CL603: Optimization
Tutorial 3

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 7, 2020

Specific Aim: To implement and compare performances of quasi-Newton line search methods, namely BFGS and DFP on a test problem.

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 $x_{\text{initguess}} = [1.5 \;\; 1.5]^T$, $N = 15000$ as the maximum number of iterations (steepest descent requires large number of iterations) and $\epsilon = 10^{-8}$ as the tolerance on square of gradient-norm.
Do the following in Python or MATLAB:

1. Implement BFGS method (updating of approximation of inverse Hessian). Once again use analytically computed gradient. The initial guess for the inverse Hessian approximation (matrix **C**) can be taken as the identity matrix.

2. Implement DFP method (updating of approximation of inverse Hessian). Once again use analytically computed gradient. The initial guess for the inverse Hessian approximation (matrix **C**) can be taken as the identity matrix.

For both the methods, use a backtracking strategy to choose an acceptable α. The parameters for this backtracking strategy are: $\bar{\alpha} = 5, \rho = 0.8, c = 0.1$. Refer to "Algorithm Backtracking Line Search" in the notes to see the pseudo code. To visualize the results, do the following:

1. Plot $\mathbf{x}$ versus iteration number i.e. x_1 with iteration number. and x_2 with iteration number in same figure.

2. Generate a figure which shows the value of $f(\mathbf{x})$ versus iteration number.

3. Label the axis and give title in each figure you generate.

Learning is fun. Best of Luck!