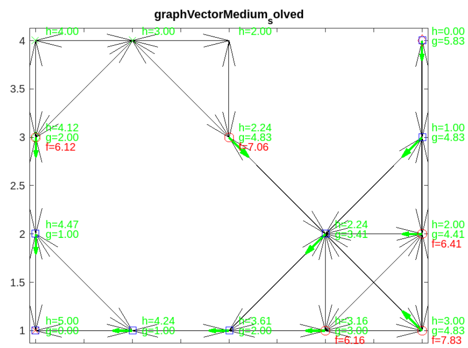
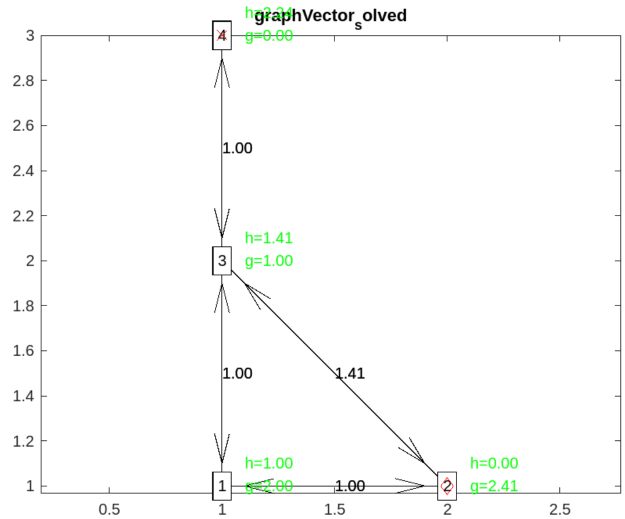
Homework 4
MATLAB
Report

Question 1.1:





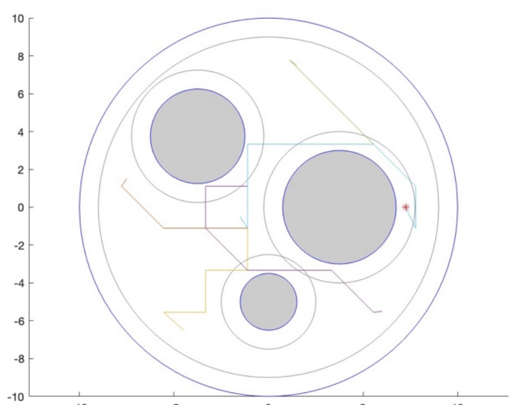
Question 1.2:
Question 2.1:

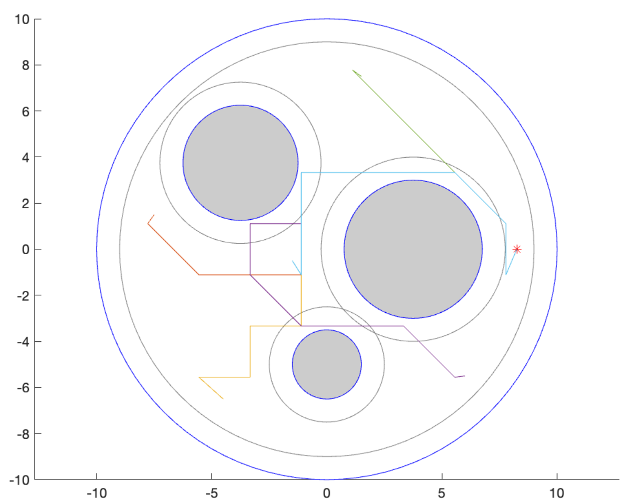
Some or all of the obstacles fuse together (NCells is too low) => N= 10

The topology of the Sphere World is well captured (NCells is "just right") =>N= 20

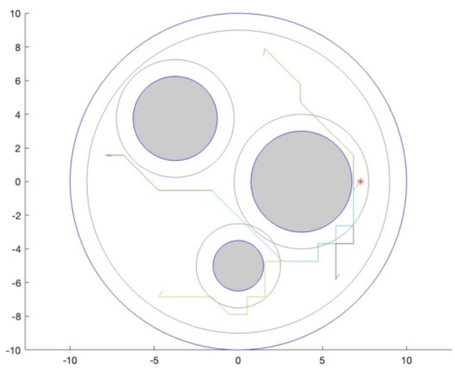
The graph is much finer than necessary (NCells is too high) =>N= 100

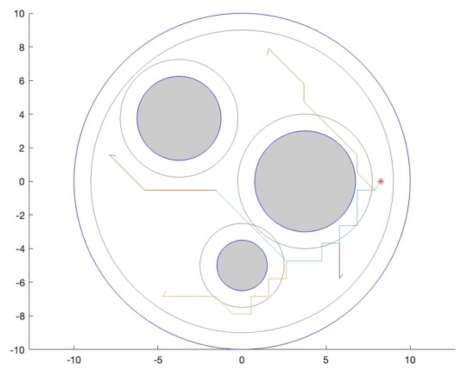
Question 2.2:
N= 10



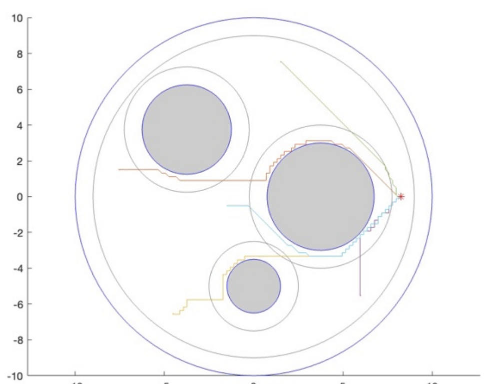


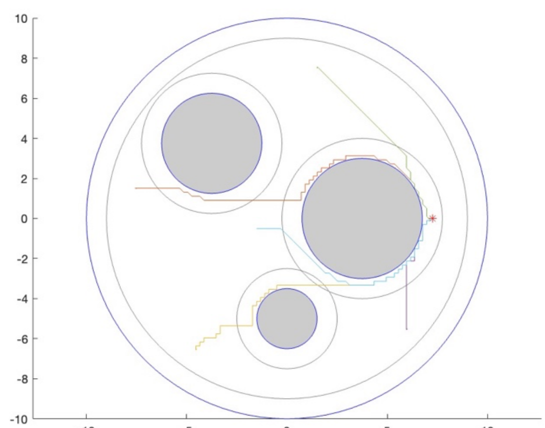
N= 20





N= 100





Question 2.3:
The planner's behavior depends on Ncell, which sets the search space's granularity, in route planning using the A* algorithm. Let's explore how various Ncell values affect the A* planner's performance using your submitted photos.

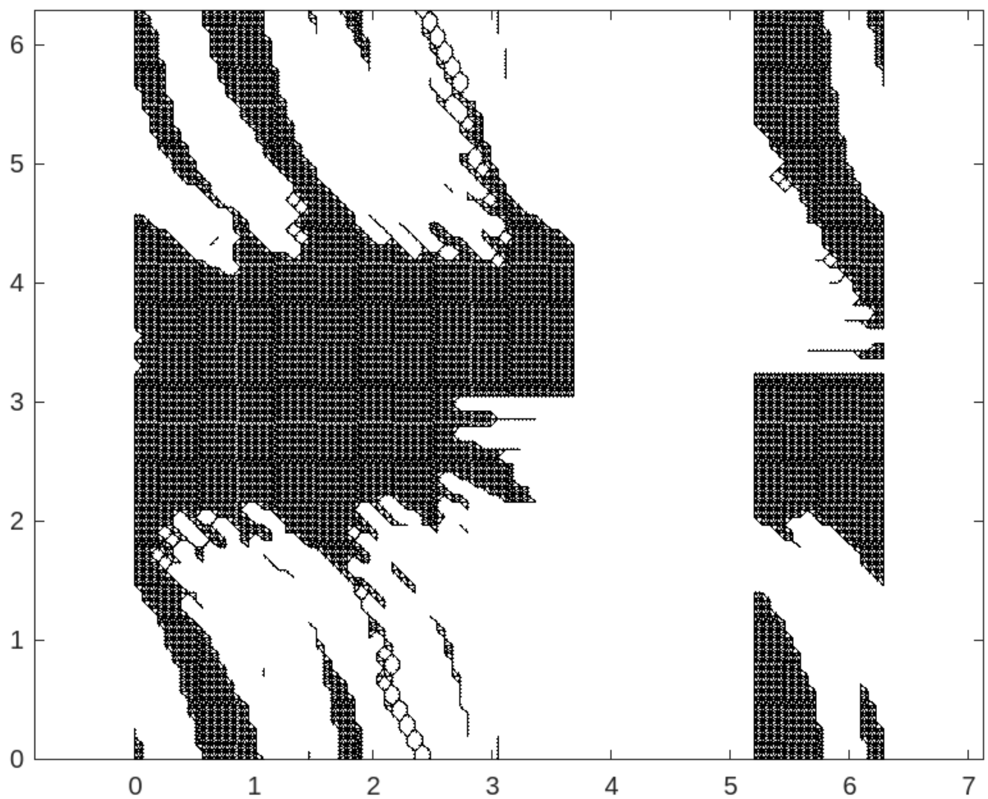
Ncell=10: Small Ncell values result in coarse spatial discretization. This makes each grid cell huge, which might cause problems:
The planner may misinterpret the geometry and free space around obstacles, resulting in less optimum or infeasible courses in a finer area.
The A* technique may locate a route rapidly owing to the reduced number of nodes to analyze, but it may not be the best path in continuous space.
If the robot's footprint is not well-represented by the big grid cells, obstacle clearance may not be effectively calculated, increasing the danger of collision.

Ncell=20: The A* planner can better mimic the continuous space and its obstacles with this finer discretization than Ncell=10.
The planner will likely locate a route closer to the best path while staying away from obstructions.
Since the robot's footprint and obstructions are better represented at higher resolution, the produced pathways are safer for navigation.
Because there are more cells to examine, the computational cost will be greater than with Ncell=10, but route quality and safety may be worth it.
Ncell=100: The discretization is quite fine, maybe too fine for practical use.
The created pathways may avoid obstacles accurately and be near to the continuous space optimum path.
The A* method may be slower in bigger or more complicated situations owing to the vast number of cells, which increases computing cost.
Given the processing burden, a high resolution may not be much better than Ncell=20.
In the photos, if pathways are near to barriers for larger Ncell values, the A* planner may deem them safe given the resolution. However, actual robotics applications must consider the robot's size and maintain a safety buffer around obstacles. During planning, inflating obstructions or post-processing the route might boost clearance.

Choose Ncell that balances computational efficiency with safe navigation detail. Too coarse a discretization risks missing feasible courses and safe navigation, while too fine might cause excessively high computational costs without practical navigation advantages.

Question 2.4:

Question3.1:



Question 3.2:

Question 3.3:

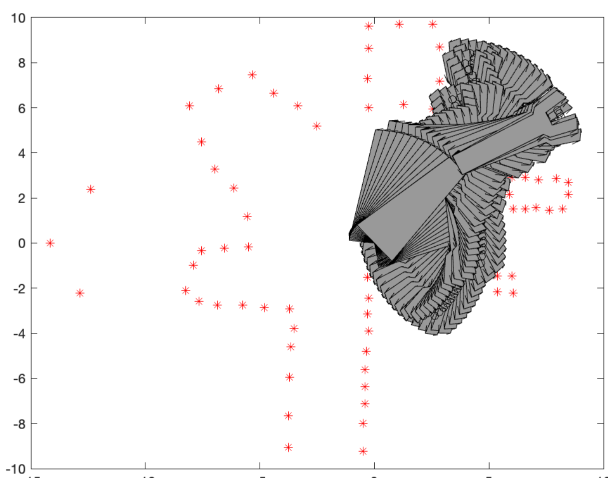


Figure 1: Easy

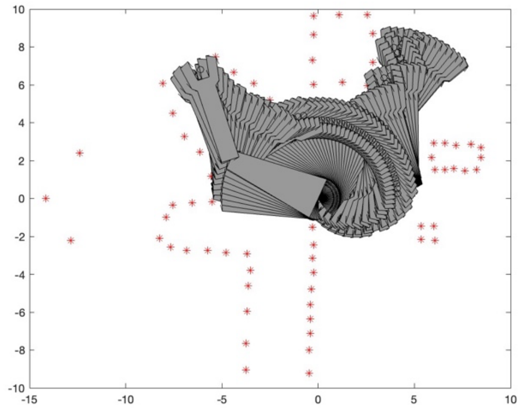


Figure 2: Medium

Question 3.4:
The "torus" choice shows the angles of the joints in a way that takes into account the fact that they wrap around. For example, 0 degrees is the same as 360 degrees, and the space is seamless. Angles between 0 degrees and 360 degrees are treated as separate points if they are not shown on a torus. The space is also treated as broken up.

When the motion planner doesn't realize that a link is in the same place at 0 degrees after it has turned 360 degrees, this "unwinding" happens. So, the planner might make a path that turns a link around multiple times when it doesn't need to and could find a faster path by noticing the angles wrapping around.

The planner doesn't understand how the robotic arm's rotating motion stays the same if you don't use the "torus" choice. Because it doesn't "see" that the link could have stayed in place after a full turn, it might not find the simple path that keeps the first link in place while moving the second link to reach the goal. This can make routes too complicated, where the first link might have to turn several times to get to a spot that could be reached without turning at all.

When you choose the "torus" choice, the 0 degrees and 360 degrees points are joined together to make a continuous place that better shows how the robotic arm's joints actually move. When the "torus" option is turned on, the motion planner can figure out that the quickest way to get to the goal might be to leave the first link alone and only move the second one.

Question 3.5:
The planner's objective is to find a route from the start to the goal without hitting barriers. According to the images , the planner is seeking ways near the barriers. In practical applications where safety margins are needed to accommodate for uncertainties like:

Sensing mistakes: Sensing equipment flaws may obscure obstacle location and form.
Control faults: Actuators or control system problems may cause the robot to deviate from its course.
Dynamic effects: Inertia may lead the robot to wander from its route at greater speeds.
Mistakes in models: The planner's mathematical model may not accurately represent robot and environment physics.
Safe operation sometimes requires safety margins or buffer zones around obstructions. You can do this:

Inflating obstacles: To guarantee safety, size them up while design.
A minimal distance from obstacles may be maintained by adding limitations to the planning algorithm.
After designing, smoothing or changing pathways to enhance obstacle distance.
Dynamic replanning: Real-time plan updates enable robots to avoid obstacles.
Method selection relies on application needs and robot and sensor system capabilities. Safety and practicality must be balanced since too wide a safety margin might render certain approaches infeasible.