Homework 3, 550.362 Optimization, Spring 2016

**Problem 1:** Consider Rosenbrock’s Function $f(x, y) = (1 - x)^2 + 100(y - x^2)^2$.

a) Using ad-hoc/elementary reasoning, identify a global min and argue that this min is unique.

b) Confirm that this function is not convex using a characterization of convexity (as opposed to the definition of convexity).

Hint: Consider the point $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$.

**Solution:**

At the point $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ note that the objective function is 0, and note that no negative values are possible for an objective function which is the sum of positive multiples of squares, so this point $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is a global min. Furthermore, the first summand in Rosenbrock’s function is 0 only when $x = 1$, and the second summand in Rosenbrock’s function is zero only when $y = x^2$, and these simultaneously occur only when $x = 1$ and then $y = 1^2 = 1$, hence the uniqueness of the global min. With regard to convexity, the gradient and Hessian of Rosenbrock’s function are

$$\nabla f \left( \begin{bmatrix} x \\ y \end{bmatrix} \right) = \begin{bmatrix} 400x^3 + 2x - 2 - 400xy \\ -200x^2 + 200y \end{bmatrix}$$

$$\nabla^2 f \left( \begin{bmatrix} x \\ y \end{bmatrix} \right) = \begin{bmatrix} 1200x^2 + 2 - 400y & -400x \\ -400x & 200 \end{bmatrix}$$

and, following the hint, we evaluate $\nabla^2 f \left( \begin{bmatrix} -1 \\ 3 \end{bmatrix} \right) = \begin{bmatrix} -106 & 40 \\ 40 & 200 \end{bmatrix}$ which, with eigenvalues $-111.1423$ and $205.1423$, is not positive semidefinite, thus $f$ is not convex.

**Problem 2:** Write a MATLAB program that solves the following one-dimensional minimization problem. Given fixed $\begin{bmatrix} u \\ v \end{bmatrix}$ and $\begin{bmatrix} a \\ b \end{bmatrix}$ in $\mathbb{R}^2$, you want to minimize $g(\alpha) := f(\begin{bmatrix} u \\ v \end{bmatrix} + \alpha \begin{bmatrix} a \\ b \end{bmatrix})$ over the real variable $\alpha$, where $f$ is Rosenbrock’s function. Your function should return the value $\alpha^*$ which exactly minimizes $g$ and, in so doing, your program is only allowed to evaluate $g$ at most three times. (From this problem and on, your arithmetic should be displayed with 14 decimal places.)

To test your program, if $\begin{bmatrix} u \\ v \end{bmatrix}$ and $\begin{bmatrix} a \\ b \end{bmatrix}$ are, respectively, $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ then $\alpha^* = .011970660581796$, and if $\begin{bmatrix} u \\ v \end{bmatrix}$ and $\begin{bmatrix} a \\ b \end{bmatrix}$ are, respectively, $\begin{bmatrix} 22 \\ 15 \end{bmatrix}$ and $\begin{bmatrix} 3 \end{bmatrix}$ then $\alpha^* = -.455493513119009$, and if $\begin{bmatrix} u \\ v \end{bmatrix}$ and $\begin{bmatrix} a \\ b \end{bmatrix}$ are, respectively, $\begin{bmatrix} 10 \\ 20 \end{bmatrix}$ and $\begin{bmatrix} 3 \end{bmatrix}$ then $\alpha^* = 4.271810092403096$.

You should submit the following:

a) Your MATLAB program.

b) A diary in which you run your program on these three examples above and three other exam-
plexes that you randomly select.

**Solution:** Note that elementary substitution of $[u \ v] + \alpha [u \ v]$ into $f$ yields (after simplifying) that

$$g(\alpha) = (200(v - u^2)(b - 2au) - 2(1 - u)a) \alpha + (100(b - 2au)^2 - 200(v - u^2)a^2 + a^2) \alpha^2$$

$$+ (-200(b - 2au)a^2) \alpha^3 + (100a^4) \alpha^4 + C$$

where $C$ is a constant; this is a polynomial of degree 4 in $\alpha$. Thus we compute the derivative of $g$

$$g'(\alpha) = (200(v - u^2)(b - 2au) - 2(1 - u)a) + (200(b - 2au)^2 - 400(v - u^2)a^2 + 2a^2) \alpha$$

$$+ (-600(b - 2au)a^2) \alpha^2 + (400a^4) \alpha^3,$$

which is a polynomial of degree 3 in $\alpha$. MATLAB can find the at-most three real roots of $g'(\alpha)$
and can compare the values of $g$ at these roots, and the root with least value of $g$ is $\alpha^*$.

**Problem 3:** Write a MATLAB program that performs the steepest descent method to minimize
Rosenbrock’s function. The first line of the program should be:

```matlab
function [output]=steepestRosenbrock(start,numiter,epsilon)
```

where `start` is a user-entered initial guess, `numiter` is a user-entered number which sets the max-
imum number iterations allowed before termination, `epsilon` is a positive number that causes
the algorithm to terminate when the norm of the gradient is less than `epsilon`, and `output` is the
final iterate at termination. (In other words, the steepest descent algorithm continues until the
gradient is “small enough” or until the number of iterations is “too large,” whichever comes first.)

To test your code, when the initial point is $[9 \ 8]$ then the subsequent iterates are $[2.887371760556714 \ 8.33957539694748]$, $[2.886880797279135 \ 8.332537571576379]$, $[2.886906749723861 \ 8.332583126153400]$, $[2.88567044963711 \ 8.325549737832029]$.

You should submit the following:

a) Your MATLAB program.

b) A diary in which you execute your program starting from $[9 \ 8]$ and run ten iterations. Make sure
your code prints the iterates; of course, the first ones should agree with what was given above.)

c) Starting from $[9 \ 8]$, how many iterations are needed to get gradient’s norm less than .00000001?

d) Run 10,000 iterations and record the location of the last iterate. Use MATLAB to plot the
iterates and print out three copies of your plot wherein you respectively zoom into different por-
tions of the trajectory corresponding to earlier iterates, intermediate iterates, then later iterates.
What is happening with the stepsize as the iterates advance?
Solution: For part c, the number of iteration required are 15051. In part d, the last iterate is at \[ \begin{bmatrix} 1.000020602327057 \\ 1.00004192202960 \end{bmatrix} \].

Problem 4: Write a MATLAB program that performs Newton’s method to minimize Rosenbrock’s function. The first line of the program should be:

\[
\text{function} \ [\text{output}]=\text{NewtonRosenbrock}(\text{start},\text{numiter},\text{epsilon})
\]

where \text{start} is a user-entered initial guess, \text{numiter} is a user-entered number which sets the maximum number iterations allowed before termination, \text{epsilon} is a positive number that causes the algorithm to terminate when the norm of the gradient is less than epsilon, and \text{output} is the final iterate at termination.

To test your code, when the initial iterate is \[ \begin{bmatrix} 9 \\ 8 \end{bmatrix} \], the next iterate is \[ \begin{bmatrix} 8.99945209232244 \\ 80.99013766186397 \end{bmatrix} \].

You should submit the following:

a) Your MATLAB program.

b) A diary in which you execute your program starting from \[ \begin{bmatrix} 9/8 \end{bmatrix} \] and run five iterations, excluding the starting vector. Make sure your program prints the iterates.

Solution: The first iterates of Newton’s Method are:

\[
\begin{bmatrix} 9/8, 8.99945209232244, 1.00048026281277, 1.000480222528553, 1.000000000000001, 1.000000000000003 \\ 80.99013766186397, -62.98258957301133, 1.00096068118738, 0.99999976938370, 1.00000000000003 \end{bmatrix}
\]