Indian Institute of Technology, Bombay — Chemical Engineering
CL603: Optimization — Assignment 2, 2023
Date Due: 21 Jan 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 by the deadline. If you do not submit by the deadline, then you would not get any credits. (ii) Students working in Matlab should submit assignment (Assignment2_Matlab) on moodle (and also on matlab grader: see instructions below), and (ii) Students working in Python should submit assignment (Assignment2_Python) on moodle.

Topic: Gradient and Hessian Computation

Part 1: Gradient Computation

Aim: To understand and be able to write a function in Python for computing the gradient of any function.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a scalar function. The gradient $\mathbf{g}$ of function f is a vector of size $n \times 1$. It is defined as follows:

$$\mathbf{g} = \nabla f = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{bmatrix} \tag{1}$$

Let $\mathbf{x}^\star$ be the value of $\mathbf{x}$ at which the gradient needs to be computed. The partial derivatives in the gradient can be computed numerically using central difference scheme as:

$$\left.\frac{\partial f}{\partial x_j}\right|_{\mathbf{x}=\mathbf{x}^\star} \approx \frac{f(\mathbf{x}^\star + h\mathbf{e_j}) - f(\mathbf{x}^\star - h\mathbf{e_j})}{2h} \tag{2}$$

$$\text{where,} \qquad \mathbf{e_j} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T \tag{3}$$

In Eq.2, the derivative is approximated by perturbing the j^{th} variable. Hence, vector $\mathbf{e_j}$ in Eq.3 has element 1 at j^{th} position and 0s elsewhere. The perturbation h in Eq. 2 is a small number.

Part 2: Hessian Computation

Aim: To understand and be able to write a function in Python for computing the hessian of any function.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a scalar function. The hessian $\mathbf{h}$ of function f is a matrix of size $n \times n$. It is defined as follows:

$$\mathbf{h} = \nabla^2 f = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} \cdots \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} \cdots \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{4}$$

Let $\mathbf{x}^\star$ be the value of $\mathbf{x}$ at which the hessian needs to be computed. The partial derivatives in the hessian can be computed numerically using central difference scheme as:

$$\frac{\partial^2 f(x_1, x_2, ..., x_n)}{\partial x_i^2} = \frac{1}{(\Delta x_i)^2} \{f(x_1, x_2, ..., x_{i-1}, x_i + \Delta x_i, x_{i+1}, ..., x_n) \\ -2f(x_1, x_2, ..., x_n) + f(x_1, x_2, ..., x_{i-1}, x_i - \Delta x_i, x_{i+1}, ..., x_n)\}$$

$$\frac{\partial^2 f(x_1, x_2, ..., x_n)}{\partial x_i \partial x_j} = \frac{1}{(4\Delta x_i \Delta x_j)}(A + B - C - D) \tag{5}$$

where,

$$A = f(x_1, ..., x_{i-1}, x_i + \Delta x_i, x_{i+1}, ..., x_{j-1}, x_j + \Delta x_j, x_{j+1}, ..., x_n) \tag{6}$$
$$B = f(x_1, ..., x_{i-1}, x_i - \Delta x_i, x_{i+1}, ..., x_{j-1}, x_j - \Delta x_j, x_{j+1}, ..., x_n) \tag{7}$$
$$C = f(x_1, ..., x_{i-1}, x_i - \Delta x_i, x_{i+1}, ..., x_{j-1}, x_j + \Delta x_j, x_{j+1}, ..., x_n) \tag{8}$$
$$D = f(x_1, ..., x_{i-1}, x_i + \Delta x_i, x_{i+1}, ..., x_{j-1}, x_j - \Delta x_j, x_{j+1}, ..., x_n) \tag{9}$$

where $1 \leq i, j \leq n$. While writing the above expression, it is assumed that $i < j$. $\Delta x_i, \Delta x_j$ are small perturbations in x_i, x_j respectively.

Python Code To Submit

A Python script is a standalone file which can contain everything from functions, computations, plotting schemes, etc. Your code may be in the following structure,

- **Part A**: For the current file, only NumPy library (**Num**erical **Py**thon) is necessary for the current tutorial.
- **Part B:** Three functions are necessary to achieve the aim of the assignment,
  1. Write a function called "`func`" which should contain the given (see below) function of two variables..
  2. Write a function called "`compute_hessian`" used to evaluate the approximate hessian of the input function at a point $\mathbf{x}^\star$ using the central difference method.
  3. Write a function called "`compute_gradient`" used to evaluate the approximate derivate of the input function at a point $\mathbf{x}^\star$ using the central difference method.
- **Part C:** Using the functions to obtain the Gradient and Hessian matrix,

```
delF = compute_gradient(func, x_input)

del2F = compute_hessian(func, x_input)
```

Submit a python file named `tut_02_ROLLNO.py where ROLLNO is your roll number`. Use the following function whose Hessian and Gradient are to be calculated,

$$f(\mathbf{x}) = 2e^{x_1}x_2 + 3x_1x_2^2 \tag{10}$$

at a point $\mathbf{x}^\star$. Take $h = 0.001$ and $\Delta x_1 = \Delta x_2 = 0.001$. The $\mathbf{x}^\star$ value will be specified by us while evaluating your submission.

Note: To check your code, you can take some $\mathbf{x}^*$, say $\mathbf{x}^* = [1\ \ 1]^T$ and execute your code. Compare the function value, gradient, and hessian computed by your functions with the analytical values to check the correctness of your results.

MATLAB Code To Submit

- Submit MATLAB code on **MATLAB Grader**. Please find 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.

Learning

You will learn the following by completing this assignment,

- Creating your own function.
- Calling the functions, passing inputs to them and obtaining outputs.
- Numerical Gradient and Hessian computation.

Learning is fun. Best of Luck!