CS101 Advanced Engineering Mathematics (I)

工程數學(一)

[Guidelines]

- All the homework in this course will involve solving advanced engineering mathematics problems (differential equations in particular) by hand and computer.
- While discussion with other classmates is allowed, you MUST work independently to generate your own solutions to the problems.
- Python programming will be used for plotting solutions. You should reference the Python Tutorial (課程講義) for detail information.
- For each homework, you must submit a ***written report*** **(書面報告)**.

[General Instructions]

To get a good grading in homework assignments, you are advised to do the following:

- Do not copy other classmate's works! (請遵守學術倫理，嚴禁抄襲)
- Provide correct answers in details. (詳細推導過程與標明正確答案)
- Prepare your written reports in good quality (使用 Template 檔並書寫工整).
- Meet the deadline! Late homework will **not** be collected. (按時繳交，逾時不候)

指導教授：張元翔

Homework Assignment 1

Review of Calculus & First-Order Differential Equations

Deadline: 11 / 10 / 2023 (星期五)

(請繳交書面報告至電學 603 計算機視覺研究室)

[Instructions]

Calculus is useful for solving optimization problems. In this homework assignment, our goal is to learn the ***Method of Least Squares*** (最小平方法), also known as the ***Linear Regression*** (線性迴歸).

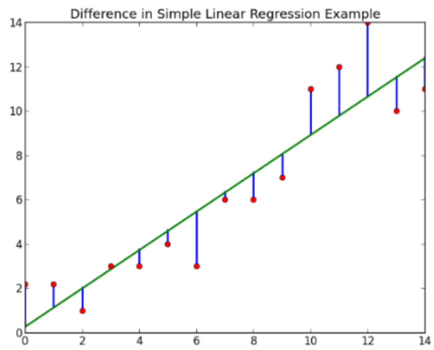
[Problems]

1. 給定一組資料點 (x_i, y_i), $i = 1 \ldots n$，**最小平方法** (Method of Least Squares) 的目的是找一直線 $y = ax + b$，使得每一點至直線的垂直距離總和 (又稱為**平方誤差和** Sum of Square Errors) 可以達到最小值：

$$\varepsilon = \sum_{i=1}^{n} \left[y_i - ax_i - b \right]^2$$

假設給定一組資料點 (2, 1)、(3, 2)、(4, 3)、(5, 2)，試利用最小平方法求得最佳之直線方程式 (10%，請列出手寫推導過程即可)

【提示】 分別**設微分為 0**，即 $\dfrac{\partial \varepsilon}{\partial a} = 0$ 與 $\dfrac{\partial \varepsilon}{\partial b} = 0$ 。



[Instructions]

The ***Gradient Descent method*** & ***Newton's method*** are useful for solving the optimization problems. In this homework assignment, your goal is to design the Python program that applies the two methods.

The iteration equation using the ***Gradient Descent method*** can be defined as:

$$x_{n+1} = x_n - \alpha f'(x_n)$$

where x_n is the *n*th iteration and α is the learning rate.

The iteration equation using the ***Newton's method*** can be defined as:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Please reference the Python Tutorial (課程講義) for details.

[Problems]

2. Suppose the function can be defined as: $y = f(x) = x^2 - 2x + 10$, please answer the following:
   (a) Design the Python program using the ***Gradient Descent*** method and print the values of $(x_n, y_n), n = 0...10$ (i.e., the first 10 iterations) given the initial guess $x_0 = 0$ and $\alpha = 0.1$. (5%)
   (b) Design the Python program using the ***Newton's method*** and print the values of $(x_n, y_n), n = 0...10$ (i.e., the first 10 iterations) given the initial guess $x_0 = 0$. (5%)
   (c) Plot the function and show $(x_n, y_n), n = 0...10$ in the ***same figure*** (For plotting, *x* is in the range of –0.5 ~ 2.5). As a result, you will generate two plots for the two methods. (10%)

Note: The figures must be carefully ***labeled***, ***titled***, and with your own ***copyright*** for full credits. 原則上，同學在跑完程式後，可將繪圖存成圖檔，然後在 Word 檔中插入圖片。

[Instructions]

Direction fields are particularly useful for solving first-order differential equations when analytic solutions can't be found. To plot a direction field for a first-order differential equation with Python programming, the equation must be in **normal form**. A first-order differential equation is in normal form if it is expressed as:

$$\frac{dy}{dx} = f(x, y)$$

Please reference the Python Tutorial (課程講義) for details.

[Problems]

3. Following the aforementioned instructions, use the Python programming to obtain the direction field for each of the following differential equations (the interval I is given for (x, y) coordinates accordingly). The figures must be carefully ***labeled***, ***titled***, and with your own ***copyright*** for full credits. (20%)

(a) $\frac{dy}{dx} = x + y$; I: [–5 : 0.5 : 5, –5 : 0.5 : 5]

(b) $\frac{dy}{dx} = x - y$; I: [–5 : 0.5 : 5, –5 : 0.5 : 5]

(c) $\frac{dy}{dx} = xy$; I: [–5 : 0.5 : 5, –5 : 0.5 : 5]

(d) $\frac{dy}{dx} = \sin x \cos y$; I: [–5 : 0.5 : 5, –5 : 0.5 : 5]

[Instructions]

Python is useful for generating 2D plots using Numpy and Matplotlib. If an *explicit* solution can be found for a differential equation, you may plot the solution curve using the Python programming. Please reference the Python Tutorial (課程講義) for details.

[Problems]

4. Solve the following initial value problems, and plot the solution curves. The interval I is given for the x-data in the plots. The figures must be carefully ***labeled***, ***titled***, and with your own ***copyright*** for full credits.

   注意：請先用手寫推導解題，再用 Python 畫圖，手寫推導過程與圖須放在同一頁面 (20% 每題 5 分)

   (a) $\dfrac{dy}{dx} = -xy,\ y(0) = 1, \quad I: [-3, 3]$

   (b) $\dfrac{dy}{dx} = (1 + x)(1 + y^2), \quad I: [0, 5]$

   (c) $\dfrac{dy}{dx} - (\cot x)y = \sin x,\ y(0) = 1, \quad I: [0, 4\pi]$

   (d) $\dfrac{dy}{dx} = (x + y + 1)^2,\ y(0) = -1, \quad I: [0, 5]$

[Instructions]

For many differential equations, *explicit* solutions may not be found. Instead, we may find the *implicit* solutions only. In these cases, we plot the 3D plot (surface plot) and level curves (contours) as the implicit solution curves. Please reference the Python Tutorial (課程講義) for details.

[Problems]

5. Solve the following initial value problems, and plot the solution curves. The interval I is given for the xy-data in the plots. The figures must be carefully ***labeled***, ***titled***, and with your own ***copyright*** for full credits.

   注意：請先用手寫推導解題，再用 Python 畫圖，手寫推導過程與圖須放在同一頁面 (20% 每題 5 分)

   (a) $(y^2 + 4y)dx + (2xy + 4x)dy = 0$, $I: [-2 \sim 2, -2 \sim 2]$

   (b) $(2x + e^y)dx + xe^y dy = 0$, $I: [0 \sim 2, 0 \sim 2]$

   (c) $\cos x dx + \left(1 + \frac{2}{y}\right)\sin x dy = 0$, $I: [0 \sim 2\pi, 0 \sim 2\pi]$

   (d) $(\sin y - y\sin x)dx + (\cos x + x\cos y - y)dy = 0$, $I: [-2\pi \sim 2\pi, -2\pi \sim 2\pi]$