**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 works iteratively where in each iteration a suitable descent direction and an appropriate step length are chosen to update the current iterate. We will discuss a new idea to find the step length parameter.

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_Lab02_Ex1.ipynb`.
- Similarly, the notebook name for Exercise 2 should be `YOURROLLNUMBER_IE684_Lab02_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. 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_Lab02` and copy your `.ipynb` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE684_Lab02.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 second Laboratory exercise aims to help you learn the exact and approximate (inexact) line search methods for step size selection.

**Exercise 1 (15 marks)** *In this exercise, we will design a procedure to find a suitable step length. We consider the following algorithm:*

**Algorithm 1** Gradient Descent Procedure with Line Search to compute step length

```
Require: Starting point $x_0$, Stopping tolerance $\tau$
Initialize $k = 0$, $p_k = -\nabla f(x_k)$
while $\|p_k\|_2 > \tau$ do
    $\eta_k \leftarrow \arg\min_{\eta \geq 0} f(x_k + \eta p_k) = \arg\min_{\eta \geq 0} f(x_k - \eta \nabla f(x_k))$
    $x_{k+1} \leftarrow x_k + \eta_k p_k$
    $p_k \leftarrow -\nabla f(x_k)$
    $k \leftarrow k + 1$
Output: $x_k$
```

*Consider the functions* $f(\mathbf{x}) = f(x_1, x_2) = 256(x_2 - x_1^2)^2 + (2 - x_1)^2$ *and* $g(\mathbf{x}) = g(x_1, x_2) = (x_1 + 49)^2 + (x_2 - 36)^2$.

1. *Write the function* $g(\mathbf{x})$ *in the form* $\mathbf{x}^\top \boldsymbol{A}\mathbf{x} + \boldsymbol{b}^\top \mathbf{x} + c$, *where* $\mathbf{x} \in \mathbb{R}^2$, $\boldsymbol{A}$ *is a symmetric matrix of size* $2 \times 2$, $\boldsymbol{b} \in \mathbb{R}^2$ *and* $c \in \mathbb{R}$. *Also find the minimizer and the minimum function value of each of* $f(\mathbf{x})$ *and* $g(\mathbf{x})$.

2. *Find the analytical solution to* $\min_{\alpha \geq 0} g(\mathbf{x} - \alpha \nabla g(\mathbf{x}))$ *in closed form. Also prove or disprove that the analytical solution to* $\min_{\alpha \geq 0} f(\mathbf{x} - \alpha \nabla f(\mathbf{x}))$ *can be found in closed form.*

3. *Implement* ***Algorithm - 1*** *for function* $g(\mathbf{x})$, *starting from the initial point* $\mathbf{x}_0 = (36, -49)$, *explore a range of tolerances* $\tau = 10^{-p}$ *for* $p = 1, 2, \ldots, 15$. *Record the number of iterations required for the algorithm to converge for each tolerance. Generate a plot illustrating the relationship between the number of iterations and the tolerance values. Compare and contrast this plot with those obtained for the same function* $g(\mathbf{x})$ *using the* ***Algorithm - 2 of LAB-01*** *where fixed step length value* ($\eta = 0.001$) *was used. Plot the level sets of the function* $g(\mathbf{x})$ *and also plot the trajectory of the optimization on the same plot for both exact line search method and the fixed step length method of gradient descent algorithm and report your observations.*

4. *What may be the shortcomings of this algorithm and suggests a possible solution to deal with it ? (**Hint:** Use the answer of the part 2.)*

**Exercise 2 (15 marks)** *Recall that we implemented the gradient descent algorithm to solve* $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$. *The key components in the gradient descent iterations include the descent direction* $\mathbf{p}_k$, *which is set to* $-\nabla f(\mathbf{x}_k)$, *and the step length* $\eta_k$, *determined by solving an optimization problem (or sometimes kept constant across all iterations). Finding a closed-form expression as a solution to the optimization problem for a suitable step length might not always be possible. To address general situations, we will attempt to devise a different procedure in this particular exercise. To determine the step length, we will use the following property: Suppose a non-zero* $\mathbf{p} \in \mathbb{R}^n$ *is a descent direction at point* $\mathbf{x}$, *and let* $\gamma \in (0, 1)$. *Then there exists* $\varepsilon > 0$ *such that*

$$f(\mathbf{x} + \alpha \mathbf{p}) \leq f(\mathbf{x}) + \gamma \alpha \nabla f(\mathbf{x})^\top \mathbf{p}, \quad \forall \alpha \in (0, \varepsilon].$$

*This condition is known as a sufficient decrease condition.*

*Utilizing the concept of sufficient decrease, the step length* $\eta_k$ *can be determined using a backtracking procedure illustrated below to find an appropriate value of* $\varepsilon$.

**Algorithm 2** Backtracking (Inexact) Line Search

```
Require: $x_k, p_k, \alpha_0, \rho \in (0,1), \gamma \in (0,1)$
Initialize $\alpha = \alpha_0, p_k = -\nabla f(x_k)$
while $f(x_k + \alpha p_k) > f(x_k) + \gamma \alpha \nabla f(x_k)^\top p_k$ do
    $\alpha = \rho\alpha$
Output: $\alpha$
```

*This is known as approximate (inexact) line search method to find the step length at each iteration.*

1. *Consider the function* $g(\mathbf{x})$ *from Exercise-1 for this part and with the starting point* $\mathbf{x}_0 = (100, 100)$ *and* $\tau = 10^{-10}$*, we will investigate the behavior of the backtracking line search algorithm for different choices of* $\alpha_0$*. Set* $\gamma = \rho = 0.5$ *and try* $\alpha_0 \in \{1, 0.9, 0.75, 0.6, 0.5, 0.4, 0.25, 0.1, 0.01\}$*. For each* $\alpha_0$*, record the final minimizer, final objective function value, and the number of iterations taken by the gradient descent algorithm with backtracking line search to terminate. Generate a plot where the number of iterations is plotted against* $\alpha_0$ *values. Provide observations on the results, and comment on the minimizers and objective function values obtained for different choices of* $\alpha_0$*. Check and comment if, for any* $\alpha_0$ *value, gradient descent with backtracking line search takes a lesser number of iterations compared to the gradient descent procedure with exact line search. Plot the level sets of the function* $g(\mathbf{x})$ *and also plot the trajectory of the optimization on the same plot for both inexact line search method and the fixed step length method of gradient descent algorithm and report your observations.*

2. *Redo (1) using the function* $f(\mathbf{x})$ *from Exercise-1 and also keep in mind the answer of the part (2) from Exercise-1.*

3. *What do you conclude from (1) and (2) regarding these two line search approaches ?*

**Exercise 3 (20 marks)** *Consider the functions* $f(\mathbf{x}) = f(x_1, x_2) = x_1^2 + x_2^2 + 9, g(\mathbf{x}) = g(x_1, x_2, x_3, ...., x_n) = \sum_{i=1}^{n} \frac{1}{P(i)}(x_i - i^2)^2$*, Where* $P(y)$ $(y \in \mathbb{R})$ *is a periodic function with the period of 4 and* $P(7) = \frac{1}{4}, P(77) = \frac{1}{16}, P(222) = \frac{1}{256}, P(4444) = \frac{1}{64}$.

1. *What is the minimizer and minimum function value of* $f(\mathbf{x})$ *and* $g(\mathbf{x})$ *? Are both the function convex ? Explain.*

2. *Implement Gradient Descent with the exact line search for* $f(\mathbf{x})$ *and also implement Newton's Method (From LAB-01) for* $f(\mathbf{x})$*. Note down the time taken, number of iterations required for convergence, record the final minimizer, final objective function value for both the implementations. Provide observations on the results, and comment on the minimizers and objective function values so obtained. Plot the level sets of the function* $f(\mathbf{x})$ *and also plot the trajectory of the optimization on the same plot for both the implementations and report your observations. (Take* $\tau = 10^{-15}, x_0 = (1000, -1000)$*.)*

3. *For* $n \in \{2, 20, 200, 2000, 5000, 10000, 15000, 20000, 30000, 50000, 100000, 200000\}$ *Implement Gradient Descent with the exact line search for* $g(\mathbf{x})$ *and also implement Newton's Method (From LAB-01) for* $g(\mathbf{x})$*. Note down the time taken, number of iterations required for convergence, record the final minimizer, final objective function value for both the implementations. Provide observations on the results, and comment on the minimizers and objective function values so obtained. Only for* $n = 2$ *plot the level sets of the function* $g(\mathbf{x})$ *and also plot the trajectory of the optimization on the same plot for both the implementations and report your observations. (Take* $\tau = 10^{-15}, x_0 = (1, 2, 3, ....., n)$*.)*

4. *Report for which value of n in (3) the Newton's Method implementation got failed due to Google Colab Crash. If we change the starting point* $x_0$ *in part (2), then will the number of iterations required for convergence decreases, increases or remains same ?, What about the same in part (3) ? Explain. Now consider* $g(\mathbf{x})$ *for this n at which Google Colab Crash occurs and devise a method, implement it such that we get the number of iterations required for convergence as 1. Here, in the implementation part you are free to choose any starting point which*

*is related to your devised method but take $\tau = 10^{-15}$. Explain the devised method clearly and provide the logical observations of the results that you got. (**Hint:** Use the results from (2) and think about relating $g(\mathbf{x})$ and $f(\mathbf{x})$.)*

## 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