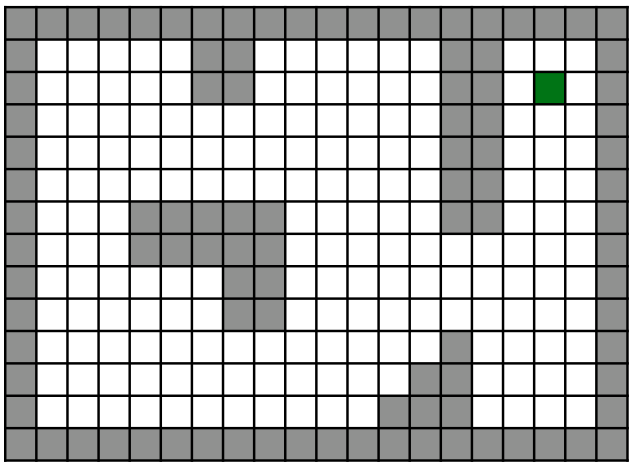
Autonomous Robots
Robot Learning – Reinforcement Learning

This document will guide you through the practical work related with the Reinforcement Learning subject of the Autonomous Robots course. The goal of this practical exercise is to implement a Reinforcement Learning algorithm to learn a policy that moves a robot to a goal position. The algorithm is the Q-learning algorithm. You will program all the code in Matlab.

1. The reinforcement learning problem

The problem consists in finding the goal in a finite 2D environment that is closed and contains some obstacles (in grey) as shown in the figure:



States and actions: The size of the environment is 20x14=280 states. The robot can only do 4 different actions: {←, ↑, →, ↓} (not diagonal movements!). Therefore the size of the Q function will be 280x4=1120 cells.

Dynamics: The robot can be located in any free cell (not in the obstacle cells!). The function that describes the dynamics is very simple: the robot will move ONE cell per iteration to the direction of the action that we select, unless there is an obstacle or the wall in front of it, in which case it will stay in the same position.

Reinforcement function: Since the goal is to reach the goal position as fast as possible, the reinforcement function will give -1 in all cells except in the goal cell, where the reward will be +1. The cell that contains the goal is (18,3).

2. The algorithm

You must implement the Q-learning algorithm. Consider the problem as episodic and repeat the learning process during hundreds of episodes. The algorithm should be like:

```
Initialize Q(s,a) to "0"
For n episodes
    Initialize s randomly in any free cell
    For m iterations repeat
        Choose a following ε-greedy policy
        Take action a, observe r, s'
        Q(s,a)←Q(s,a) + α ( r + γ · maxQ_a'(s',a')-Q(s,a))
        s←s'
        if the goal is achieved then finish the episode
    endFor
endFor
```

You will have to set several parameters experimentally: n, m, ε, α and γ.

3. Testing the learnt policy

In order to test the learnt policy, after some episodes you can test the policy. Just run an episode with $\varepsilon = \alpha = 0$, and sum the rewards. Since the initial position is randomly selected, run several episodes and compute the average of the accumulated rewards. This average should decrease with respect to the number of episodes in which the learning was enabled.

You can also plot the greedy policy by checking for each of the 280 cells the best action. Then you can determine if the best action is the one you would choose.

4. Implementation

You must program a Matlab function that has the following definition. It is very important that the input and output arguments have the following format.

```
Q=q_learning(map,q_goal,alpha,gamma,epsilon,n_episodes,n_iterations)
```

where:

map: matrix that represents an occupancy grid map

q_goal: X and Y coordinates of the cell that contains the goal state.

alpha: learning rate of the algorithm

gamma: discount factor of the algorithm

epsilon: random action probability

n_episodes: number of episode repetitions

n_iterations: number of iterations per episode

Q: 3D matrix of size: number of columns; number of rows; number of actions. The matrix contains the Action Value Function.

When your function is called with the following arguments,

```
map=[
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1;
1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1;
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1;
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1;
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1;
1 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 1;
1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1;
1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1;
1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1;
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1;
1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1;
1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1;
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;];

Q=q_learning(map,[18,3],0.1,0.9,0.3,20000,50)
```

your function MUST show the following numerical and GRAPHICAL info:

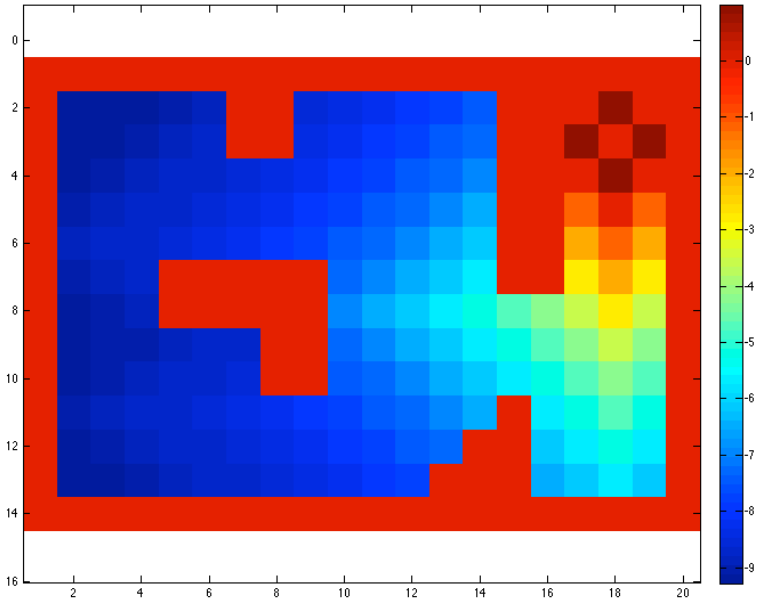
a) Q function:

```
Q(:,:,1) =
         0         0         0         0         0         0         0         0         0         0         0         0         0
         0   -9.3528   -9.2857   -9.2101   -9.1220   -9.0249   -9.1140   -9.1982   -9.0700   -9.0250   -8.9154   -8.8548   -8.8109         0
         0   -9.3576   -9.2889   -9.2103   -9.1226   -9.0251   -9.1219   -9.2021   -9.0116   -8.9343   -8.8715   -8.8076   -8.7616         0
         0   -9.2886   -9.2103   -9.1226   -9.0251   -8.9168   -9.0251   -9.1201   -8.9493   -8.8841   -8.7690   -8.7458   -8.6869         0
         0   -9.2091   -9.1226   -9.0251   -8.9168   -8.7964         0         0   -8.8528   -8.7708   -8.7499   -8.6390   -8.6137         0
         0   -9.1225   -9.0251   -8.9168   -8.7964   -8.6627         0         0   -8.7209   -8.6498   -8.5762   -8.5154   -8.5631         0
         0         0         0   -8.7964   -8.6627   -8.5141         0         0   -8.6463   -8.6070   -8.5127   -8.4253   -8.3842         0
         0         0         0   -8.6627   -8.5141   -8.3490         0         0         0         0   -8.4205   -8.2733   -8.2844         0
         0   -8.6488   -8.5138   -8.5141   -8.3490   -8.1655         0         0         0         0   -8.2317   -8.1329   -8.1352         0
         0   -8.6502   -8.5140   -8.3490   -8.1655   -7.9617   -7.4835   -7.2039   -7.3434   -7.5073   -7.9638   -8.0035   -8.0193         0
         0   -8.5137   -8.3490   -8.1655   -7.9617   -7.7352   -7.4836   -7.2039   -7.4055   -7.5274   -7.8446   -7.7384   -7.8234         0
         0   -8.3489   -8.1655   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -7.1952   -7.2717   -7.4600   -7.6141   -7.6533         0
         0   -8.1653   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.8927   -7.1452   -7.2497   -7.3445         0         0
         0   -7.9596   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.1645   -6.5481   -6.8884   -6.9592         0         0         0
         0         0         0         0         0         0         0   -5.7384   -6.1645   -6.5401         0         0         0         0
         0         0         0         0         0         0         0   -5.2649   -5.7384   -6.1559   -5.3643   -5.1496   -5.0409         0
         0   -0.4292   -0.1022   -1.0900   -1.9810   -2.7829   -3.5046   -4.7387   -5.2649   -5.7287   -5.2207   -5.0848   -5.0428         0
         0   -0.1022         0   -1.0900   -1.9810   -2.7829   -3.5046   -4.1541   -4.7382   -5.1380   -5.0205   -4.9299   -4.9081         0
         0   -0.1000    1.0000   -0.1000   -1.0900   -1.9810   -2.7829   -3.5046   -4.1511   -4.6094   -4.7428   -4.8315   -4.8774         0
         0         0         0         0         0         0         0         0         0         0         0         0         0         0
Q(:,:,2) =
         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0   -9.3572   -9.3545   -9.2878   -9.2101   -9.1225   -9.0251   -9.1226   -9.0929   -9.0163   -8.9421   -8.8693   -8.8107         0
         0   -9.2881   -9.2887   -9.2103   -9.1226   -9.0251   -8.9168   -9.0251   -9.0216   -8.9416   -8.8914   -8.7941   -8.7677         0
         0   -9.2100   -9.2103   -9.1226   -9.0251   -8.9168   -8.7964   -8.9168   -8.9476   -8.8680   -8.7764   -8.7583   -8.7197         0
         0   -9.1224   -9.1226   -9.0251   -8.9168   -8.7964         0         0   -8.8410   -8.7934   -8.6638   -8.6211   -8.6204         0
         0   -9.0250   -9.0251   -8.9168   -8.7964   -8.6627         0         0   -8.7559   -8.6480   -8.5924   -8.5146   -8.5043         0
         0         0         0   -8.6627   -8.6627   -8.5141         0         0   -8.6323   -8.5527   -8.5633   -8.4015   -8.3910         0
         0         0         0   -8.5141   -8.5141   -8.3490         0         0         0         0   -8.2806   -8.2603   -8.2779         0
         0   -8.6481   -8.6623   -8.5141   -8.3490   -8.1655         0         0         0         0   -8.1108   -8.1033   -8.1387         0
         0   -8.5131   -8.5141   -8.3490   -8.1655   -7.9617   -7.7352   -7.4832   -7.2035   -7.4797   -7.7293   -7.9124   -7.9744         0
         0   -8.3487   -8.3490   -8.1655   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -7.2032   -7.4797   -7.7055   -7.8148         0
         0   -8.1654   -8.1655   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.8932   -7.2019   -7.4648   -7.6571         0
         0   -7.9614   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.1645   -6.5481   -6.8928   -7.1991         0         0
         0   -7.7351   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.1645   -5.7384   -6.1645   -6.5481         0         0         0
         0         0         0         0         0         0         0   -5.2649   -5.2649   -5.7384         0         0         0         0
         0         0         0         0         0         0         0   -4.7387   -4.7387   -5.2649   -5.4014   -5.2307   -5.0838         0
         0   -0.2817   -0.9744   -0.1000   -1.0900   -1.9810   -2.7829   -3.5046   -4.1541   -4.7387   -5.1635   -5.0535   -5.0120         0
         0         0         0    1.0000   -0.1000   -1.0900   -1.9810   -2.7829   -3.5046   -4.1541   -4.7304   -4.9144   -4.9196         0
         0   -0.3590   -0.8772   -0.1000   -1.0900   -1.9810   -2.7829   -3.5046   -4.1510   -4.6200   -4.6901   -4.8627   -4.9194         0
         0         0         0         0         0         0         0         0         0         0         0         0         0         0
Q(:,:,3) =
         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0   -9.2893   -9.2103   -9.1226   -9.0251   -8.9168   -9.0251   -9.1226   -9.0679   -8.9899   -8.9111   -8.8398   -8.8102         0
         0   -9.2103   -9.1226   -9.0251   -8.9168   -8.7964   -8.9168   -9.0251   -9.0048   -8.9293   -8.8536   -8.7955   -8.7643         0
         0   -9.1226   -9.0251   -8.9168   -8.7964   -8.6627   -8.9168   -9.0244   -8.9270   -8.8455   -8.7677   -8.7174   -8.6919         0
         0   -9.0251   -8.9168   -8.7964   -8.6627   -8.5141         0         0   -8.8273   -8.7443   -8.6471   -8.6208   -8.6087         0
         0   -9.0250   -8.9168   -8.6627   -8.5141   -8.3490         0         0   -8.7241   -8.6327   -8.5038   -8.5132   -8.5008         0
         0         0         0   -8.5141   -8.3490   -8.1655         0         0   -8.6523   -8.5358   -8.3403   -8.3945   -8.3871         0
         0         0         0   -8.3490   -8.1655   -7.9617         0         0         0         0   -8.1576   -8.2599   -8.2601         0
         0   -8.5141   -8.3490   -8.1655   -7.9617   -7.7352         0         0         0         0   -7.9549   -8.0988   -8.1235         0
         0   -8.3490   -8.1655   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -7.2034   -7.4799   -7.7294   -7.9128   -7.9669         0
         0   -8.1655   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.8933   -7.2032   -7.4796   -7.7014   -7.8114         0
         0   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.1645   -6.5481   -6.8932   -7.2022   -7.4654   -7.6690         0
         0   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.1645   -5.7384   -6.1645   -6.5481   -6.8928   -7.2379         0         0
         0   -7.7349   -7.4836   -7.2039   -6.8933   -6.5481   -6.1645   -5.2649   -5.7384   -6.1645   -6.7509         0         0         0
         0         0         0         0         0         0         0   -4.7387   -5.2649   -5.7384         0         0         0         0
         0         0         0         0         0         0         0   -4.1541   -4.7387   -5.2649   -5.3514   -5.1766   -5.1043         0
         0   -0.1000    1.0000   -0.1000   -1.0900   -1.9810   -2.7829   -3.5046   -4.1541   -4.7387   -5.1670   -5.0832   -5.0254         0
         0         0         0   -1.0900   -1.9810   -2.7829   -3.5046   -4.1541   -4.7214   -4.9848   -4.7700   -4.8877   -4.9230         0
         0   -0.1967   -0.1129   -1.0900   -1.9810   -2.7829   -3.4958   -4.1387   -4.3236   -4.6549   -4.7230   -4.8129   -4.9622         0
         0         0         0         0         0         0         0         0         0         0         0         0         0         0
Q(:,:,4) =
         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0   -9.2893   -9.2103   -9.1226   -9.0251   -9.1224   -9.1933   -9.1451   -9.0657   -8.9870   -8.9108   -8.8477   -8.8268         0
         0   -9.2103   -9.1226   -9.0251   -8.9168   -9.0251   -9.1225   -9.0810   -9.0055   -8.9327   -8.8537   -8.8000   -8.7682         0
         0   -9.1226   -9.0251   -8.9168   -8.7964   -8.9168   -9.0251   -9.0154   -8.9285   -8.8476   -8.7740   -8.7166   -8.6918         0
         0   -9.0251   -8.9168   -8.7964   -8.6627   -8.6627         0         0   -8.8311   -8.7466   -8.6623   -8.6346   -8.6111         0
         0   -8.9168   -8.7964   -8.6627   -8.5141   -8.5141         0         0   -8.7282   -8.6347   -8.5285   -8.5180   -8.5163         0
         0         0         0   -8.5141   -8.3490   -8.3490         0         0   -8.6263   -8.5015   -8.4539   -8.4006   -8.4039         0
         0         0         0   -8.3490   -8.1655   -8.1655         0         0         0         0   -8.3025   -8.2595   -8.2708         0
         0   -8.5141   -8.3490   -8.1655   -7.9617   -7.9617         0         0         0         0   -8.2027   -8.1241   -8.1313         0
         0   -8.3490   -8.1655   -7.9617   -7.7352   -7.4836   -7.2039   -7.4779   -7.5165   -7.7749   -7.9817   -7.9254   -7.9781         0
         0   -8.1655   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -7.2039   -7.3171   -7.5309   -7.6824   -7.7722   -7.8433         0
         0   -7.9617   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.8933   -7.1729   -7.2346   -7.4378   -7.5094   -7.6515         0
         0   -7.7352   -7.4836   -7.2039   -6.8933   -6.5481   -6.1645   -6.5481   -6.8923   -7.0974   -7.2453   -7.2997         0         0
         0   -7.4836   -7.2039   -6.8933   -6.5481   -6.1645   -5.7384   -6.1645   -6.5481   -6.8869   -6.7771         0         0         0
         0         0         0         0         0         0         0   -5.7384   -6.1645   -6.1547         0         0         0         0
         0         0         0         0         0         0         0   -5.2649   -5.7384   -5.6456   -5.3735   -5.1648   -5.0795         0
         0   -0.0919   -1.0031   -1.9810   -2.7829   -3.5046   -4.1541   -4.7387   -5.2649   -5.5410   -5.1781   -5.0560   -5.0192         0
         0         0         0   -1.0900   -1.9810   -2.7829   -3.5046   -4.1541   -4.7382   -5.0725   -4.7974   -4.8834   -4.8972         0
         0   -0.0817   -0.8656   -1.9810   -2.7829   -3.5042   -4.1443   -4.6406   -4.4476   -4.6091   -4.8099   -4.8900   -4.8947         0
         0         0         0         0         0         0         0         0         0         0         0         0         0         0
```

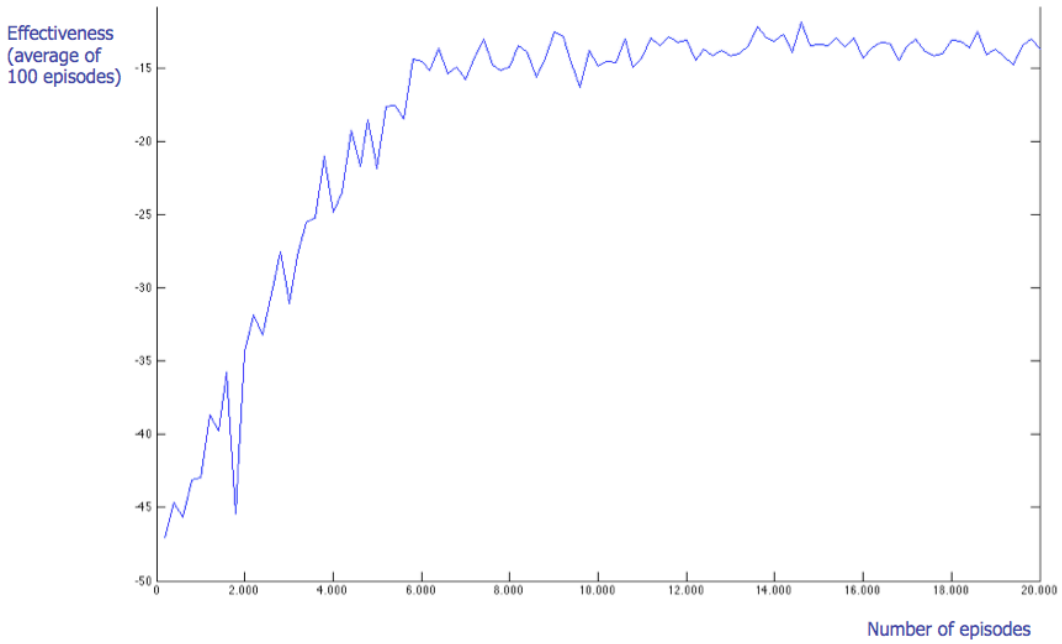
b) The optimal policy using some kind of representation such as:

```
o o o o o o o o o o o o o o o o o o o o o
o v > > > v o o > > > > > v o o v ^ v o
o > > > > v o o > > > > > v o o > G < o
o > > > > > > > > > > > > v o o > ^ < o
o > > > > > > > > > > > > v o o > ^ < o
o > > > > > > > > > > > > v o o ^ ^ ^ o
o ^ ^ ^ o o o o o > > > > v o o ^ ^ < o
o ^ ^ ^ o o o o o > > > > > > > ^ ^ ^ o
o v > > > < v o o > > ^ ^ ^ ^ ^ ^ ^ ^ o
o v > > > > v o o ^ > > > > ^ > ^ ^ v o
o v > > > > > > > ^ > ^ ^ ^ o > ^ ^ ^ o
o > ^ v > > > v > ^ > ^ ^ o o < ^ v > o
o > < < > > < > > > > v o o o < ^ v < o
o o o o o o o o o o o o o o o o o o o o o
```

c) A graphical representation of the State Value Function V such as:



d) A graphical representation of the evolution of the effectiveness such as:



5. Submission.

WORK TO DO: Submit a report in pdf and the Matlab file “q_learning.m”. Explain in detail with graphical information, in the report, the work done. Explain also the problems you found.

NOTE that your function “q_learning.m” will be tested with other environments and other parameters. Remember that your function MUST return the Q function and SHOW GRAPHICALLY the optimal policy, the state value function and the effectiveness evolution.