One report/jupyter notebook per team must be submitted as soft copies to deepakns@iisc.ac.in. As before, please typeset it in jupyter notebook if you use Python or Julia or R, or latex (template is provided) if you use any other language or software. In the latter case, the codes should also be included with your submission. Place all files in a folder and submit the .zip (don't submit a tar.gz etc).

Please clearly specify the role of each team member in the solution. You must specify which problem was tackled by who all. Reference codes that you use from elsewhere.

Please keep the subject line of your email submission DS211:2019:MP1

Typically, this assignment should take your team between 12 to 20 person-hours.

Problem 1

Line Search Strategy.

(a) Consider the steepest descent method with exact line searches applied to the convex quadratic function $\frac{1}{2}\mathbf{x}^T A\mathbf{x} + b^T\mathbf{x} + c$. Mathematically, show that if the initial point is such that $\mathbf{x}_0 - \mathbf{x}^*$ is parallel to an eigenvector of A, then the steepest descent method will find the solution in one step.

(b) Program the steepest descent and Newton algorithms using the backtracking line search discussed in class.

(c) Use the above programs to minimize the classic Rosenbrock function. Set the initial step length to 1. At each iteration store the step lengths used by each method and make plots. Show the step lengths taken and iterates as plots. Do these for a start point of search $x_0 = (1.2, 1.2)^T$ and then for the starting point $x_0 = (-1.2, 1)^T$.

(d) Plot the convergence of the iterates and the objective function value. Evaluate the rate of convergence.

(e) Call built-in functions for steepest descent and newton method for your program of choice, and show the results for the above. Compare and evaluate your program.

(f) Compare the run-time of your program and built-in function. Is there a difference? Why or why not?

Problem 2

Trust Region Strategy.

(a) Write a program that implements the dogleg method. Choose B_k to be the exact Hessian. Apply it to solve the Rosenbrock's function. Experiment with the update rule for the trust region by changing the thresholds. Plot and show your results in as much detail as possible, highlighting the iterates, trust regions, and convergence.

(b) Let $f(x) = 20(x_2 - x_1^2)^2 + (1 - x_1)^2$. Draw its contour lines. At $x = (0, -1)$, draw the contour lines of the quadratic model assuming that B is the Hessian of f. Draw the family of solutions of the quadratic sub-problem as the trust region radius varies from 0 to 2. Repeat this at $x = (0, 0.5)$. Comment on what you observe.

Problem 3

Conjugate Gradient.

(a) Derive the Fletcher Reeves method.

(b) Show that for a quadratic function, with exact line search, the Polak-Reibiere formula, Hestenes-Stiefel formula and Fletcher-Reeves formula are identical.

(c) Implement the conjugate gradient method with Fletcher Reeves formula and use to it solve linear systems in which the Hessian is the Hilbert matrix, whose elements are $H_{i,j} = 1/(i + j - 1)$. Set the right-hand-side to $b = (1, 1, ..., 1)^T$ and the initial point to $x_0 = \mathbf{0}$. Try dimensions $n = 5, 8, 12, 20$ and report the number of iterations required to reduce the residual below 10^{-6}.

(d) Construct matrices with various eigenvalue distributions (clustered and non-clustered) and apply the CG method to them. Comment on the behavior of CG method in terms of convergence.

Problem 4

Quasi-Newton.

(a) Write a program that implements the BFGS method. Apply it to minimize the Rosenbrock's function. Plot and show your results in as much detail as possible, highlighting the iterates, and convergence. Compare it to the other methods implemented in the above problems.

(b) Write a program that implements the SR1 Trust Region method. Apply it to minimize the Rosenbrock's function. Plot and show your results in as much detail as possible,

highlighting the iterates, and convergence. Compare it to the BFGS update formula. What are the advantages or disadvantages?