Indian Institute of Technology, Bombay — Chemical Engineering
CL603: Optimization — Assignment 6, 2023

**Date Due**: 25 Mar 2023 (Sat) by 11.59 PM (midnight).
**Mode of Submission**: Moodle submission and MATLAB Grader
**Important Instructions:**(i) There is NO concept of LATE SUBMISSION. So, please submit it by the deadline. If you do not submit it by the deadline, then you would not get any credits. (ii) Students working in Matlab should submit assignment (Assignment6_Matlab) on moodle (and also on matlab grader: see instructions below), and (ii) Students working in Python should submit the assignment (Assignment6_Python) on moodle.

**Topic: Trust Region- Powell Dogleg Method**

## Aim

Implement Powell Dogleg Trust Region method for unconstrained minimization of a function.
Use $x_{initguess} = [1.5\ 1.5]^T, N = 15000$ as the maximum number of iterations and $\epsilon = 10^{-6}$ as the tolerance on square of gradient-norm. Thus, the iterations should terminate if either the maximum number of iterations has been reached or if $||\nabla f(x^k)||^2 < \epsilon$.

## Python Code To Submit

A Python script is a standalone file that can contain everything from functions, computations, plotting schemes, etc. This time we will use **Python Grader** for evaluation of your Python Submission. On Moodle specific template is uploaded, and you have to write your logic as per the template.

> **Note**: (1) Do not edit any statement provided on the template.
> (2) Submit a python file named `tut_06_ROLLNO.py where ROLLNO is your roll number`

Your python script should do the following:

1. Implement Powell Dogleg Trust Region method. The various parameters to be used in the trust region approach are: $\bar{\Delta} = 1,\ \Delta_0 = 0.5,\ \eta = 0.2$. Refer to pseudo code for **Trust Region Approach** and the related discussions in the notes to interpret the meaning of the parameters listed above. In this approach $\mathbf{p}^k$ is obtained by **Powell Dogleg method** which either takes a Newton step or a Cauchy step (corresponding to a steepest descent step) or a combination of the two. Once again refer to notes to see the details.

> **Note**: (1) You can call your Gradient and Hessian Function that you have already created in Assignment 2 whenever you need to evaluate the gradient and hessian.
> (2) Consider initial guess of $X$ as value of first iteration ($K = 1$).

- Your code should generate plots as listed below:
  1. Plot $X$ versus iteration number i.e. $x_1$ versus iteration number and $x_2$ versus iteration number in the same figure but as separate subplots.
  2. Generate a figure which shows the value of $f(x)$ versus the iteration number.
     Label the axes and give a title to each figure you generate.
- Your code should also print the final (converged) value of $X$, and the corresponding $f(x)$ and $\nabla f(x)$. In case the iterations do not converge within the specified upper limit of $N$ iterations, your code should print that "Maximum iterations reached but convergence did not happen" and also print the latest values of x, the function, and the gradient.

## MATLAB Code To Submit

- Submit MATLAB code on **MATLAB Grader**. Please find the link for MATLAB grader. `https://grader.mathworks.com/courses/96342-cl-603-optimization` Your code will be auto-graded on MATLAB grader. A template file is available on MATLAB grader. Write your code in that template file. You can run the code to check its correctness/outputs. After you are satisfied, then submit the code. Important: You will be able to submit the code ONLY ONCE. After submission, you will be able to see your marks and errors (if any), but will not be able to modify your code. Thus, submit only after you are satisfied with your code. Submit before the deadline.

- Also submit all your MATLAB files (you can zip them) on Moodle for our records. Grading will however be on MATLAB grader.

## Test functions for optimization

- Four test functions with an optimum point ($X^*$), optimum function value $F(X^*)$, and converged value of gradient $\nabla F(x^*)$ are given in (Table:1) to check the correctness of your code.

Table 1: Test function for Powell Dogleg Trust Region method

| S.no | Test function | Optimum point($X^*$) | $F(X^*)$ | $\nabla F(x^*)$ |
|---|---|---|---|---|
| 1 | $f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 - x_1 x_2$ | $\begin{bmatrix} 2.0000 \\ 2.0000 \end{bmatrix}$ | -2 | $\begin{bmatrix} 0.7705 \\ 0.7705 \end{bmatrix} \times 10^{-10}$ |
| 2 | $f(x_1, x_2) = x_1^2 + x_2^2 + (0.5x_1 + x_2)^2 + (0.5x_1 + x_2)^4$ | $\begin{bmatrix} 0.0869 \\ 0.1715 \end{bmatrix} \times 10^{-5}$ | 0 | $\begin{bmatrix} 0.3887 \\ 0.7730 \end{bmatrix} \times 10^{-5}$ |
| 3 | $f(x_1, x_2) = -0.0001(\lvert sin(x_1)sin(x_2)exp(\lvert 100 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\rvert)\rvert)^{0.1}$ | $\begin{bmatrix} 1.3494 \\ 1.3494 \end{bmatrix}$ | -2.0626 | $\begin{bmatrix} -0.6115 \\ -0.6115 \end{bmatrix} \times 10^{-6}$ |
| 4 | $f(x_1, x_2) = \lvert x_1\rvert^2 + \lvert x_2\rvert^3$ | $\begin{bmatrix} 0 \\ 0.0120 \end{bmatrix}$ | $1.7098 \times 10^{-6}$ | $\begin{bmatrix} 0.0000 \\ 0.4300 \end{bmatrix} \times 10^{-3}$ |

## Learning

You will learn the following by completing this assignment,

- Implementation of Powell Dogleg Trust Region methods for minimizing a function.

---

Learning is fun. Best of Luck!