BME 493/593, Spring 2019: Computational Methods for Imaging Science Assignment 3 (due March 25, 2018)

Instructions: The problems below require a mix of paper-and-pencil work and MATLAB implementation. Remember that, as stated in the syllabus, you should both

1. Submit a hard copy of your solutions (which will be collected on Monday, March 25 **before** the class starts), **and**
2. Upload a digital copy of your solutions on Canvas by 9:59 AM on Monday, March 25.

Please include printouts of the MATLAB code together with the generated results and figures in the hard copy of your solutions.
The digital copy uploaded on Canvas must be a scan (or *readable* cellphone picture) of the hard copy that you handed in at the beginning of class. The instructors will grade only the hard copy of your solutions. The digital copy uploaded on Canvas only serves as proof that the assignment was turned in on time. If you are late or miss class on the day the assignment is due, uploading your solution on Canvas before the due date ensures that your assignment is considered on time.

For this assignment you will need the following files (that can be found on canvas):

- `objectiveFunction.m`: This Matlab script implements the function $f(\boldsymbol{x})$ (together with its gradient and Hessian) that you will need for Problem 3.
- `cg_steihaug.m`: This Matlab script implements the conjugate gradient (CG) algorithm with early termination if a direction of negative curvature is encountered. You will need this function for your implementation of the inexact Newton CG in Problem 3.

Problem 1: Consider the unconstrained optimization problem

$$\min f(x, y) \equiv -\cos x \cos(y/10).$$

A) Find and classify all stationary points in the region $-\pi/2 \leq x \leq \pi/2, -10\pi/2 \leq y \leq 10\pi/2$.

B) There is a portion of the problem region within which the Hessian matrix of $f(x, y)$ is positive definite. Give expressions for this portion. You should be able to do this analytically.

C) Derive expressions for the search directions associated with the steepest descent and Newton methods.

D) Write a program that performs both iterations, both without a line search and with an exact line search. Note that you will not be able to find the value of the optimal step length analytically; instead, determine it numerically[1].

[1] You can use a built-in one-dimensional minimization function `fzero` in MATLAB.

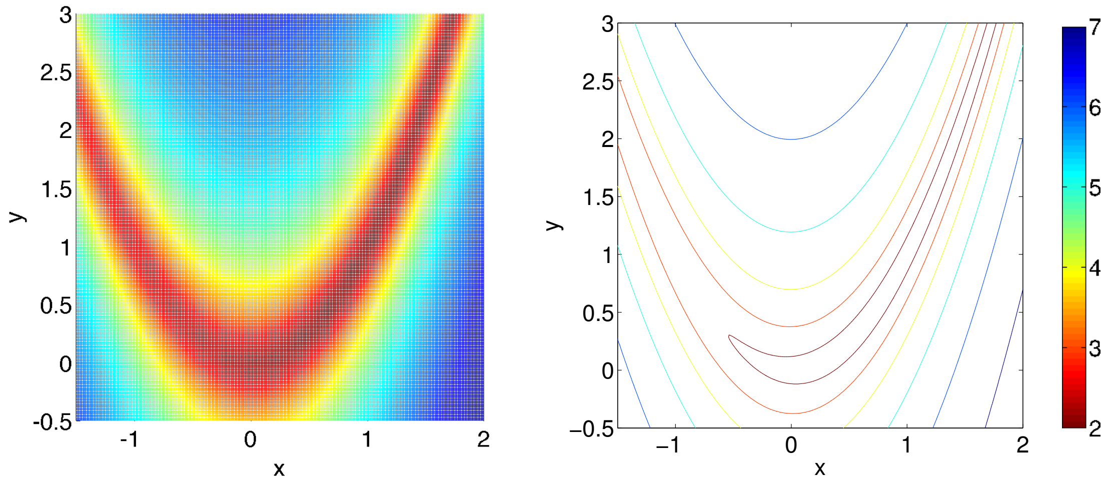

Figure 1: Two-dimensional Rosenbrock function with $\gamma = 100$ (we plot the log-function for better visualization).

E) Run your program for various initial guesses within the region. Verify the following:

   (a) Steepest descent converges to the minimum x^* for any starting point within the region.
   (b) Newton's method with line search converges to the minimum only for initial points for which the Hessian matrix is positive definite.
   (c) Newton's method without line search has an even more restricted radius of convergence.

F) What do you observe about the convergence rate in these cases?

Problem 2: Considering the following optimization problem:

$$\min f(x_1, x_2) := \gamma(x_1^2 - x_2)^2 + (x_1 - 1)^2.$$

The function $f(x_1, x_2)$ is known as the two-dimensional Rosenbrock's function and is shown in Fig. 1.

A) Find an expression for the gradient and Hessian of f.

B) Compare the performance of Newton and steepest descent for $\gamma = 1$ and $\gamma = 100$. Plot the sequence of iterates for each method on top of the contour plot of $f(\boldsymbol{x})$. What do you observe?

Problem 3: Write a program that implements the inexact Newton-conjugate gradient method as described in class[2] and use it to solve the following nonlinear least squares optimization problem[3]

$$\min f(\boldsymbol{x}) := \frac{1}{2}\|\boldsymbol{K}\boldsymbol{x} + \exp(\boldsymbol{x}) + \boldsymbol{b}\|^2 + \frac{1}{2}\|\boldsymbol{x}\|^2,$$

[2]Reference: S.C. Eisenstat and H.F Walker, *Globally convergent inexact Newton's method*, SIAM Journal on Optimization, Vol. 4, p.393–422, 1994.

[3]See provided Matlab file `objectiveFunction.m` for the implementation of $f(\boldsymbol{x})$ and its gradient & Hessian.

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^n$, ans $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. The entries of $\boldsymbol{b}$ and $\boldsymbol{K}$ are

$$[\boldsymbol{b}]_i = ih, \quad [\boldsymbol{K}]_{ij} = h \exp\left\{\frac{(i-j)^2}{2} h^2\right\} \quad i = 1, \ldots, n, j = 1, \ldots, n,$$

where $h = 1/n$.

Your implementation should have the following features:

- Terminate the CG iterations when $||\boldsymbol{H}_k \boldsymbol{p}_k + \boldsymbol{g}_k|| \leq \eta ||\boldsymbol{g}_k||$, and implement the following three options for η:
  - $\eta_k = 0.5$
  - $\eta_k = \min(0.5, \sqrt{||\boldsymbol{g}_k||/||\boldsymbol{g}_0||})$
  - $\eta_k = \min(0.5, ||\boldsymbol{g}_k||/||\boldsymbol{g}_0||)$
- Also terminate the CG iterations when a direction of negative curvature is detected within the CG iteration[4].
- For a line search, implement a backtracking Armijo line search as described in class.

A) Verify that the gradient $\boldsymbol{g}(\boldsymbol{x})$ and Hessian $\boldsymbol{H}(\boldsymbol{x})$ of $f(\boldsymbol{x})$ are given by

$$\boldsymbol{g}(\boldsymbol{x}) := \boldsymbol{J}^t \boldsymbol{r} + \boldsymbol{x}, \quad \boldsymbol{H}(\boldsymbol{x}) := \boldsymbol{J}^t \boldsymbol{J} + \boldsymbol{S} + \boldsymbol{I},$$

where $\boldsymbol{J} = \boldsymbol{K} + \operatorname{diag}\big(\exp(\boldsymbol{x})\big)$, $\boldsymbol{r} = \boldsymbol{K}\boldsymbol{x} + \exp(\boldsymbol{x}) + \boldsymbol{b}$, $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ is the identity matrix, and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with entries

$$[\boldsymbol{S}]_{ii} = \exp([\boldsymbol{x}]_i)[\boldsymbol{r}]_i.$$

B) Experiment with the different choices of η_k for $n = 1000$. Plot the norm of the gradient at each iteration in a *semilogy* scale and verify the theoretical convergence rates for the different choices of η. Use $\boldsymbol{x}_0 = \boldsymbol{0}$ as initial guess, and stop when the norm of the gradient is reduced by 8 orders of magnitude (i.e., $||\boldsymbol{g}_k||/||\boldsymbol{g}_0|| \leq 10^{-8}$).

C) What do you observe about the number of iterations taken by Newton as the problem dimension increases from $n = 250$ to $n = 500$ to $n = 1000$?

[4] Use the provided Matlab file `cg_steihaug.m`