MATH - 5486 Name:

Due: 3/30/20 *Homework 1: §2-3*

Instructions:

- You should put all files (source code, plots, write-up, etc) into a single zip file with the name `<lastname>_homework1.zip`. So mine would be `lattimer_homework1.zip`. This file should be submitted via Canvas.
- You are free to ask for my help. Otherwise, all your work should be your own. You are not allowed to copy code or use code snippets from the internet. You are not allowed to copy each other's work. These actions will be considered a violation of the honor code.
- All source code should be appropriately commented to include, at a minimum, a description of the project, your name, date created/modified, and a short description of each function or method.
- If you plan to write your code in something other than Matlab, Python, C++, or Fortran, please discuss with me first.
- Your write-up should be typed (Word or Latex is fine) and submitted as a PDF.

For this homework assignment we will focus on the optimization problem where we minimize the the Booth objective function given by

$$f(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \tag{1}$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{x} \in [-10, 10] \times [-10, 10]$$

A. Analysis

1. Analytically determine $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$.
2. What is a stationary point? Determine the stationary point for the Booth function.
3. Is this stationary point a strict local minimum? Why? Please use theorem(s) from class.
4. Is this stationary point a global minimum? Why? Please use theorem(s) from class.

B. Computation
Recall that with the typical line-search method, each iterate is defined as $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ where α_k is the optimal step length and $\mathbf{p}_k$ is the optimal direction at each k. The different methods vary only in how those values are chosen.

1. In a software language of your choice, write functions for $f(\mathbf{x})$, $\nabla f(\mathbf{x})$, and $\nabla^2 f(\mathbf{x})$. Then implement the steepest descent and the Newton line search methods with set the following conditions:
   i) Initial guess is $\mathbf{x}_0 = [0, 0]$
   ii) Stopping criteria is when one of the following is reached:
      - (Converged) Error tolerance is 10^{-10}. That is $f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \leq 10^{-10}$
      - (Not converged) Max number of iterations set to 100 ($k = 0, 1, \ldots, 99$)

- (Not converged) $\mathbf{x}_k$ is outside the domain $[-10, 10] \times [-10, 10]$

2. In this question, we evaluate different methods for choosing the step length for the steepest descent method. Using the above criteria, run your application using each of the following values for α_k? Does the algorithm converge? If not, why? If so, how many steps does it take to converge? Either way, what is the final error?

   a) $\alpha_k = 1$

   b) $\alpha_k = \|\nabla f(x_k)\|$

   c) $\alpha_k = 1/(k+1)$

3. Now let $\alpha_k = 1$. How many iterations does it take for the Newton's line search method to converge?

Bonus: Implement the bracket and zoom methods in Algorithm 3.5 and Algorithm 3.6 to find the optimal α_k. Using this, does it improve the convergence of the steepest descent algorithm?