CS9670: Lab 1 Report

The MDP solver code was run on `test_mdp_maze.py`. The results for the prompts are given below.

1. *Report the policy, value function and number of iterations needed by value iteration when using a tolerance of 0.01 and starting from a value function set to 0 for all states.*

> We get:

```
policy = array([3, 3, 3, 1, 3, 3, 3, 1, 1, 3, 3, 1, 3, 3, 3, 0, 0], dtype=int32)
V = array([ 60.62388836,  66.03486523,  71.80422632,  77.09196339,
        59.81429704,  65.18237783,  77.83066489,  84.14118981,
        58.09361039,   7.98780239,  84.86704922,  91.78159355,
        69.49584217,  76.80962081,  91.78159355, 100.        ,
         0.        ])
nIterations = 20
```

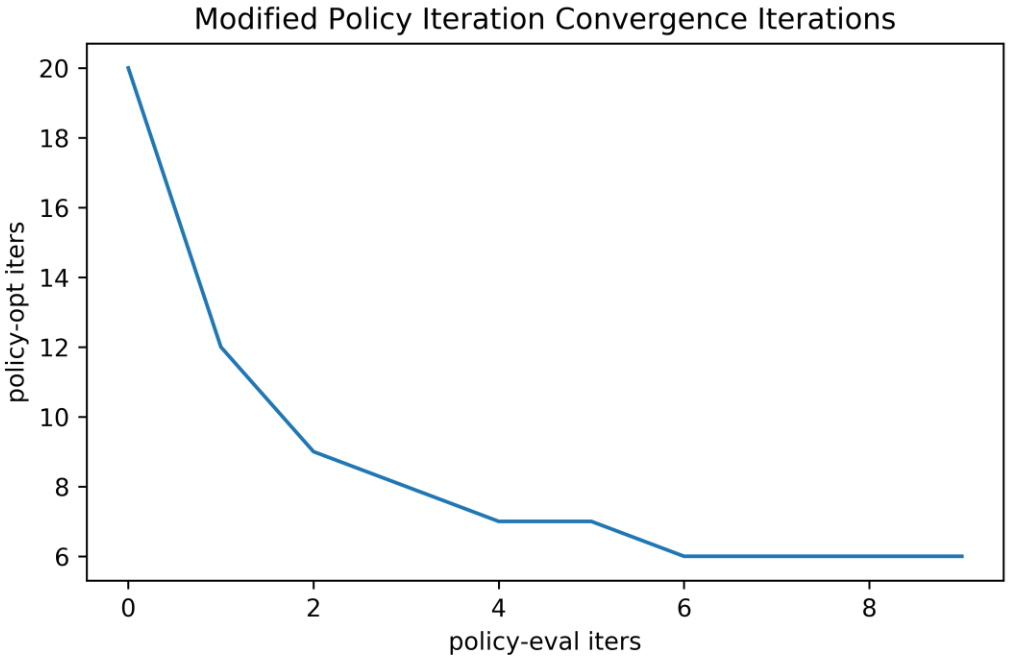
2. *Report the policy, value function and number of iterations needed by policy iteration to find an optimal policy when starting from the policy that chooses action 0 in all states.*

> We get:

```
policy = array([3, 3, 3, 1, 3, 3, 3, 1, 1, 3, 3, 1, 3, 3, 3, 0, 0], dtype=int32)
V = array([ 60.63256172,  66.03897428,  71.8062328 ,  77.09295576,
        59.81945165,  65.18457679,  77.83151901,  84.14149059,
        58.0955782 ,   7.98862928,  84.86730581,  91.78165089,
        69.4968138 ,  76.80991653,  91.78165089, 100.        ,
         0.        ])
nIterations = 5
```

3. *Report the number of iterations needed by modified policy iteration to converge when varying the number of iterations in partial policy evaluation from 1 to 10. Use a tolerance of 0.01, start with the policy that chooses action 0 in all states and start with the value function that assigns 0 to all states.*

> We get:



4. *Discuss the impact of the number of iterations in partial policy evaluation on the results and relate the results to value iteration and policy iteration.*

> As we can see in the graph above, as the number of iterations required for convergence go down and eventually plateau as the evaluation runs for longer. This intuitively makes sense as it implies that policy improvement is based on more confident value estimates (gained through Richardson iterations) and thus more informed changes to the policy are possible, which makes it converge to optimality faster. This is thus a hybrid version of policy and value iteration, and involves the best of both techniques – direct policy optimization with fast value estimates.