CS-4-6TE3/CES722-723
Assignment 1 Solution Set
Christopher Anand, Shefali Kulkarni-Thaker
October 25, 2011

Consider the so-called Rosenbrock (banana) function:

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Is this function convex? Make surface and contour plots to find out. Use the definition of convexity or the equivalent condition in terms of derivatives to prove that it is convex or not convex.

Rosenbrock (banana) function is not a convex function. This can be seen in the surface and contour plots shown below:

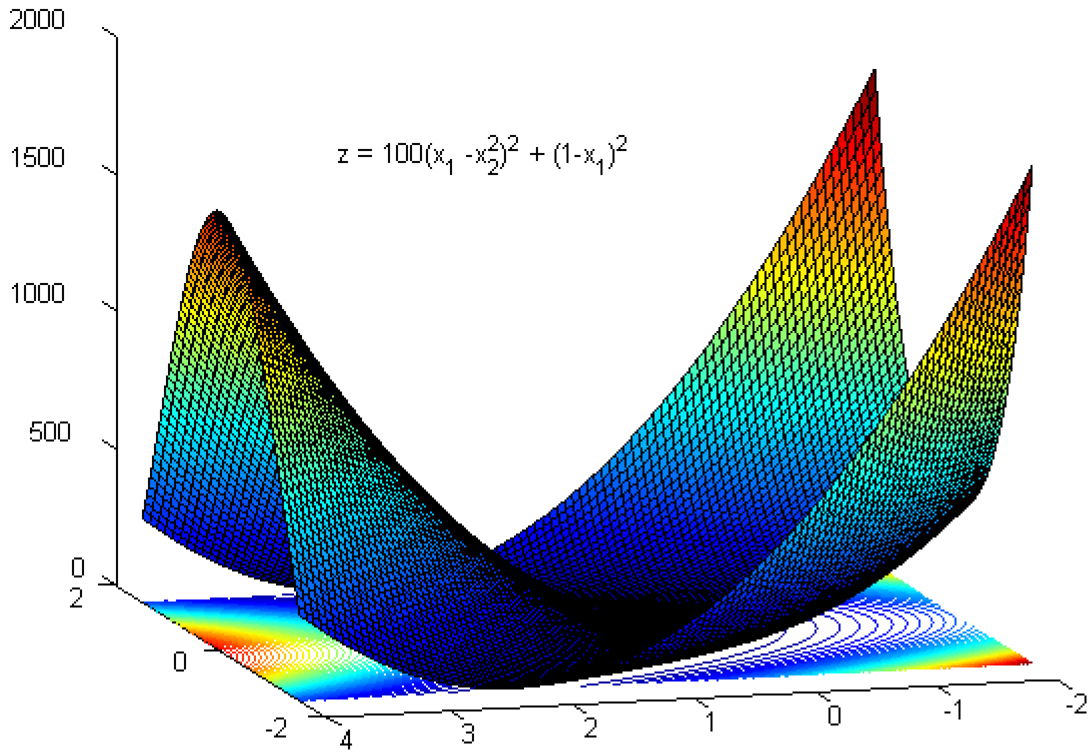

Figure 1: Banana Function $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

- Proof of non-convexity by counter-example is given by points $x^1 = (0, 0)$ and $x^2 = (-1, 2)$ and

$\lambda = 0.5$.

$$\begin{aligned} f(x^1) &= f(0,0) = 1 \\ f(x^2) &= f(-1,2) = 104 \\ f(\lambda x^1 + (1-\lambda)x^2) &= f(-0.5,1) = 58.5 \\ \lambda f(x^1) + (1-\lambda) f(x^1) &= \frac{1+104}{2} = 52.5 \end{aligned}$$

Thus, for the given points x^1 and x^2, we have shown that$f(\lambda x^1 + (1-\lambda)x^2) > \lambda f(x^1) + (1-\lambda)f(x^1)$ which violates the definition of convexity.

Apply the Gradient Descent method to minimize the Rosenbrock banana function, using the initial guess,

$$x^0 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \qquad x^1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \qquad x^2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

and making exact line searches, i.e. make a step size which gets as close to the minimum as possible. Perform 3 iterations.

$$\begin{aligned} f &= 100(x_2 - x_1^2)^2 + (1-x_1)^2 \\ \nabla f &= \begin{bmatrix} -400x_1x_2 + 400x_1^3 + 2x_1 - 2 \\ 200x_2 - 200x_1^2 \end{bmatrix} \qquad \nabla f(x^0) = \nabla f(-2,1) = \begin{bmatrix} -2406 \\ -600 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} x^1 &= x^0 + \alpha_0(-\nabla f(x^0)) \\ &= \begin{bmatrix} -2 \\ 1 \end{bmatrix} + \alpha_0 \begin{bmatrix} 2406 \\ 600 \end{bmatrix} \\ &= \begin{bmatrix} 2406\alpha_0 - 2 \\ 600\alpha_0 + 1 \end{bmatrix} \end{aligned}$$

Exact Line Search:

$$\min_{\alpha_0 > 0} f(x^0 + \alpha_0 d^0)$$

$$\min_{\alpha_0 > 0} f\left(\begin{bmatrix} 2406\alpha_0 - 2 \\ 600\alpha_0 + 1 \end{bmatrix} \right)$$

Minimum is given at $f' = 0$. i.e. $\nabla f(x)^T d^0 = 0$. The following MATLAB code will solve the problem of minimization and rest of the steps.

```
function Q2(x,y,iter)
dx = @(x,y)(-400*x*y + 400*x^3-2*x-2);
dy = @(x,y)(200*y-200*x^2);
banana = @(x,y)(100*(y-x^2)^2 + (1-x)^2);
str = sprintf('%f\t%f\t%f\n',x,y,banana(x,y));
disp(str);
for i =1:iter
    alpha = fminbnd(@(alpha) ros(alpha,x,y),0,1000);
    a = x - alpha*dx(x,y);
    b = y - alpha*dy(x,y);
    x = a;
```

```
    y = b;
    fval = banana(x,y);
    str = sprintf('%f\t%f\t%f\t%f\n',x,y,alpha,fval);
    disp(str);
end
end
```

```
function val = ros(alpha,x,y)
dx = @(x,y)(-400*x*y + 400*x^3 - 2*x-2);
dy = @(x,y)(200*y-200*x^2);
banana = @(x,y)(100*(y-x^2)^2 + (1-x)^2);
a = x - dx(x,y)*alpha;
b = y - dy(x,y)*alpha;
val = banana(a,b);
end
```

```
>> Q2(-2,1,3)
-2.000000    1.000000    909.000000
1.386733     1.847389    0.001412     0.721687
-1.766836    3.130411    0.084812     7.662954
-1.767092    3.130353    0.000033     7.662787

>> Q2(-1,2,3)
-1.000000    2.000000    104.000000
-1.354807    1.822597    0.000887     5.561768
1.760633     3.102210    0.495789     0.579130
1.761087     3.102180    0.000063     0.579310

>> Q2(7,2,7)
7.000000     2.000000    220936.000000
0.502552     2.464160    0.000049     489.365723
1.295484     1.680550    0.001772     0.087827
1.295904     1.680517    0.000073     0.087691
```

Again with the band function, perform two steps of a Golden Section line-search starting at point $x^0 = \begin{bmatrix}-1\\-1\end{bmatrix}$ with search direction $\begin{bmatrix}1\\1\end{bmatrix}$ on step-length interval $\begin{bmatrix}0\\1\end{bmatrix}$

Here $a_0 = (-1,-1)$. We choose a b_0 in the direction of $d = (1,1)$. The choice of b_0 will determine the interval in which we are seeking the minimum. Let $b_0 = [0,0]$. (Note, here that the minimum is at point b_0 where the function value is 1. This is not the global minimum for the function, but local minimum in the given initial interval.) Then, any point x of the segment can be described as a function of a parameter α.

$$x(k) = a + \alpha(b-a) \tag{1}$$

The two points, which are used in Golden-Section line-search to divide the interval, correspond to $\alpha_1 = 1-\tau$ and $\alpha_2 = \tau$, where $\tau = 0.618$ is the decreasing ratio. So, $x_1 = x(\alpha_1) = x(1-\tau)$ and $x_2 = x(\alpha_2) = x(\tau)$ .The iterative procedure works as follows:

$$\begin{aligned}
\text{if } f(x_1) < f(x_2) \textit{ then } [a,b] &\implies [a,x_2]\\
\text{if } f(x_1) > f(x_2) \textit{ then } [a,b] &\implies [x_1,b]\\
\text{if } f(x_1) = f(x_2) \textit{ then } [a,b] &\implies [x_1,x_2]
\end{aligned}$$

$$
\begin{aligned}
&x_1 = (-1,-1) + 0.382(1,1) = (-0.618,-0.618) \qquad f(x_1) = 102.6\\
&x_2 = (-1,-1) + 0.618(1,1)^T = (-0.382,-0.382)^T \qquad f(x_1) = 29.78\\
&f(x_1) > f(x_2) \implies [x_1, b]\\
&a = (-0.618,-0.618) \qquad b = (0,0)^T\\
&x_1 = (-0.618,-0.618) + 0.382(0.618,0.618) = (-0.382,-0.382) \qquad f(x_1) = 29.78\\
&x_2 = (-0.618,-0.618) + 0.618(0.618,0.618) = (-0.2361,-0.2361) \qquad f(x_2) = 10.04\\
&f(x_1) > f(x_2) \implies [x_1, b]\\
&a = (-0.382,-0.382) \qquad b = (0,0)^T\\
&x_1 = (-0.382,-0.382) + 0.382(0.382,0.382) = (-0.2361,-0.2361) \qquad f(x_1) = 10.04\\
&x_2 = (-0.382,-0.382) + 0.618(0.382,0.382) = (-0.1459,-0.1459) \qquad f(x_2) = 4.11\\
&f(x_1) > f(x_2) \implies [x_1, b]\\
&a = (-0.2361,-0.2361) \qquad b = (0,0)^T\\
&\vdots
\end{aligned}
$$

Other approach is to reduce the step-length interval.

$$
\begin{aligned}
&x_1 = (-1,-1) + 0.382(1,1) = (-0.618,-0.618) \qquad f(x_1) = 102.6\\
&x_2 = (-1,-1) + 0.618(1,1)^T = (-0.382,-0.382)^T \qquad f(x_1) = 29.78\\
&f(x_1) > f(x_2) \implies \text{new step length interval is } [0.382, 1]\\
&\alpha_3 = \tau * (1 - 0.382) + 0.382 = 0.764\\
&x_3 = (-1,-1) + 0.764(1,1) = (-0.236,-0.236) \qquad f(x_1) = 10.04\\
&f(x_3) > f(x_2) \implies \text{ new step length interval is } [0.618, 1]\\
&\vdots
\end{aligned}
$$

Which of the following are convex problems

$$
\begin{aligned}
&\min \quad xcos(x)\\
&\min \quad -log(x) + x^2\\
&\min \quad x\sqrt{x}\\
&\min \quad -log(x) - log(10 - x)
\end{aligned}
$$

- Non-convex. $f(x) = xcos(x)$. Proof can be given by counter example using $x_1 = 0$, $x_2 = 1.6$ and $\lambda = 0.5$
- Convex. $f(x) = -log(x) + x^2$. $f'(x) = \frac{-1}{x} + 2x$ and $f''(x) = \frac{1}{x^2} + 2 > 0$.
- Convex. $f(x) = x\sqrt{x} = x^{\frac{3}{2}}$. $f'(x) = \frac{3}{2}\sqrt{x}$ and $f''(x) = \frac{3}{4\sqrt{x}} > 0$.
- Convex. $f(x) = -log(x) - log(10 - x)$. $f'(x) = \frac{-1}{x} + \frac{1}{10 - x}$ and $f''(x) = \frac{1}{x^2} + \frac{1}{(10 - x)^2} > 0$

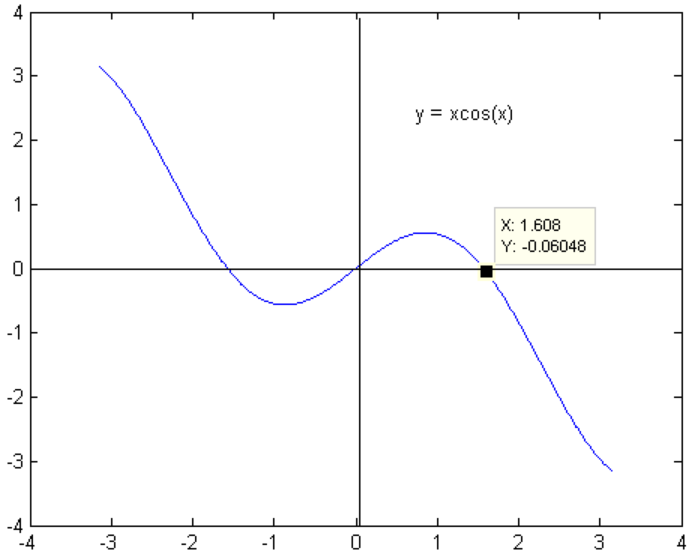

Figure 2: Graph for $f(x) = xcos(x)$

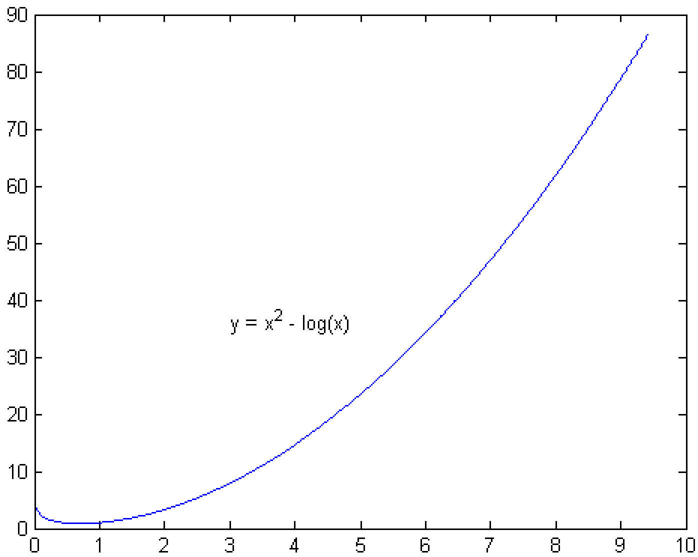

Figure 3: Graph for $f(x) = -log(x) + x^2$

In the general optimization strategy for unconstrained optimization, why do we need stopping criteria, and give three possible stopping criteria with a justification of the advantage of each

1. Number of iterations: This is safeguard that guarantees the termination of an algorithm when other stopping criteria fail due to numerical instability or poor step length choice.
2. $\|f_k - f_{k-1}\| \leq \epsilon$. Here ϵ is a small number that mostly equals the machine precision. If the algorithm is not sufficiently decreasing the function value, it could mean that either we are near the minimum or our choice or step length is not good. In either case the algorithm terminates.
3. $\|x_k - x_{k-1}\| \leq \epsilon$. Here ϵ is a small number that mostly equals the machine precision. If the

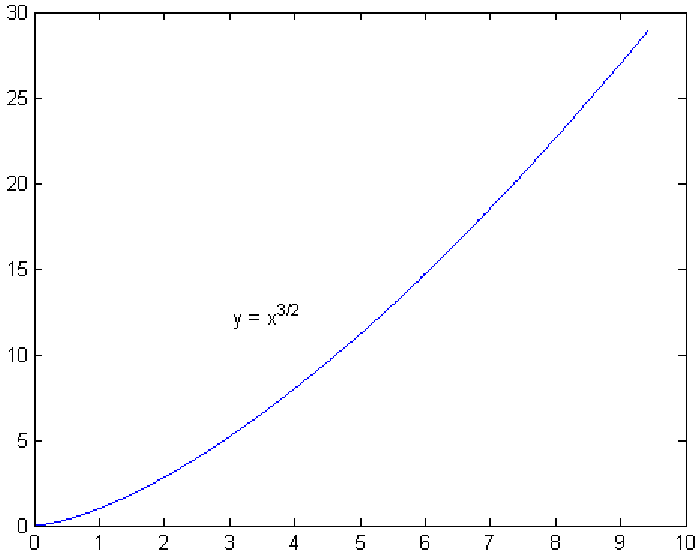

Figure 4: Graph for $f(x) = x\sqrt{x}$

algorithm is does not find significant change in the solution $x - vector$, it could mean that it is close to minimum, causing termination of algorithm.

4. $\|\nabla f(x)\| \leq \epsilon$. For an unconstrained optimization problem the minimum is found at the point where the gradient of the function is zero. This criteria will check if the gradient is zero within acceptable tolerance at the given point, causing the algorithm to terminate.