Problem 1. Let A, B, C be points in 2D space. Let a be the distance between points B and C and let γ be the value of the obtuse angle $\angle BAC$ in degrees. In class, I derived the position fix for the case of a subtended angle with two finite beacons. For the case of a right angle and for the case of an acute angle, I proved that the position fix is a portion of a circle with a specified center and radius. **Prove the corresponding result for the case where the subtended angle is obtuse** by showing that A is located in a portion of a circle that contains points A, B, and C, and give the radius of the circle in terms of a and γ. (HINT: $\gamma > 90^o$, but $180^o - \gamma < 90^o$.)

Problem 2. Consider the 2D case of a position fix using two range measurements. I gave the equations for this case in class. Use fminunc to solve these equations.

a) **Test and demonstrate your code using an example of your choice.** For example, let L_1 and L_2 be positioned at coordinates $(0, 0)$ m and $(5, 5)$ m respectively. Assume that these yield range measurements of $R_1 = 2.5$ m and $R_2 = 5$ m, respectively. Using two different initial position guesses ($(10, 0)$ and $(0, 10)$ for example, NOT near the actual circle intersections), use fminunc to obtain the two possible position fixes. Plot the two circles and place dots on the position fixes you obtain. (NOTE: Use the 'OptimalityTolerance' option to increase the accuracy of fminunc)

b) **Using one of your initial guesses from a), plot the distance error versus iteration plot from the fminunc optimization to check on the rate of convergence.** Let the distance error be the measured distance between the final output of fminunc and the position at each iteration of the optimization. Use semilogy. (NOTE: plot should look similar to the ones in Figures 2 and 4 from the fminunc tutorial in the Canvas page)

c) Since there are two possible position fixes, **start fminunc at a grid of points in a rectangle in the plane and determine what position fix each point converges to.** Then, use scatter to plot each of these initial points and color code them depending on which of the position fixes they converged to. For example, let the points of a grid be such that $p \in \{-5, -2.5, \ldots, 7.5, 10\} \times \{-5, -2.5, \ldots, 7.5, 10\}$. Then, put a red dot on each of the points that converges to one position fix and put a blue dot on each one that converges to the other position fix. (NOTE: this is a 2D version of Figure 6 from the fminunc tutorial in the Canvas page)

Problem 3. Modify your code in Problem 2 for the case of three range measurements.

a) **Choose locations of the beacons and your location to test and demonstrate your code.** For example, let the beacons L_1, L_2, and L_3 be at $(0, 0)$, $(5, 5)$, and $(2.5, 0)$ m respectively and let your position be $(0.7212, 2.4080)$ m. Then, the distance from each beacon to you is approximately $R_1 = 2.5$ m, $R_2 = 5$ m, and $R_3 = 3$ m respectively. Use the positions of the beacons and their measured distances to obtain your position fix. Plot the circles and place dots on the position fixes you obtain through different initial guesses.

b) **Test your code on an inconsistent case, that is, where there is no point where the three circles meet.** For example, in the previous case, change the measured distances to $R_1 = 2.5$ m, $R_2 = 6$ m, and $R_3 = 2$ m. Plot the circles and place dots on the position fixes you obtain through different initial guesses.

c) **Check with a grid of initial conditions whether or not all converge to the same position fix.** For that purpose, first determine the points to which initial guesses converge and present a grid similar to the one in Problem 2 c), where all points in the grid are color coded depending on the point to which they converge.

Problem 4. Continuing Problem 3a), assume now that each range measurement is corrupted by noise modeled by a*(rand(1) - 0.5) in Matlab.

a) **Fix a (2 for example) and run your code for 100 random cases using a single initial guess.** For each iteration, add a*(rand(1) - 0.5) to R_1, R_2, and R_3 and save the obtained position fix. Then, use scatter to plot all of these on top of the three true circles in red. Put a dot of a different color on the true position fix (chosen in Problem 3).

b) **For a range of values of** a, ($a \in [0.05, 10]$ **for example) repeat what you did in a) without plotting all 100 cases.** Instead, for each value of a, plot the maximum distance from the true position out of the 100 cases and show how the position fix degrades as a function of sensor error. (NOTE: You might need to use semilogy instead of plot).