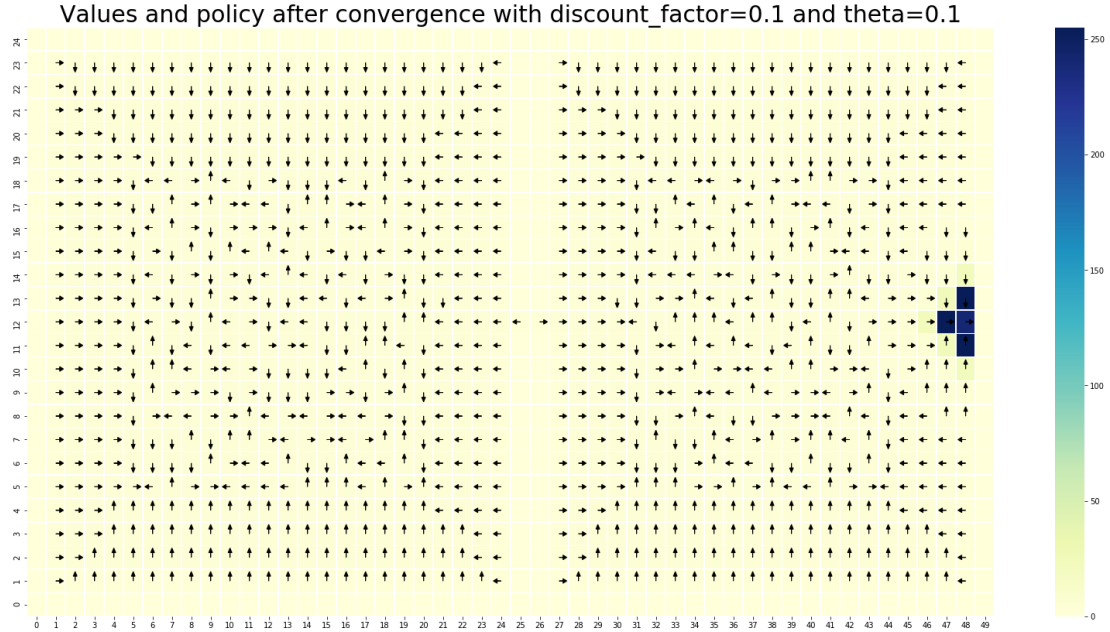


Figure 2: Result of value iteration and the final policy assigned to each state after convergence with discount factor($\gamma = 0.1$) and convergence threshold($\theta = 0.1$).

The discount factor is very low and hence the computation in previous iterations is not given much weightage so in a way the computation depicts just a few steps look ahead. Therefore, the value near

the goal are high as the reward for the goeal is high. The policy for the cells which are two three steps away from the wall is to go in opposite direction of nearest wall as collision with wall induces negative reward. And in the center it is almost randomly assigned. The rewards in transitioning to middle cell is 0 so the values assigned are close to 0. Due to low discount factor the algorithm is not able to induce the reward/values from the goal state to other state.

(b) Increase the discount factor to $\gamma = 0.99$ and plot the value function at iterations 20, 50 and 100.

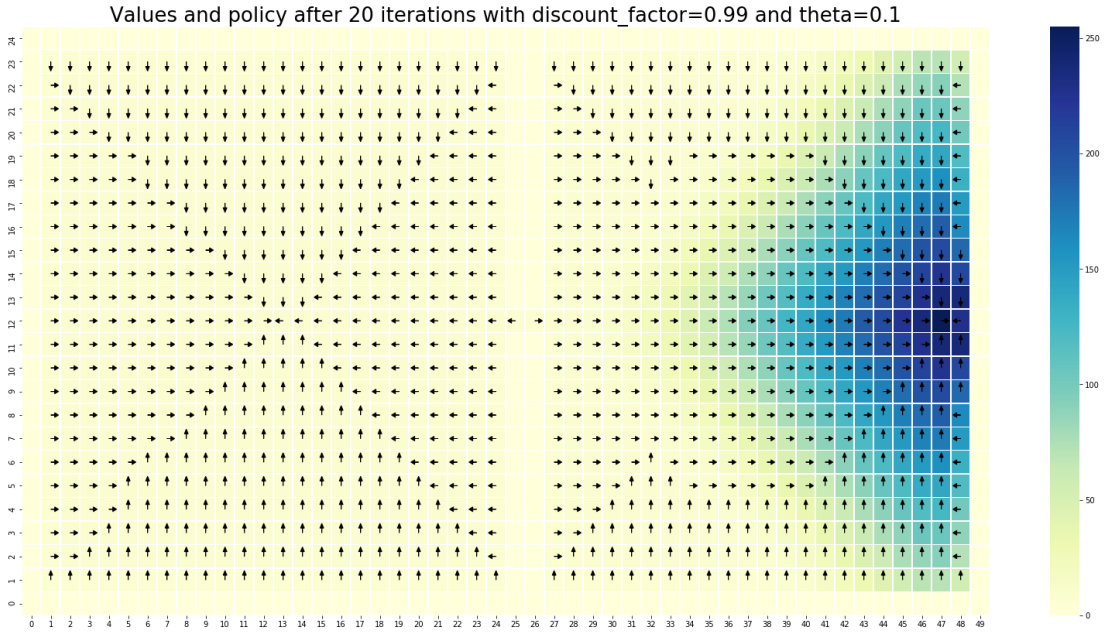


Figure 3.1: Values and the policy assigned to each state after 20 iterations with discount factor($\gamma = 0.99$) and convergence threshold($\theta = 0.1$).

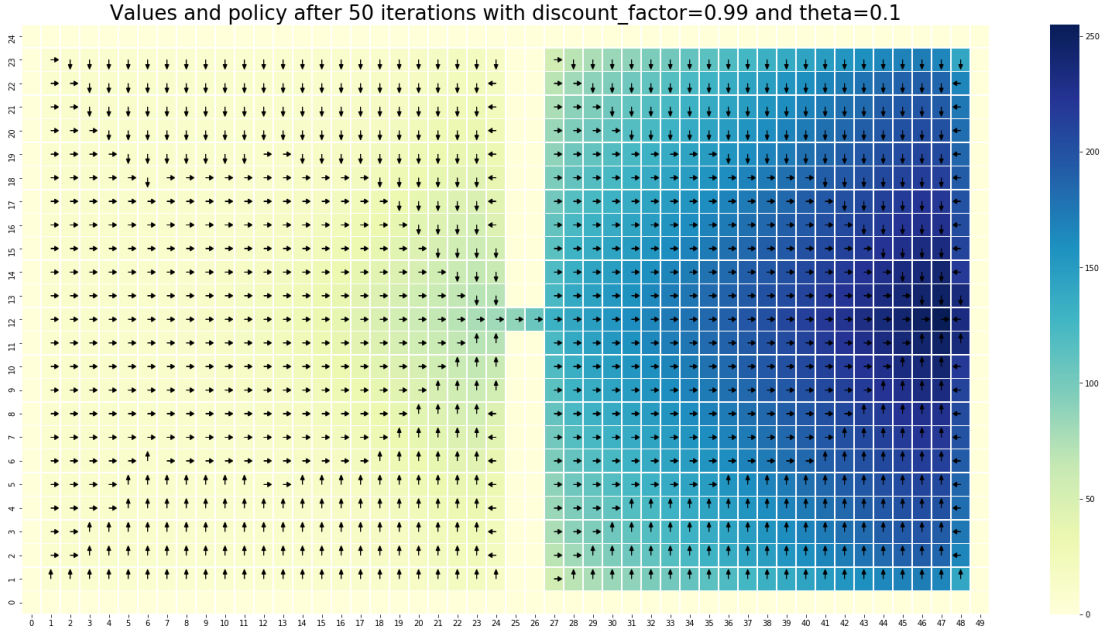


Figure 3.2: Values and the policy assigned to each state after 50 iterations with discount factor($\gamma = 0.99$) and convergence threshold($\theta = 0.1$).

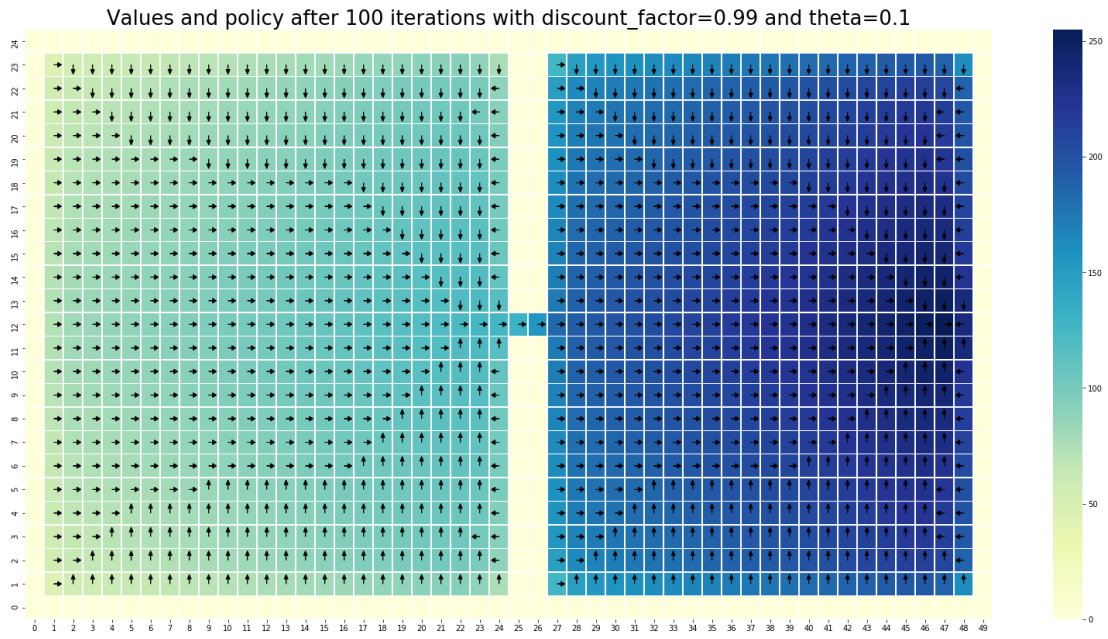


Figure 3.3: Values and the policy assigned to each state after 100 iterations with discount factor($\gamma = 0.99$) and convergence threshold($\theta = 0.1$).

A video has been attached that shows how the policy and value assigned changes over the iteration. Due to high discount factor the value assigned to nearby states creates a high ripple effect and over the iterations it is transmitted to other states as well and that value gets stabilised as the iteration increases and once that happens the policy assigned also gets stabilised.

(c) Draw a sample execution of the policy with $\gamma = 0.99$ estimated above starting from an initial state $(1, 1)$. Simulate policy execution 200 times and determine the number of times a state is visited.

Solution:

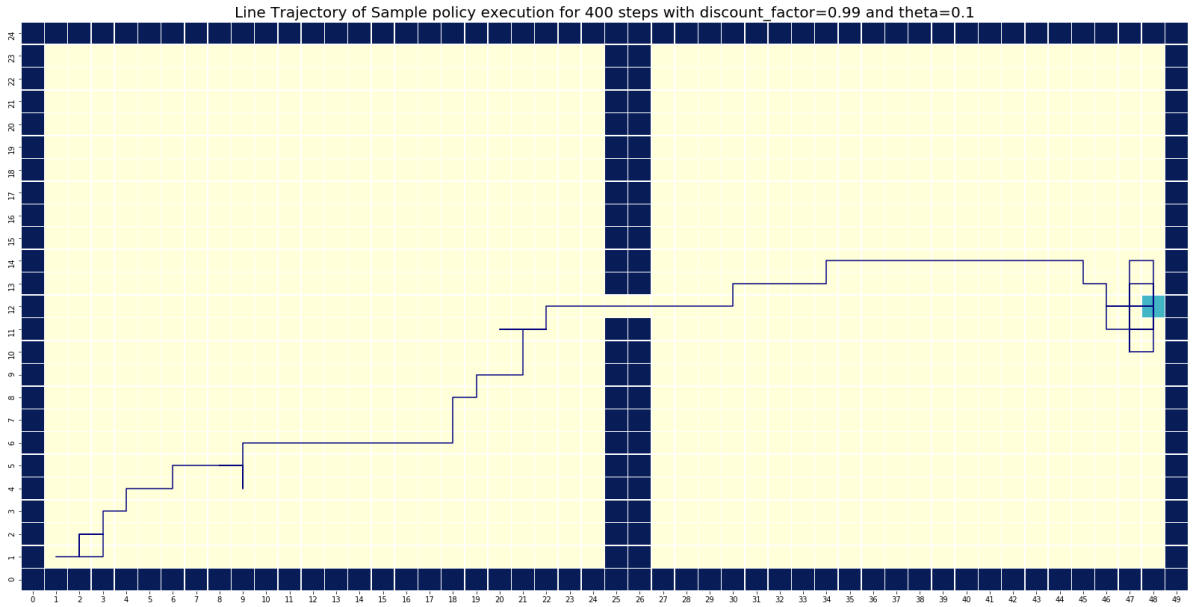


Figure 4.1: A line trajectory of the robot depicting a sample execution of policy assigned after 100 iterations with $\gamma = 0.99$ & $\theta = 0.1$. The execution is done for 400 steps.

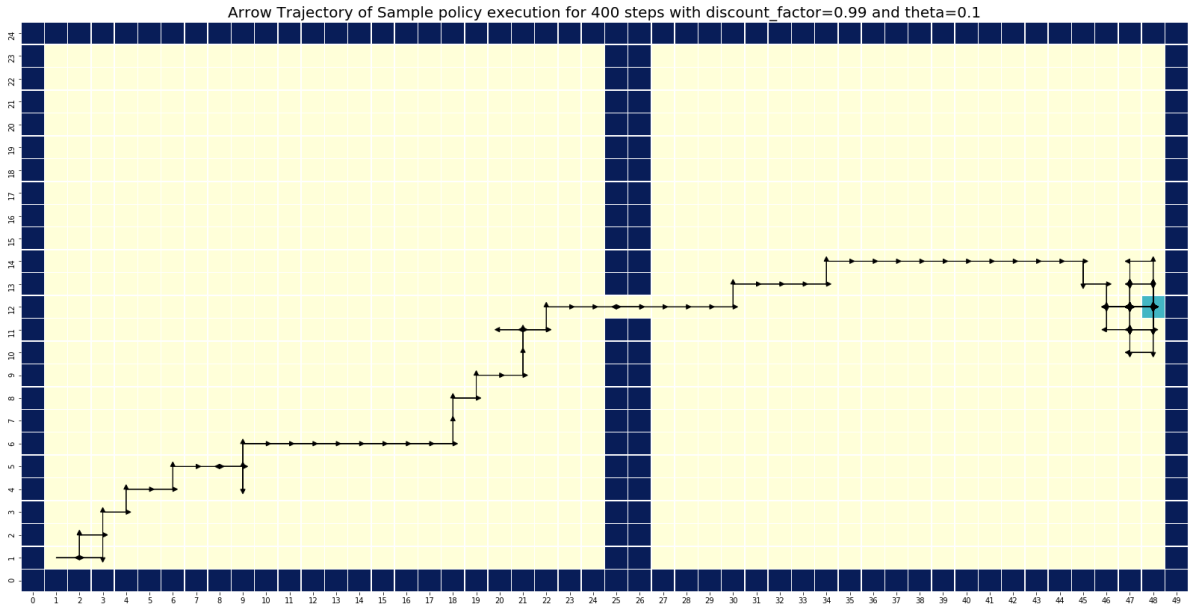
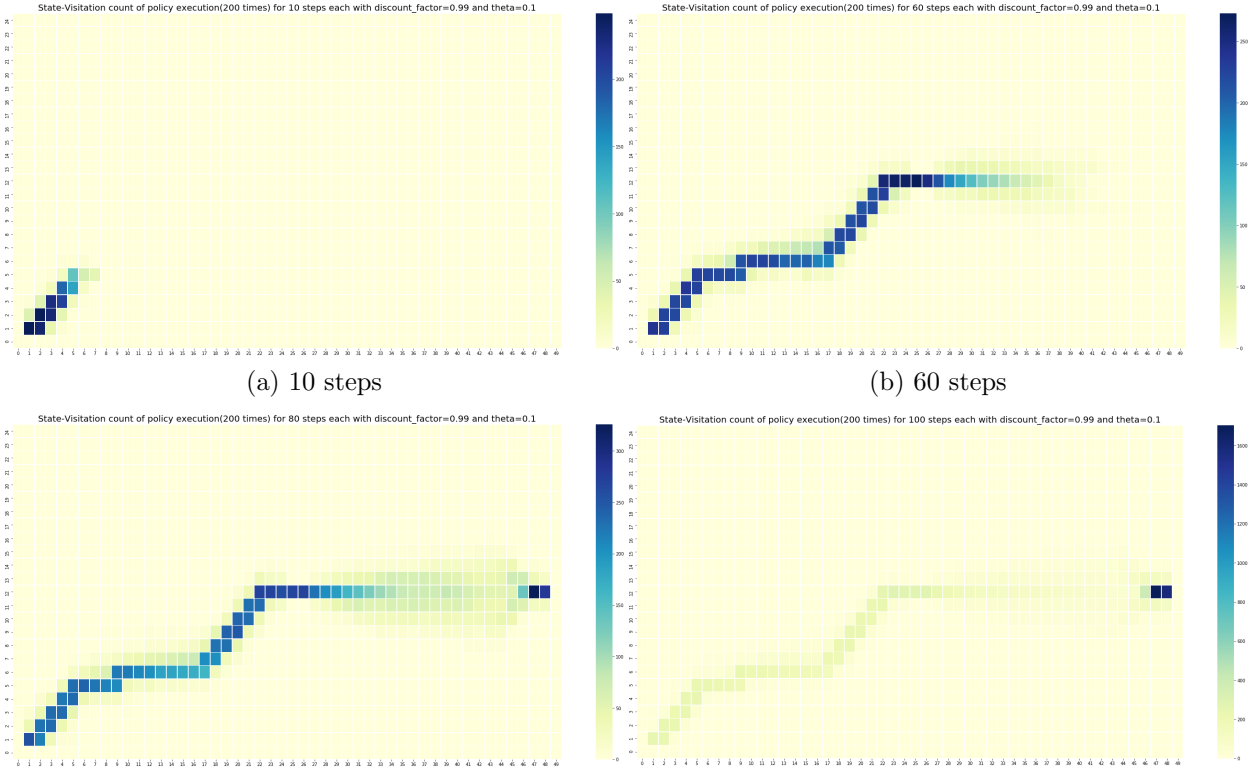


Figure 4.2: An arrow trajectory of the robot depicting a sample execution of policy assigned after 100 iterations with $\gamma = 0.99$ & $\theta = 0.1$. The execution is done for 400 steps.

A video of the sample execution of the policy assigned after 100 iterations (with $\gamma = 0.99$ & $\theta = 0.1$) has been attached in the submission. The video depicts the first 100 steps taken by the robot.



(a) 10 steps (b) 60 steps

(c) 80 steps (d) 100 steps

Figure 5: Aggregate number of times each state has been visited after 200 episodes of the policy have been run, each for n number of steps and each episode with initial position of $(1, 1)$.

The robot almost takes the minimum cost path except for a few steps other than the policy assigned and reaches the goal in more or less 75 steps.

(d) Study how the max-norm for the successive value functions decreases over successive iterations for the discount factors $\gamma = 0.99$ and $\gamma = 0.01$.

Solution:

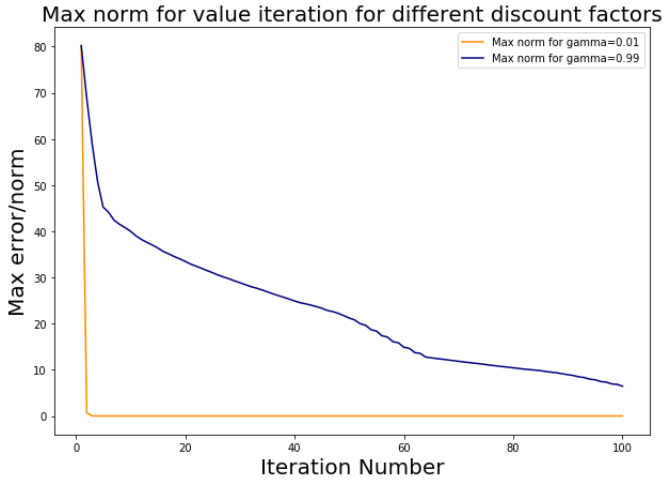


Figure 6.1: $max - norm$ distance between successive iteration over values assigned to different states.

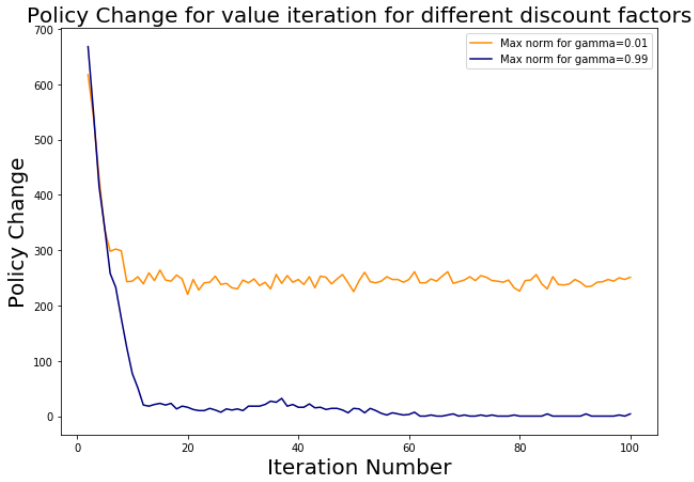


Figure 6.2: Number of changes in policy assignment between successive iterations.

The max-norm for low γ converges very quickly because the ripple created for low γ has small effect and settles down really quickly whereas it is different for high γ. But one more thing that can decide whether a policy has converged or not if we see the number of changes in action assigned to each state between successive policy is low or not. If action assigned to 99% or more states haven't changed over 10 or so successive iterations hasn't changed then we can say that the policy has converged. Using this logic we can conclude that the policy assignment has converged for $\gamma = 0.99$ in around 50 iterations. Though the value has still not converged. This is also visible in Figure 3.2 and Figure 3.3 as there is not much change in policy assigned to each state in the two figures. So we can have two convergence criteria and if either of those two gets fulfilled then we can conclude that the policy has converged. In case of $\gamma = 0.01$ it doesn't get satisfied as the action to middle states are getting randomly assigned in each iteration due to low ripple effect and hence policy changes are more, but here the value has converged so we can say that the algorithm gets converged. Two videos for each γ have been attached.