Exercise 1.

(a) Give a definition for a complete planning algorithm.

An algorithm is said to be complete if it computes a path to the goal if it exists in finite amount of time or returns failure if the path does not exist.

(b) Give a definition for an optimal planning algorithm.

An optimal planning algorithm is the one which produces a path that minimizes the Cost function (generally dependent on time taken, length of the path and computational cost) or maximizes the Reward function among all the feasible paths.

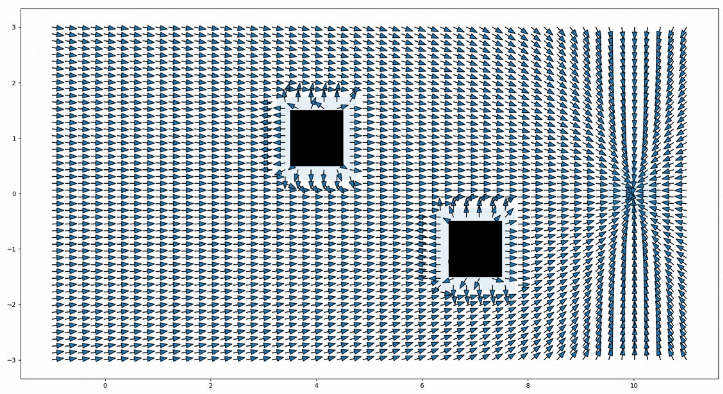
(c) Recall the wave-front planner from Lecture 8. Is it a complete planner? Is it an optimal planner?

Wave-front planner is a complete with respect to the defined grid. It will find a path from start grid to goal grid if it exists in finite time. Also if the path does not exist with respect to the grid it will search the entire free area in the workspace and exit with failure.
Wave-front planner is an optimal planner in a certain sense. The reason being it will always choose the path with least distance (L1 norm if the neighbors are the ones who share a facet or diagonal if neighbors also share a vertex) between the start and goal. It is only optimal in a discrete world, in continuous it can be semi-optimal.

Exercise 2.

(a)

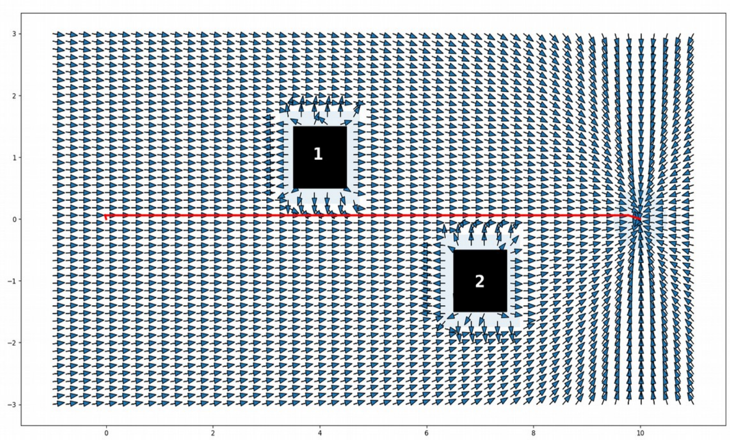
i. Vector field



ii.
d*=1
Q*=[0.25,0.25]

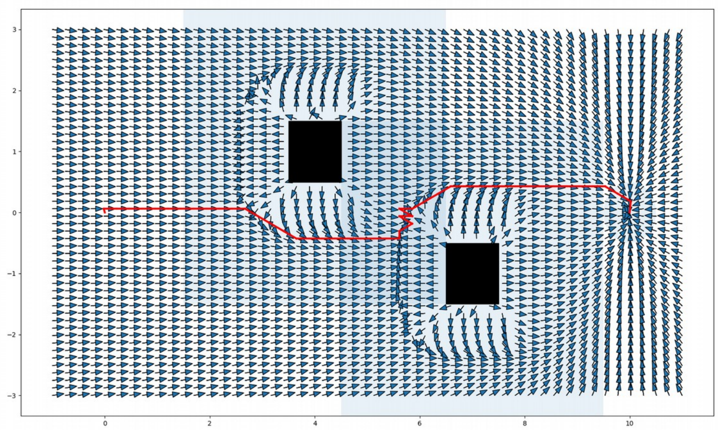
Here the obstacles do not obstruct the line of vision between start and goal hence with any quantity of d* the robot is sure to converge in the field. The Q* are chosen such that if by mistake the robot attempts to go towards the obstacles it will get repelled before going too close to it.

iii. Path



iv. Length of the path is 10.093 units.

v. No, as the potential function will change because of the change in d* and Q* and it will change the path taken by the robot.



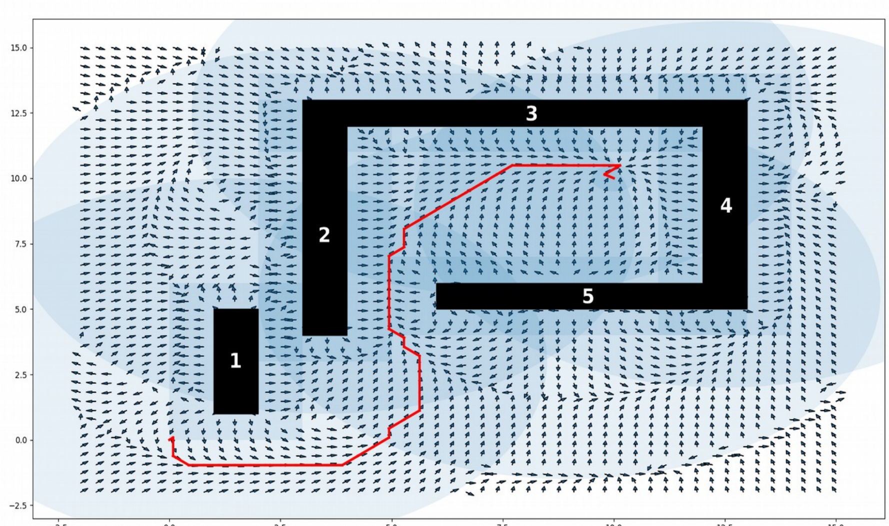
Length of the path is 11.739 units for Q*=[2,2]

(b) Workspace 1

i. d*=1
Q*=[1,1,1,1,1]
centroid radius=[7,7,7,7,7]

Here I used one extra repulsive potential function which repelled the robot from the centroid of each obstacle. It became easier to to go around the obstacle 1 without getting stuck in the minima near it. The Q* was still used to keep the robot away from the obstacles by a certain amount. 1 unit was chosen for all the obstacles due to the gap between obstacles 2 and 5 is 2 unit. This eliminated the jitters and found a straight path between them. The centroid radius was chosen such that the robot will travel away from the obstacle 5 to avoid getting stuck in the local minima below it.

ii.



iii. Length of the path is 21.542 units

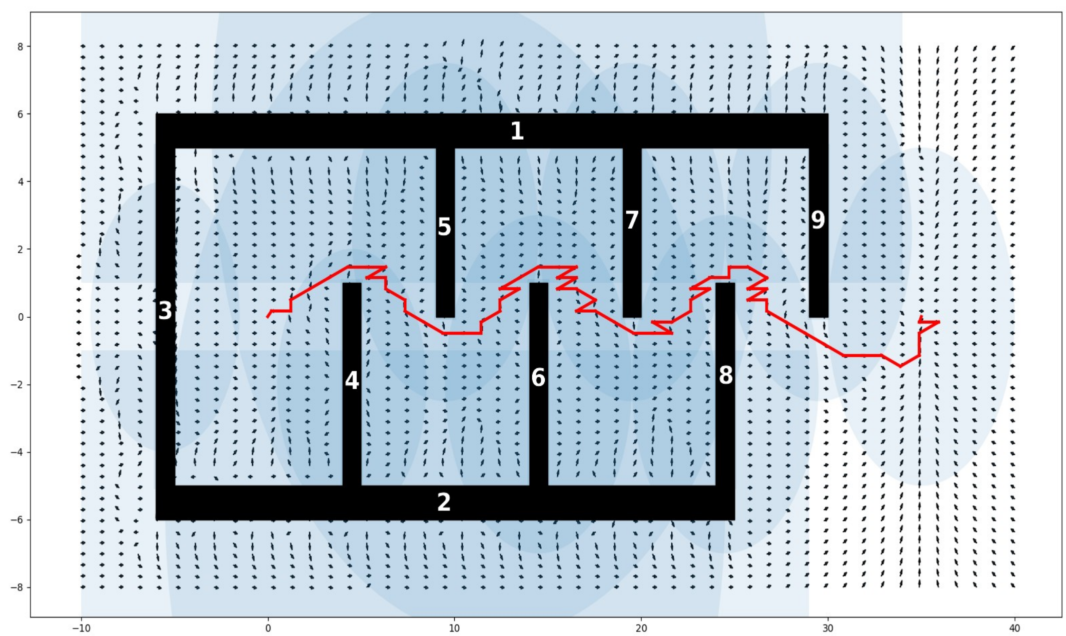
iv. No, as the potential function will change because of the change in d* and Q* and it will change the path taken by the robot.

(b) Workspace 2

i. d*=10
Q*=[4,4,0.1,0.1,0.1,0.1,0.1,0.1,0.1]
centroid radius=[15,15,4,4,5,5,5,5,5]

Here I used one extra repulsive potential function which repelled the robot from the centroid of each obstacle. The Q* and centroid radius for obstacles 1 and 2 are chosen such that it will repel the robot away in a zig-zag fashion between the other obstacles. This allows the robot to move forward without getting stuck in any local minima (in front of the obstacles 4 to 9). The centroid radius of obstacles 5 to 9 is chosen as 5 so that the robot will go around each obstacle keeping distance from it.
D* is just used to speed up the convergence of the robot after it gets out of the maze.

ii.

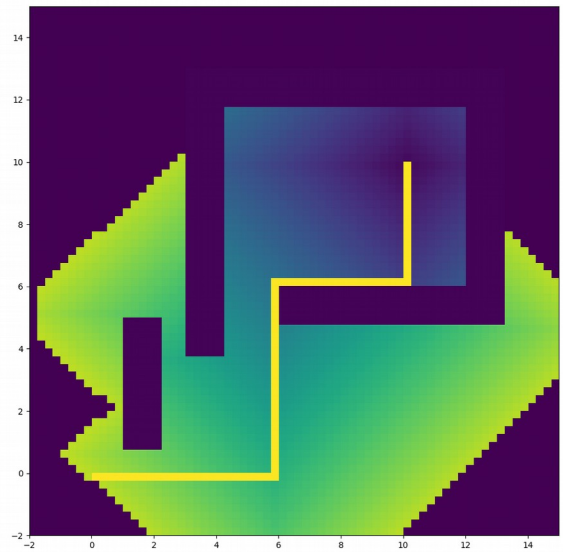


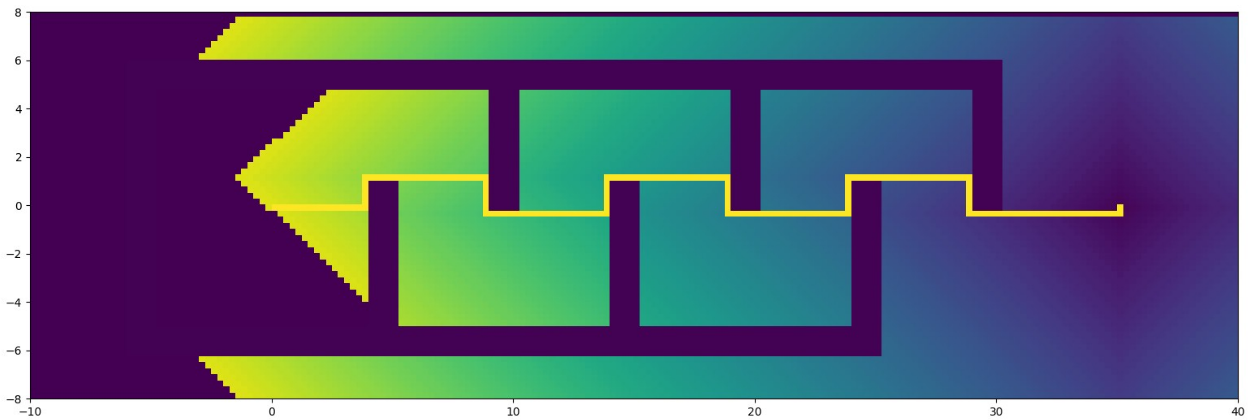
iii. Length of the path is 60.401 units

iv. No, as the potential function will change because of the change in d* and Q* and it will change the path taken by the robot.

Exercise 3.

(a) Plot the paths generated by the planner. (Assume the robot transverses the adjacent cells using a line that connects their centers.)





(b) What are the lengths of the paths?

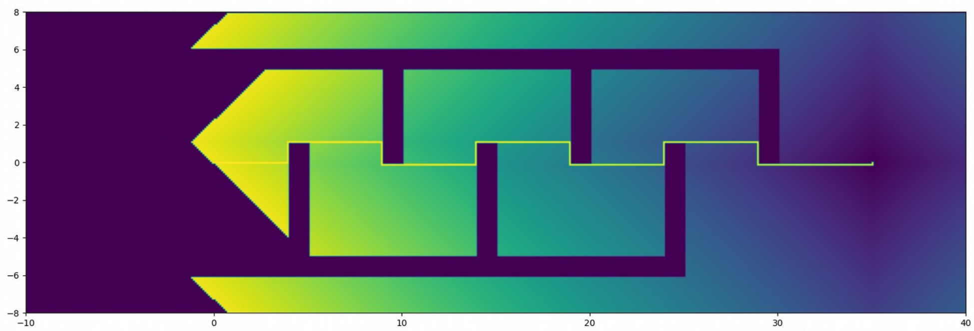
Workspace 1: 20 units

Workspace 2: 44 units

(c) Would you expect the path lengths to get smaller as the grid size gets smaller?

No, as the size of the grid gets smaller the distance between the two grid reduces but the number of grids on the path increases by the same margin. Wavefront planner is an optimal planning algorithm,

hence it gives the same (optimal) path length. This is true only for the L1 norm case (neighbors only share a facet).
In some cases where the path is obstructed head-on by an obstacle the length of the path reduces by the amount it can go closer to the obstacle (Workspace 2) if the grid size changes.



Length of the path is 42.2 with grid size 0.1 units.

(d) How does this wave-front planner perform in comparison to the gradient descent planner in Exercise 2 part (b)?

Workspace 1

Algorithm	Time taken	Length of the path
Wave-front planner	7.242 secs	20 units
Gradient descent planner	16.241 secs	21.542

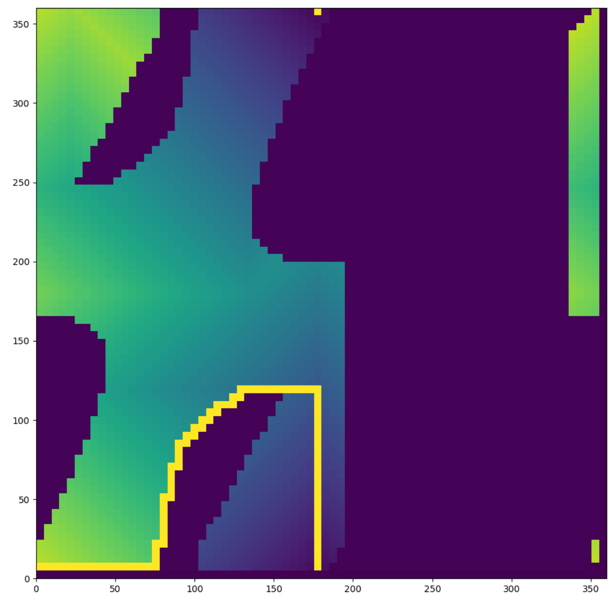
Workspace 2

Algorithm	Time taken	Length of the path
Wave-front planner	16.720 secs	44 units
Gradient descent planner	29.948 secs	60.401 units

Wavefront planner is doing better in both aspects namely Length of the path and Computational time. But it is only limited to the grid space and not the continuous space.

Exercise 4.

Path in Cspace:



Snapshots of manipulators: