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 start with implementation of algorithms for solving nonlinear optimization problems. Today, we will discuss gradient descent to solve problems of the form $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$.

Gradient descent is one of the oldest algorithms (dating back to Cauchy), yet simple and popular in many scientific communities even today. Gradient descent works iteratively where in each iteration a suitable descent direction and an appropriate step length are chosen to update the current iterate.

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_Lab1_Ex1.ipynb`.
- Similarly, the notebook name for Exercise 2 should be `YOURROLLNUMBER_IE684_Lab1_Ex2.ipynb`, etc.
- Please post your doubts in MS Teams Discussion Forum channel so that TAs can clarify.

There are only 2 exercises in this lab. Try to solve all problems on your own. If you have difficulties, ask the Instructors or TAs.

Only the questions marked [**R**] need to be answered in the notebook. 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. Please see the demo video to know how to write LaTeX in Google notebooks. 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_Lab1` and copy your `.ipynb` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE684_Lab1.zip`. Then upload only the `.zip` file to Moodle. There will be extra marks for students who follow the proper naming conventions in their submissions.

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

Exercise 1: Gradient Descent

We will start with a procedure which helps to find a minimizer of the function $f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n$.

We will use the following gradient descent type algorithm:

```
Input: Starting point $x^0$, Stopping tolerance $\tau$, Step length $\eta$.
Initialize  $k = 0$
while $\|\nabla f(\mathbf{x}^k)\|_2 > \tau$ do
    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \eta \nabla f(\mathbf{x}^k)$
    $k = k + 1$
end
Output:  $\mathbf{x}^k$.
```

Algorithm 1: Gradient Descent Procedure with constant step length

1. Note that in the notebook file shared with you, Algorithm 1 is implemented to solve $f(\mathbf{x}) = f(x_1, x_2) = (x_1 + 100)^2 + (x_2 - 25)^2$ with a constant step length $\eta = 0.1$, stopping tolerance $\tau = 0.001$ and with the starting point $\mathbf{x}^0 = (10, 10)$.
2. **[R]** What is the minimizer and minimum function value of $f(\mathbf{x}) = f(x_1, x_2) = (x_1 + 100)^2 + (x_2 - 25)^2$?
3. **[R]** With starting point $\mathbf{x}^0 = (10, 10)$ and $\eta = 0.1$, we will now study the behavior of the algorithm for different tolerance values. Try $\tau = 10^{-p}$ where $p = 1, 2, \ldots, 10$. For each τ, record the final minimizer, final objective function value and number of iterations taken by the algorithm to terminate. Prepare a plot where the number of iterations is plotted against τ values. Comment on the observations. Comment about the minimizers and objective function values obtained for different choices of the tolerance values.
4. **[R]** With starting point $\mathbf{x}^0 = (10, 10)$ and $\tau = 10^{-5}$, we will study the behavior of the algorithm for different step length values. Try $\eta \in \{0.0001, 0.001, 0.01, 0.1, 0.2, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. For each η, record the final minimizer, final objective function value and number of iterations taken by the algorithm to terminate. Prepare a plot where the number of iterations is plotted against η values. Comment on the observations. Comment about the minimizers and objective function values obtained for different choices of the step length values.
5. **[R]** With $\tau = 10^{-5}$ and $\eta = 0.1$, we will study the behavior of the algorithm for different starting points. Consider $\mathbf{x}^0 \in \{(10000, 10000), (500, 0), (0, 1000), (1, 1), (-500, -2)\}$. Prepare a table listing the final minimizer, final objective function value and number of iterations taken by the algorithm to terminate for the different starting points. Comment on your observations.

Exercise 2: Line Search

In this exercise, we will design a procedure to find a suitable step length. We consider the following algorithm:

```
**Input:** Starting point $x^0$, Stopping tolerance $\tau$.
**Initialize** $k = 0$
**while** $\|\nabla f(\mathbf{x}^k)\|_2 > \tau$ **do**
    $\eta^k = \arg\min_{\eta \geq 0} f(\mathbf{x}^k - \eta \nabla f(\mathbf{x}^k))$
    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \eta^k \nabla f(\mathbf{x}^k)$
    $k = k + 1$
**end**
**Output:** $\mathbf{x}^k$.
```

Algorithm 2: Gradient Descent Procedure with line search to compute step length

1. **[R]** Write the function $f(\mathbf{x}) = f(x_1, x_2) = (x_1 + 100)^2 + (x_2 - 25)^2$ in the form $\mathbf{x}^\top \mathbf{A}\mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + c$, where $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{A}$ is a symmetric matrix of size 2×2, $\mathbf{b} \in \mathbb{R}^2$ and $c \in \mathbb{R}$.

2. **[R]** It turns out that for a function of the form $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + c$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$, the analytical solution to $\min_{\alpha \geq 0} f(\mathbf{x} - \alpha \nabla f(\mathbf{x}))$ can be found in closed form. Find the solution.

3. Use the answers to Questions 1 and 2 to design a procedure to compute the step length η^k in Algorithm 2 for the function $f(\mathbf{x}) = f(x_1, x_2) = (x_1 + 100)^2 + (x_2 - 25)^2$. Implement the procedure as a Python function (complete the code in `compute_steplength` module in the notebook).

4. **[R]** With starting point $\mathbf{x}^0 = (10, 10)$ and the new module to compute η^k, try $\tau = 10^{-p}$ where $p = 1, 2, \ldots, 10$. For each τ, record the number of iterations taken by the algorithm to terminate. Prepare a plot where the number of iterations is plotted against τ values. Compare and contrast the plot with the plots obtained in Exercise 1 with fixed step length values.