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 Newton's method to solve problems of the form $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$. In this lab, we shall investigate the behavior of Newton's method on some problems and compare its performance against gradient descent algorithm.

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

In some cases you might need to use the matrix square root function `sqrtm` from the package `scipy.linalg`. You can use `from scipy.linalg import sqrtm` and then use `sqrtm(A)` to find the matrix square root of matrix `A`.

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_Lab4_Ex1.ipynb`.
- Similarly, the notebook name for Exercise 2 should be `YOURROLLNUMBER_IE684_Lab4_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 (posted in Lab 1) 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. Please include all answers in your `.pynb` files.

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_Lab4` and copy your `.ipynb` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE684_Lab4.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: Newton's method

Recall that in the last lab, we had implemented Newton's method as a specific case of gradient descent with scaling. In this lab, we will focus on the performance of Newton's method on some problems. We consider the Newton's method implementation illustrated in Algorithm 1.

Input: Starting point x^0, Stopping tolerance τ
Initialize $k = 0$
while $\|\nabla f(\mathbf{x}^k)\|_2 > \tau$ **do**
  $\eta^k = \arg\min_{\eta \geq 0} f(\mathbf{x}^k - \eta(\nabla^2 f(\mathbf{x}^k))^{-1}\nabla f(\mathbf{x}^k))$
  $\mathbf{x}^{k+1} = \mathbf{x}^k - \eta^k(\nabla^2 f(\mathbf{x}^k))^{-1}\nabla f(\mathbf{x}^k)$
  $k = k + 1$
end
Output: $\mathbf{x}^k$.

Algorithm 1: Newton's method

---

1. Please reuse the code you had written in the past labs to implement the Newton's method in Algorithm 1.
2. Consider the function $f(\mathbf{x}) = 400x_1^2 + 0.004x_2^4$. Write code for implementing all relevant modules necessary for Newton's method and gradient descent method to solve $\min_{\mathbf{x}} f(\mathbf{x})$.
3. **[R]** Consider $\eta^k = 1, \forall k = 1, 2, \ldots$ in Algorithm 1. With starting point $\mathbf{x}^0 = (2, 2)$ and a stopping tolerance $\tau = 10^{-9}$, find the number of iterations taken by the Newton's method. Compare the number of iterations with that taken by Newton's method (with backtracking line search) in Algorithm 1. Note the minimizer and minimum objective function value in each case. Comment on your observations.
4. **[R]** Compare the number of iterations obtained for the two variants of Newton's method in the previous question, with that of gradient descent algorithm (without scaling) with backtracking line search, gradient descent algorithm (with scaling using a diagonal matrix) with backtracking line search (implemented in previous labs), with 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.

Exercise 2:

Consider the function

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

1. Write code for implementing all relevant modules necessary for Newton's method and gradient descent method to solve $\min_{\mathbf{x}} q(\mathbf{x})$.
2. **[R]** Consider $\eta^k = 1, \forall k = 1, 2, \dots$ in Algorithm 1. With starting point $\mathbf{x}^0 = (2, 2)$ and a stopping tolerance $\tau = 10^{-9}$, find the number of iterations taken by the Newton's method. Compare the number of iterations with that taken by Newton's method (with backtracking line search) in Algorithm 1. Note the minimizer and minimum objective function value in each case. Comment on your observations.
3. **[R]** Compare the number of iterations obtained for the two variants of Newton's method in the previous question, with that of gradient descent algorithm (without scaling) with backtracking line search (implemented in previous labs), with 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. **[R]** Consider $\eta^k = 1, \forall k = 1, 2, \dots$ in Algorithm 1. With starting point $\mathbf{x}^0 = (8, 8)$ and a stopping tolerance $\tau = 10^{-9}$, find the number of iterations taken by the Newton's method. Compare the number of iterations with that taken by Newton's method (with backtracking line search) in Algorithm 1. Note the minimizer and minimum objective function value in each case. Comment on your observations.
5. **[R]** Compare the number of iterations obtained for the two variants of Newton's method in the previous question, with that of gradient descent algorithm (without scaling) with backtracking line search (implemented in previous labs), with starting point $(8, 8)$. 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.
