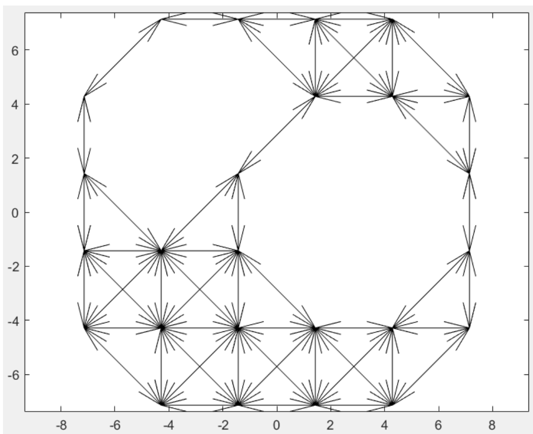
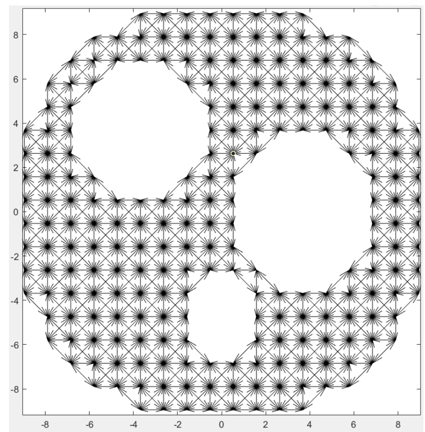
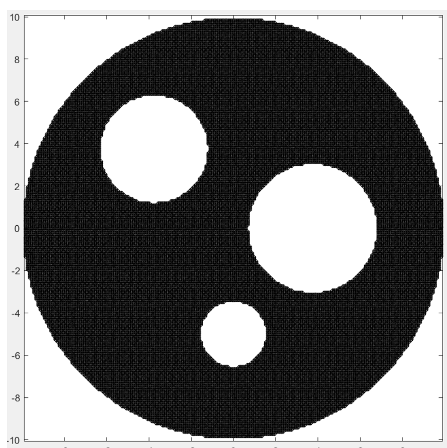
Zachary Weiss
ME570 HW4
Professor Tron
17 November 2020

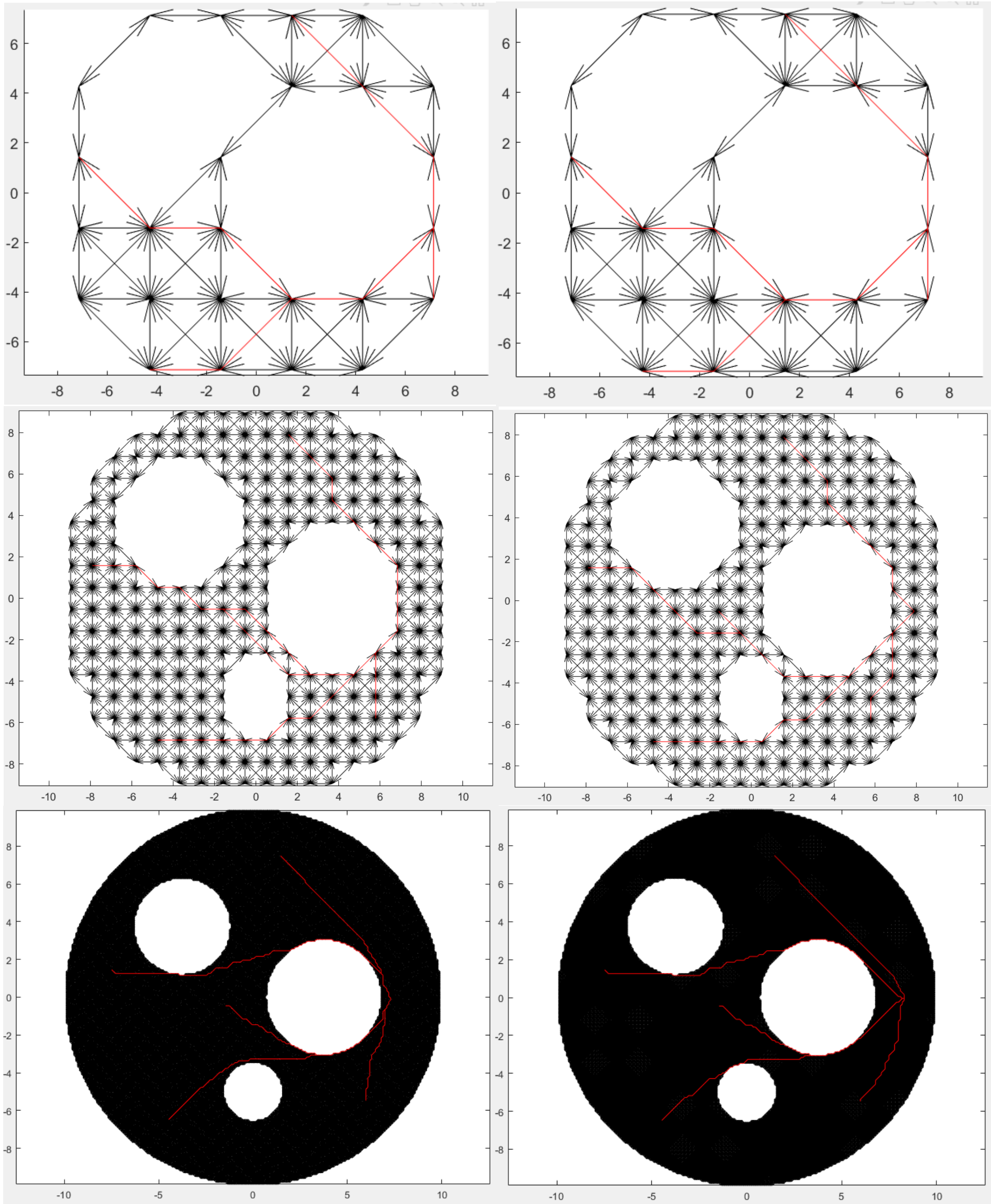
Q2.1:
Discretization was too course, well-tuned, and too fine, at an NCells of 8, 20, and 200 respectively. Their graphs are as follow:







Q2.2:

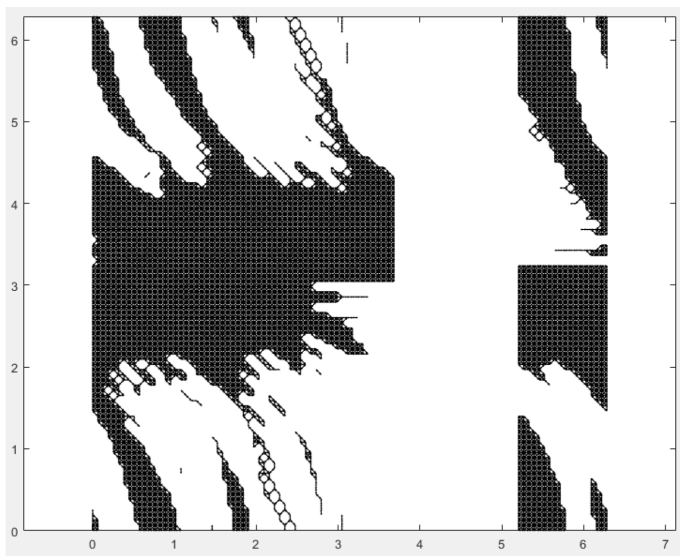


Q2.3:

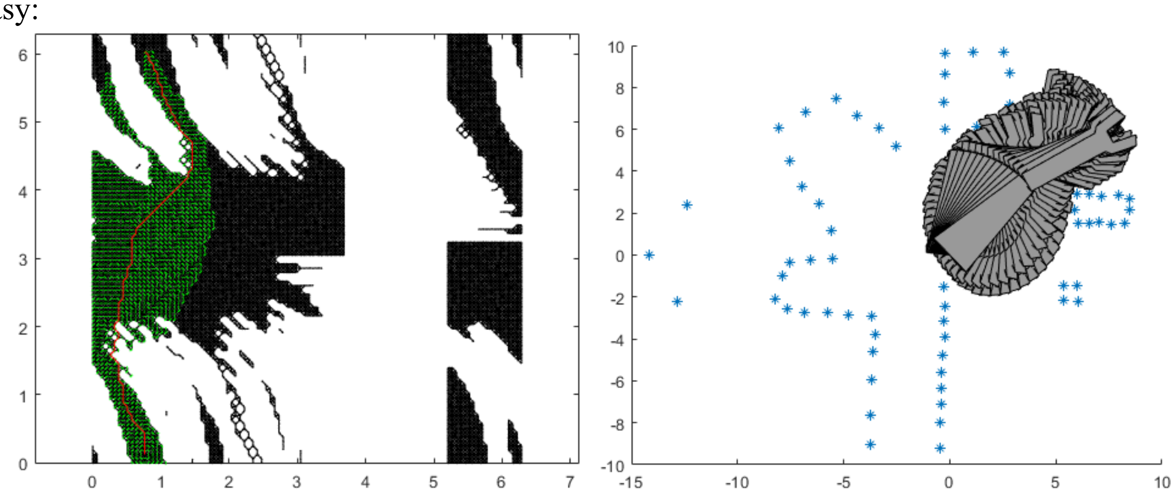
In instances with too few cells, there isn't sufficient granularity to distinguish the goals, and as such both plots appear the same. In the 'just right' number of cells, starts and goals are sufficiently distinguished, and paths are refined to slightly more optimal / shorter routes. In the overly-fine cell count, paths are smoother (and one path finds a shorter route going above the largest obstacle rather than below, indicating the 'just right' cell count could possibly be slightly higher for more optimal planning), but computation takes much longer.

Q2.4:
A* finds similar paths to the potential planner, but due to discretization appears more rough / unnatural. It additionally has different behavior by obstacles, approaching as close as the grid granularity allows, whereas the potential planner keeps a distance away, proportional to the repulsive weight set. With A*, there is no practical considerations beyond perhaps cell count between the two goal locations, whereas with the potential planner one must ensure the repulsive weight and potential shape are conducive to global convergence to the goal (the one within the radius of influence of an obstacle may become unreachable if one is not careful, and similarly, one must worry about local minima that are not the goal).

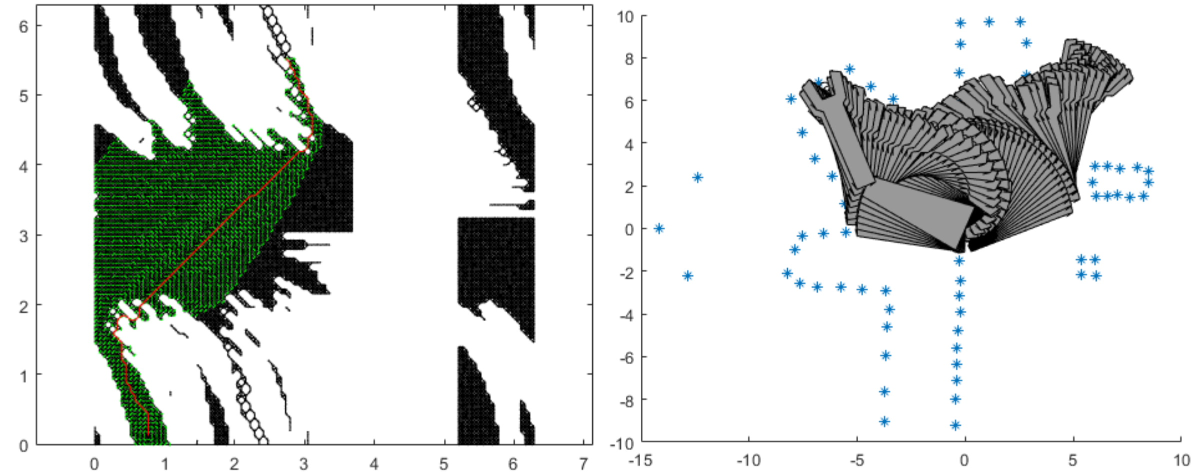
Q3.1:
The freespace of the twolink is represented below (bounds are from zero to 2pi along both axes, maps to torus).



Q3.2:
Easy:



Medium:



Q3.3:
If the configuration space had been mapped via a torus, the planner would have found the trivial (by inspection) route of simply rotating slightly clockwise, rather than the complex anticlockwise path it took. As it was not considered on a torus, but rather a plane (with 2pi and zero not corresponding to the same value / wrapping), it had to take the longer path pictured within the configuration space graph above.

Q3.4:
The planner approaches obstacles as closely as the discretization of the environment allows. Practically, this could lead to challenges where, when collisions or features would occur on a granularity finer than the current discretization, while the discretized motion appears to be entirely valid, the movement between discretized points is illegal. This appears to happen in both the easy and medium cases above; at multiple points in both paths, one of the obstacle points is centered within the end effector of the twolink, and in the next move, the effector has rotated further, past the obstacle point. As the discretization step is large enough that by the next point, the twolink is entirely clear of the obstacle, it fails to realize the move is illegal, and that when smoothly moving between the two points, collision will occur. Ways to fix this could include padding all objects within an environment such that grid points are not allowed within a certain safety-margin distance from obstacles, or alternatively, some form of adaptive grid sizing based on proximity to obstacles (course grid far from obstacles, finer near obstacles).

Q4.1:
This homework took less time than HW3, possibly around 10hrs total, in part due to some former familiarity with dynamic programming and A*/Dijkstra-style algorithms. It seemed to build well upon the earlier concepts (and code); the included testing data and autograder made debugging much easier than it would have been otherwise.