Code Documentation

Q2. For the gridworld following 5 cases are considered -

a) Corner Cell
The equation in this case would be -

$$V_{pi}(s) = (gamma/4)*\Sigma_n V_{pi}(n) + 2*(1/4)*(-1 + gamma*V_{pi}(s))$$

where n represents valid neighbours of cells.

b) Edge Cell
The equation in this case would be -

$$V_{pi}(s) = (gamma/4)*\Sigma_n V_{pi}(n) + (1/4)*(-1 + gamma*V_{pi}(s))$$

c) Other Cell
The equation in this case would be -

$$V_{pi}(s) = (gamma/4)*\Sigma_n V_{pi}(n)$$

d) Cell A
The equation in this case would be -

$$V_{pi}(A) = 10 + gamma*V_{pi}(A')$$

e) Cell B
The equation in this case would be -

$$V_{pi}(B) = 5 + gamma*V_{pi}(B')$$

The coefficient matrix A and constant term matrix B are formed using the above cases.
Then V_{pi}'s are found using
$X = A^{-1}B$,
where X is vector of V_{pi}'s.

Hence we get -

```
X = array([[ 3.3,  8.8,  4.4,  5.3,  1.5],
           [ 1.5,  3. ,  2.3,  1.9,  0.5],
           [ 0.1,  0.7,  0.7,  0.4, -0.4],
           [-1. , -0.4, -0.4, -0.6, -1.2],
           [-1.9, -1.3, -1.2, -1.4, -2. ]])
```

Q4. In this case, the optimal policy is defined as -

$$v^*(s) = \max_{action} r + gamma*v^*(s')$$

where $s \rightarrow (action) \rightarrow s'$

Hence, instead of max we take inequality because $v^*(s)$ is greater than or equal to

r + gamma*$v^*(s')$ for any of the four actions.

Now Ax>=B is solved with objective function ΣX_i.

On solving we get -

V* = [[22. 24.4 22. 19.4 17.5]
[19.8 22. 19.8 17.8 16.]
[17.8 19.8 17.8 16. 14.4]
[16. 17.8 16. 14.4 13.]
[14.4 16. 14.4 13. 11.7]]

Pi*

(0, 0): right |
(0, 1): up | down | left | right
(0, 2): left |
(0, 3): up | down | left | right
(0, 4): left |
(1, 0): up |right |
(1, 1): up |
(1, 2): up |left |
(1, 3): left |
(1, 4): left |
(2, 0): up |right |
(2, 1): up |
(2, 2): up |left |
(2, 3): up |left |
(2, 4): up |left |
(3, 0): up |right |
(3, 1): up |
(3, 2): up |left |
(3, 3): up |left |
(3, 4): up |left |
(4, 0): up |right |
(4, 1): up |
(4, 2): up |left |
(4, 3): up |left |
(4, 4): up |left |

Q6. With both policy iteration and value iteration the following optimal policy and optimal state values are obtained.

Optimal Policy is -

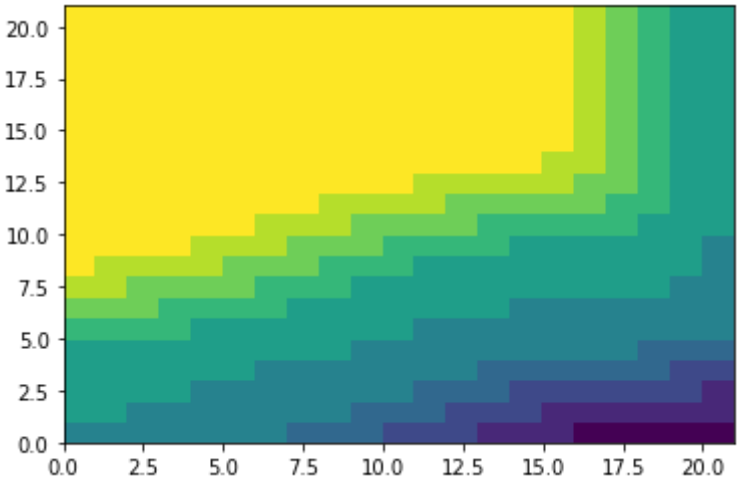
pi* = [['T' 'l' 'l' 'dl']
['u' 'ul' 'udlr' 'd']
['u' 'udlr' 'dr' 'd']
['ur' 'r' 'r' 'T']]

Optimal value function

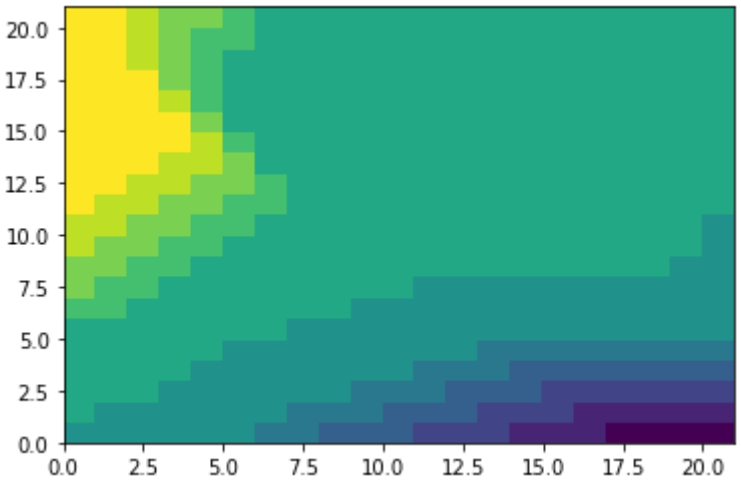
v* = [[0. -1. -2. -3.]
[-1. -2. -3. -2.]
[-2. -3. -2. -1.]
[-3. -2. -1. 0.]]

Q7. The plots of successive policies learnt are -

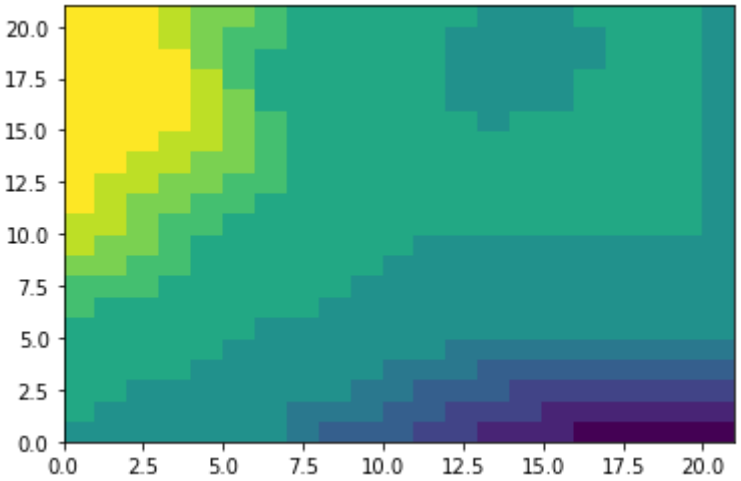
a.



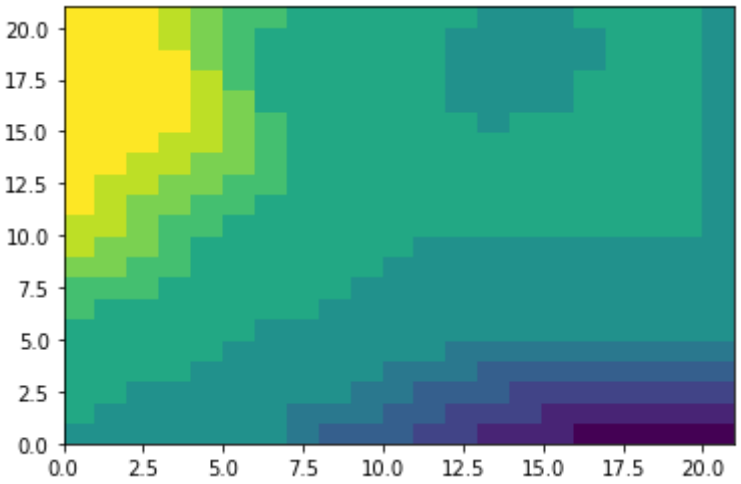
b.



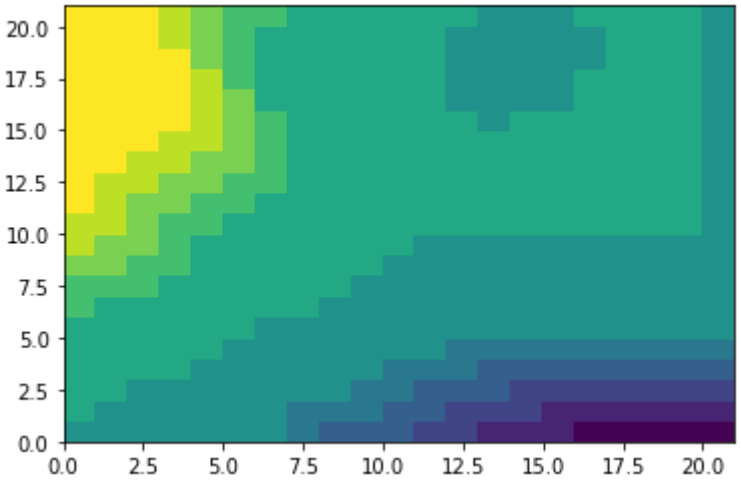
c.



d.



e. Optimal Policy



The plot of state value function for the optimal policy is given by