Jay Lanzafane

RMP HW #3

Problem 1.)

* 1. This function just uses a quick conversion from polar coordinates to (x,y) in order to quickly trace out the outer edge of each sphere. Note that the function works for a single sphere or n spheres.
  2. These are 2 quick functions. The bulk of the work was done in sphere\_plot and, for sphereworld\_plot, we simply call sphere\_plot once for the obstacles and once for the sphere of influence around each of those obstacles. After that we mark the goal location with a simple plot command. Sphereworld\_plot\_test is even simpler, we load the provided sphereworld.mat file and then call sphereworld\_plot with the (now loaded) variables world and xGoal.

Problem 2.)

2.1) sphere\_potential and sphere\_potentialGrad use almost exactly the same structure but the function to evaluate the potential at each point is slightly different. I will briefly walk through the structure of the code, though it is commented in-line as well. In both cases, we need to sum the effects of each obstacle at the position xEval. To do this, we iterate through each obstacle individually and then sum the result to the total value, which starts at zero. For each iteration, we first need to determine if xEval is inside the obstacle (or outside, in the case of the world’s boundary). If we are, then we set the potential equal to NaN and immediately return (invalid location). If not, then we need to proceed and determine if we’re inside that obstacle’s zone of influence. If we are, then we compute U or gradU and sum that to the totalvalue (which we set to zero initially in both cases). The functions for U and gradU were provided and are relatively straightforward to compute. After we have totaled all the individual contributions to the total field, we then return that value.

2.2) attractive\_potential is a quick function to determine the attractive potential of a field of type potential.type at a position xEval. The formula is given in the homework and is fast to compute, we simply need to determine whether we are using p = 1 (conic) or p = 2 (quadratic) by checking potential.type. potential\_attractive is a similarly quick function, but in this case we need to compute the gradient. This formula was not given, but is a simple calculation away. We can easily show that, in the case p = 1, the gradient of becomes, where x,y are the coordinates of X

Meanwhile, when p = 2, the gradient becomes

Which is, of course, easy to code into matlab.

2.3) The functions potential\_total and potential\_totalGrad are extremely quick functions which call their respective generating functions for both the attractive and repulsive potentials at a position xEval and returns the sum of the attractive and repulsive potentials/gradients.

2.4) The potential planner is just a gradient descent algorithm which loops through, up to 1000 times, and evaluates the gradient at the current location and then moves in the (negative) direction of the gradient multiplied by a step size epsilon.