ECE 490: Introduction to Optimization **Spring 2017**

Project I: Unconstrained Optimization

R. Srikant **Due: Feb. 20**

For this project we will explore different ways to solve the problem

$$\min_{x \in \mathbb{R}^n} x^T Q x + b^T x + c := f(x) \tag{1}$$

Recall that coding in Python is required. Your code should be flexible enough to run for different values of n, Q, b and c. Here Q is assumed to be a positive definite matrix. One group member will be required to run the code in front of a TA and the grade will be handed out for all group members right then. The code should load the matrix Q, vector b, and scalar c from a file. See the website for practice files (you can use **numpy.loadtxt** to load the values).

(a)

Write your own code to solve (1) using an iterative method where the step size α_k is chosen using Armijo's rule (also known as backtracking line search). The book has guidelines on how to choose the parameters in Armijo's rule. Write the code so that the error tolerance, ϵ, is a variable so that you can try out different values for ϵ. By error tolerance we mean your code stops the iterations and outputs your estimate for the minimum when the following condition is satisfied at iteration k

$$\|\nabla f(x_k)\| < \epsilon$$

Please make sure to print the output x^*, $f(x^*)$, ϵ, and the number of iterations required for convergence.

(b)

Use matrix inversion (found in packages such as **numpy.linalg** or **scipy.linalg**) to check your solution to part (a). Print x^* and $f(x^*)$.

(c)

Use the package **scipy.optimize** to check your solutions to parts (a)-(b). Print x^* and $f(x^*)$.

What will happen during the demonstration session:

You will be given a set of $Q, b,$ and c, and you will be asked to use your code to generate the solutions. You are expected to be able to comfortably explain your code to the TA, and answer some follow-up questions.