CL603: Optimization
Tutorial 2

Course Instructor: Prof. Mani Bhushan (mbhushan@iitb.ac.in)
Course TAs: Om Prakash (prakash.om@iitb.ac.in),
Varun (varun.p@iitb.ac.in)

February 1, 2020

Specific Aim: To understand and be able to write a code in Python or MATLAB to obtain a minimizer of a function using the **Newton's Method**.

A pseudo-code for Newton's method with full step is given at the end.

Consider the following function (known as Rosenbrock function):

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{1}$$

Use $x_{\text{initguess}} = [2 \;\; 2]^T$, $N = 100$ as the maximum number of iterations and $\epsilon = 10^{-8}$ as the tolerance on square of gradient-norm.
Do the following in Python or MATLAB:

1. Implement the Newton's Method with **analytically** computed gradient vector and Hessian matrices.
   (a) Plot $\mathbf{x}$ versus iteration number i.e. x_1 with iteration number and x_2 with iteration number in same figure.
   (b) Generate a figure which shows the value of $f(\mathbf{x})$ versus iteration number.
   (c) Label the axis and give title in each figure you generate.
2. Now, implement the Newton's Method with **numerically** computed gradient vector and Hessian matrices and also generate the figure as mentioned in part 1. Numerical computation of gradient was discussed in tutorial 1. For the Hessian, the following expressions can be used:

$$\frac{\partial^2 f(x_1,x_2)}{\partial x_1^2} = \frac{f(x_1+\Delta x_1, x_2) - 2f(x_1,x_2) + f(x_1-\Delta x_1, x_2)}{(\Delta x_1)^2}$$

$$\frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2} = \frac{1}{(4\Delta x_1 \Delta x_2)} (f(x_1+\Delta x_1, x_2+\Delta x_2) + f(x_1-\Delta x_1, x_2-\Delta x_2) - f(x_1+\Delta x_1, x_2-\Delta x_2) - f(x_1-\Delta x_1, x_2+\Delta x_2))$$

   where $\Delta x_1, \Delta x_2$ are small perturbations in x_1, x_2 respectively.

Pseudo Code for Full Step Newton's Method:
Choose a starting point $\mathbf{x}^0$, $k = 0$.
While $\|\nabla f(\mathbf{x}^k)\|^2 > \epsilon_2$ and $k \leq N$:

1. At $\mathbf{x}^k$, evaluate $\nabla f(\mathbf{x}^k)$ and $\nabla^2 f(\mathbf{x}^k)$. If $\nabla^2 f(\mathbf{x}^k)$ is singular, STOP and exit with an error statement
2. Solve the linear system $\nabla^2 f(\mathbf{x}^k)\mathbf{p}^k = -\nabla f(\mathbf{x}^k)$.

3. Set $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{p}^k$ and $k = k + 1$.

[For step 2: to solve the system of linear equations, you can use (i) linsolve in matlab, or (ii) numpy.linalg.solve in python. If you have trouble with these, you can also $\mathbf{p}^k = -[\nabla^2 f(\mathbf{x}^k)]^{-1}\nabla(\mathbf{x}^k)$ but this requires explicit inversion of the Hessian]

Learning is fun. Best of Luck!