Instructions: (Please read carefully and follow them!)

Try to solve all problems on your own. If you have difficulties, ask the instructor or TAs.

In this session, we will continue with the implementation of gradient descent algorithm to solve problems of the form $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$. Recall that gradient descent takes a large number of iterations for some problems. In this lab, we will see some techniques to make gradient descent converge to the optimal point faster. We shall investigate the behavior of Newton's method on some problems and compare its performance against gradient descent algorithm. We will also discuss BFGS method to solve problems of the form $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$.

The implementation of the optimization algorithms in this lab will involve extensive use of the `numpy` Python package. It would be useful for you to get to know some of the functionalities of `numpy` package. For details on `numpy` Python package, please consult https://numpy.org/doc/stable/index.html

For plotting purposes, please use `matplotlib.pyplot` package. You can find examples in the site https://matplotlib.org/examples/.

Please follow the instructions given below to prepare your solution notebooks:

- Please use different notebooks for solving different Exercise problems.
- The notebook name for Exercise 1 should be `YOURROLLNUMBER_IE684_Lab03_Ex1.ipynb`.
- Similarly, the notebook name for Exercise 2 should be `YOURROLLNUMBER_IE684_Lab03_Ex2.ipynb`, etc and so on.

There are only 3 exercises in this lab. Try to solve all the problems on your own. If you have difficulties, ask the Instructors or TAs.

You can either print the answers using `print` command in your code or you can write the text in a separate text tab. To add text in your notebook, click `+Text`. Some questions require you to provide proper explanations; for such questions, write proper explanations in a text tab. Some questions require the answers to be written in LaTeX notation. (**Write the comments and observations with appropriate equations in LaTeX only.**) Some questions require plotting certain graphs. Please make sure that the plots are present in the submitted notebooks.

After completing this lab's exercises, click File → Download .ipynb and save your files to your local laptop/desktop. Create a folder with name `YOURROLLNUMBER_IE684_Lab03` and copy your `.ipynb` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE684_Lab03.zip`. Then upload only the `.zip` file to Moodle. **There will be some penalty for students who do not follow the proper naming conventions in their submissions**.

Please check the **submission deadline announced in moodle.**

The third Laboratory exercise aims to help you learn the **Quasi Newton (BFGS)** and **Gradient Descent with Scaling** methods.

Exercise 1 (15 marks) *In the last labs, when we tried to solve certain problems of the form* $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ *using gradient descent algorithm, we noticed that the algorithm needed a large number of iterations to find the minimizer. here we will discuss some remedy measures for this issue.*

The condition number of the Hessian plays a major role in the progress of the iterates of gradient descent towards the optimal solution point. Typically a large value of the condition number indicates that the problem is ***ill-conditioned*** *and hence leads to slow progress of the iterates towards the optimal solution point. Now we shall discuss a method which would help in better* ***conditioning*** *of the problem and hence would help in speeding up the progress of the iterates towards the optimal solution point.*

Let us first illustrate an equivalent transformation of the problem $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$*. Consider the transformation* $\mathbf{x} = \mathbf{My}$ *where* $\mathbf{M} \in \mathbb{R}^{n\times n}$ *is an invertible matrix and* $\mathbf{y} \in \mathbb{R}^n$ *and consider the equivalent problem* $\min_{\mathbf{y}\in\mathbb{R}^n} g(\mathbf{y}) \equiv \min_{\mathbf{y}\in\mathbb{R}^n} f(\mathbf{My})$*.*

Check: *Why are the two problems* $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ *and* $\min_{\mathbf{y}\in\mathbb{R}^n} g(\mathbf{y})$ *equivalent?*

Note that the gradient $\nabla_{\mathbf{y}} g(\mathbf{y}) = \mathbf{M}^\top \nabla_{\mathbf{x}} f(\mathbf{x})$ *and the Hessian is* $\nabla^2_{\mathbf{y}} g(\mathbf{y}) = \mathbf{M}^\top \nabla^2_{\mathbf{x}} f(\mathbf{x})\mathbf{M}$*.*

Hence the gradient descent update to solve $\min_{\mathbf{y}\in\mathbb{R}^n} g(\mathbf{y})$ *becomes:*

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \eta \nabla_{\mathbf{y}} g(\mathbf{y}^k) \tag{1}$$
$$\tag{2}$$

Pre-multiplying by $\mathbf{M}$*, we have:*

$$\mathbf{My}^{k+1} = \mathbf{My}^k - \eta \mathbf{M}\nabla_{\mathbf{y}} g(\mathbf{y}^k) \tag{3}$$
$$\implies \mathbf{x}^{k+1} = \mathbf{x}^k - \eta \mathbf{MM}^\top \nabla_{\mathbf{x}} f(\mathbf{x}^k) \tag{4}$$

Letting $\mathbf{D} = \mathbf{MM}^\top$*, we see that the update is of the form:*

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \eta \mathbf{D} \nabla f(\mathbf{x}^k) \tag{5}$$

Note that the matrix $\mathbf{D}$ *is symmetric and positive definite and hence can be written as* $\mathbf{D} = \mathbf{B}^2$*, where* $\mathbf{B}$ *is also symmetric and positive definite. Denoting* $\mathbf{B} = \mathbf{D}^{\frac{1}{2}}$*, we see that a useful choice for the matrix* $\mathbf{M}$ *is* $\mathbf{M} = \mathbf{B} = \mathbf{D}^{\frac{1}{2}}$*.*

The matrix $\mathbf{D}$ *is called a* ***scaling*** *matrix and helps in scaling the Hessian. We will consider* $\mathbf{D}$ *to be a diagonal matrix. Thus it would be useful to find a suitable candidate of the scaling matrix at each iteration which could help in significant progress of the iterates towards the optimal solution.*

This discussion leads to the following algorithm:

Algorithm 1 Gradient Descent with Scaling

Require: Starting point x_0, Stopping tolerance τ

Initialize $k = 0, p_k = -\nabla f(x_k)$

while $\|p_k\|_2 > \tau$ **do**

  Choose a suitable scaling matrix D_k

  $\eta_k = \arg\min_{\eta \geq 0} f(x_k + \eta D_k p_k) = \arg\min_{\eta \geq 0} f(x_k - \eta D_k \nabla f(x_k))$

  $x_{k+1} = x_k + \eta_k D_k p_k = x_k - \eta_k D_k \nabla f(x_k)$

  $k = k + 1$

Output: x_k

Consider the function $f(\mathbf{x}) = f(x_1, x_2) = x_1^2 + 4x_1x_2 + 1600x_2^2$.

1. *Write code to find the Hessian matrix of the function* $f(\mathbf{x})$ *and its condition number. Also find the minimizer and the minimum function value of* $f(\mathbf{x})$.

2. *In theory provided above, we claimed* $\mathbf{D}$ *is symmetric and positive definite. Provide justification for that claim. Also based on our discussion on condition number and the derivation of the gradient descent scheme with scaling, can you identify and write down the matrix* $\mathbf{Q}$ *whose condition number needs to be analyzed in the new gradient scheme with scaling ?*

3. *Based on the matrix* $\mathbf{Q}$*, can you come up with a useful choice for* $\mathbf{D}_k$ *(assuming* $\mathbf{D}_k$ *to be diagonal) ?, Implement* ***Algorithm - 1*** *for function* $f(\mathbf{x})$*, With starting point* $x_0 = (1, 4000)$ *and* $\tau = 10^{-12}$*, we will now study the behavior of gradient descent algorithm (without scaling) with backtracking line search, gradient descent algorithm (with scaling) with backtracking line search, for different choices of* ρ*. Take* $\alpha = 1, \gamma = 0.5$*. Try* $\rho \in \{0.9, 0.8, 0.75, 0.6, 0.5, 0.4, 0.25, 0.1, 0.01\}$*.For each* ρ*, record the final minimizer, final objective function value and number of iterations to terminate, for the gradient descent algorithm (without scaling) with backtracking line search and the gradient descent algorithm (with scaling) with backtracking line search. Prepare a plot where the number of iterations for both the algorithms are plotted against* ρ *values. Use different colors and a legend to distinguish the plots corresponding to the different algorithms. Comment on the observations. Comment about the minimizers and objective function values obtained for different choices of the* ρ *values for both the algorithms. Plot the level sets of the function* $f(\mathbf{x})$ *and also plot the trajectory of the optimization on the same plot for both with scaling and without scaling gradient descent algorithm and report your observations. (****Without scaling is nothing but the Algorithm 01 from Lab 02****.)*

4. *Based on our discussion on condition number and the derivation of the gradient descent scheme with scaling, can you identify and write down the matrix* $\mathbf{Q}$ *whose condition number needs to be analyzed in the new gradient descent scheme with scaling with* $\mathbf{D}_k = (\nabla^2 f(\mathbf{x}))^{-1}$ *?, For the problem* $\min_{\mathbf{x}} g(\mathbf{x}) = 512(x_2 - x_1^2)^2 + (4 - x_1)^2$*, Implement* ***Algorithm - 1*** *with starting point* $x_0 = (8, 8)$ *and a stopping tolerance* $\tau = 10^{-5}$ *, find the number of iterations taken by the gradient descent algorithm (without scaling) with backtracking line search, gradient descent algorithm (with scaling) with backtracking line search. For backtracking line search, use* $\alpha_0 = 1, \rho = 0.5, \gamma = 0.5$*. Note the minimizer and minimum objective function value in each case. Comment on your observations. Also note the condition number of the Hessian matrix involved in the gradient descent algorithm (without scaling) with backtracking line search and condition number of the matrix* $\mathbf{Q}$ *involved in the gradient descent algorithm (with scaling) with backtracking line search in each iteration. Prepare a plot depicting the behavior of condition numbers in both algorithms against iterations. Use different colors and legend to distinguish the methods. Comment on your observations.*

5. *Now can you solve the Part 04 of Exercise 03 from the last lab i.e. Lab 02 ?, Is your devised method aligns with this* ***Algorithm - 1*** *?*

Exercise 2 (15 marks) *Recall that in the last two labs and in this lab's first exercise, we had implemented Newton's method as a specific case of gradient descent with scaling. In this exercise, we will focus on the performance of Newton's method. We consider the Newton's method implementation illustrated in Algorithm 2.*

Algorithm 2 Newton's Method With Line Search

Require: Starting point x_0, Stopping tolerance τ
1: Initialize $k = 0$
2: **while** $\|\nabla f(x_k)\|_2 > \tau$ **do**
3: $\quad \eta_k = \arg\min_{\eta \geq 0} f(x_k - \eta(\nabla^2 f(x_k))^{-1} \nabla f(x_k))$
4: $\quad x_{k+1} = x_k - \eta_k (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$
5: $\quad k = k + 1$
6: **Output:** x_k

Consider the function

$$q(\mathbf{x}) = q(x_1, x_2) = \sqrt{x_1^2 + 4} + \sqrt{x_2^2 + 4}.$$

1. *What is the minimizer and minimum function value of $q(\mathbf{x})$?, Is the minimizer in unique ?, Is it local or global minima ?, Is the function $q(\mathbf{x})$ convex ?, explain each of them.*

2. *Consider $\eta_k = 1, \forall k = 1, 2, \ldots$ in **Algorithm 2**. With starting point $x_0 = (2, 2)$ and a stopping tolerance $\tau = 10^{-9}$, find the number of iterations taken by Newton's method. Compare the number of iterations with that taken by Newton's method (with backtracking line search) in **Algorithm 2**. Note the minimizer and minimum objective function value in each case. Comment on your observations. Plot the level sets of the function $q(\mathbf{x})$ and also plot the trajectory of the optimization on the same plot for both the Newton's method with and without backtracking line search.*

3. *Compare the number of iterations obtained for the two variants of Newton's method in the previous part with that of the gradient descent algorithm (without scaling) with backtracking line search (implemented in previous lab) using the starting point $(2, 2)$. For backtracking line search, use $\alpha_0 = 1, \rho = 0.5, \gamma = 0.5$. Also, compare the minimizer and minimum objective function value in each case. Comment on your observations.*

4. *Consider $\eta_k = 1, \forall k = 1, 2, \ldots$ in **Algorithm 2**. With starting point $x_0 = (16, 16)$ and a stopping tolerance $\tau = 10^{-9}$, find the number of iterations taken by Newton's method. Compare the number of iterations with that taken by Newton's method (with backtracking line search) in **Algorithm 2**. Note the minimizer and minimum objective function value in each case. Comment on your observations. Plot the level sets of the function $q(\mathbf{x})$ and also plot the trajectory of the optimization on the same plot for both the Newton's method with and without backtracking line search.*

5. *Compare the number of iterations obtained for the two variants of Newton's method in the previous part with that of the gradient descent algorithm (without scaling) with backtracking line search (implemented in previous lab) using the starting point $(16, 16)$. For backtracking line search, use $\alpha_0 = 1, \rho = 0.5, \gamma = 0.5$. Also, compare the minimizer and minimum objective function value in each case. Comment on your observations. .*

Exercise 3 (20 marks) *Recall that to solve problems of the form* $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$*, the update rule involved in Newton's method is of the form:*

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \eta^k(\nabla^2 f(\mathbf{x}^k))^{-1}\nabla f(\mathbf{x}^k). \tag{6}$$

Now we will discuss a method which avoids explicit computation of the inverse of Hessian matrix at each iteration, but is nearly efficient as the Newton's method. This method will be called BFGS named after the famous applied Mathematicians Broyden, Fletcher, Goldfarb and Shanno. The theoretical discussion as a reference to move ahead with this exercise can be found at BFGS Theory. ***Note :*** *Please go through the BFGS Theory without a fail as some steps may be needed to implement the algorithm which are not mentioned here in this sheet explicitly.*

Algorithm 3 BFGS Algorithm

Require: Starting point x_0, Stopping tolerance τ

Initialize $k = 0, B_0 =??$ ▷ Initialize Hessian approximation
while $\|\nabla f(x_k)\|_2 > \tau$ **do**
  Compute descent direction: $p_k = -B_k\nabla f(x_k)$
  Choose step size: $\alpha_k = \arg\min_{\alpha\geq 0} f(x_k + \alpha p_k)$
  Update iterate: $x_{k+1} = x_k + \alpha_k p_k$
  Compute new gradient: $s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
  Update Hessian approximation: $B_{k+1} =??$ ▷ Replace ?? by the update mentioned in BFGS Theory
  $k = k + 1$
Output: x_k

Consider the functions $f(\mathbf{x}) = f(x_1, x_2, x_3, ..., x_n) = \sum_{i=1}^{n-1}[4(x_i^2 - x_{i+1})^2 + (x_i - 1)^2], g(\mathbf{x}) = g(x_1, x_2, x_3, ..., x_n) = \sum_{i=1}^{n}[(x_1 - x_i^2)^2 + (x_i - 1)^2]$.

1. *What is the minimizer and minimum function value of* $f(\mathbf{x})$ *and* $g(\mathbf{x})$ *? Are both the function convex ? What is a suitable initial choice of* B *(denoted by* B_0*, i.e. Replacement of first ?? in the* ***Algorithm 3****)? Justify with proper reasons.*

2. *Implement* ***Algorithm 3*** *for solving* $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$*, Use backtracking line search with* $\alpha_0 = 0.9, \rho = 0.5, \gamma = 0.5$*. Take the starting point to be* $\mathbf{x_0} = (0, 0, ..., 0)$*. Take* $n \in \{1000, 2500, 5000, 7500, 10000\}$*, find minimizer of the objective function in each case and compute the time taken by the BFGS method with backtracking line search. Tabulate the time taken by BFGS method for each* n*.*

3. *Implement* ***Algorithm 3*** *for solving* $\min_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{x})$*, Use backtracking line search with* $\alpha_0 = 0.9, \rho = 0.5, \gamma = 0.5$*. Take the starting point to be* $\mathbf{x_0} = (0, 0, ..., 0)$*. Take* $n \in \{1000, 2500, 5000, 7500, 10000\}$*, find minimizer of the objective function in each case and compute the time taken by the BFGS method with backtracking line search. Tabulate the time taken by BFGS method for each* n*.*

4. *Implement* ***Algorithm 2*** *for solving* $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$*, Use backtracking line search with* $\alpha_0 = 0.9, \rho = 0.5, \gamma = 0.5$*. Take the starting point to be* $\mathbf{x_0} = (0, 0, ..., 0)$*. Take* $n \in \{1000, 2500, 5000, 7500, 10000\}$*, find minimizer of the objective function in each case and compute the time taken by the Newton's method with backtracking line search. Tabulate the time taken by Newton's method for each* n*.*

5. *Implement* ***Algorithm 2*** *for solving* $\min_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{x})$*, Use backtracking line search with* $\alpha_0 = 0.9, \rho = 0.5, \gamma = 0.5$*. Take the starting point to be* $\mathbf{x_0} = (0, 0, ..., 0)$*. Take* $n \in \{1000, 2500, 5000, 7500, 10000\}$*, find minimizer of the objective function in each case and compute the time taken by the Newton's method with backtracking line search. Tabulate the time taken by Newton's method for each* n*.*

6. *Compare the time taken by* ***Algorithm 3 - BFGS Method*** *with backtracking line search against the time taken by* ***Algorithm 2 - Newton's Method*** *with backtracking line search for each value of* n*. Plot the time taken by both methods vs* n *using different colors. Comment on your observations.*

Bibliography

1. Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2003, vol. 87.
2. Jiantao Jiao, Course Materials, https://people.eecs.berkeley.edu/~jiantao/227c2022spring/material.html
3. Matthias L. R. Haußer, Lecture Slides, https://people.maths.ox.ac.uk/hauser/hauser_lecture2.pdf
4. Jhon A. Jhonorio, CS 52000: Optimization: Quasi-Newton Methods, https://www.cs.purdue.edu/homes/jhonorio/16spring-cs52000-quasinewton.pdf