Numerical Method

National Cheng Kung University Department of Engineering Science

Instructor: Chi-Hua Yu

HW 5

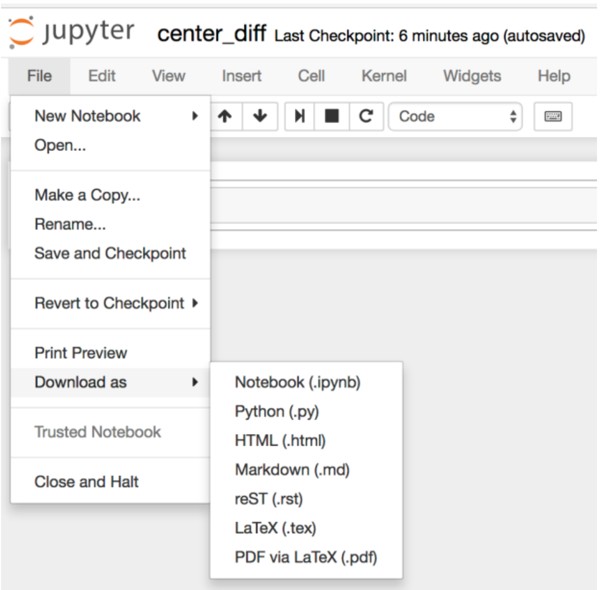
Programming, Due 09:00, Wednesday, April 13th , 2022

注意事項：

1. Homework 的時間為公布題目後至下次上課前結束(上課當天 09:00)。
2. 請在規定的時段內完成作業，並用你的學號與 HW number 做一個檔案夾 (e.g., N96091350_HW3), 將你的全部 ipynb 檔放入檔案夾，壓縮後上傳至課程網站 (e.g., N96091350_HW3.zip)，超過期限後不予補交。

Homework Submission Procedure (請仔細閱讀)

1. You should submit your Jupyter notebook and Python script (`*.py`, in Jupyter, click File, Download as, Python (`*.py`)).



2. Name a folder using your student id and lab number (e.g., n96081494_HW1), put all the pdf and all the Jupyter notebooks and python scripts into the folder and zip the folder (e.g., n96081494_HW1.zip).

3. Submit your lab directly through the course website.

1. **(50%)** Name your pdf file HW5_student id (e.g., HW5_n96081494.pdf). Please state the details and derivations in a professional format and submit a pdf file to Moodle. Given a real matrix

$$\boldsymbol{A} = \begin{bmatrix} 2 & 0 & 1 \\ -1 & 4 & -1 \\ -1 & 2 & 0 \end{bmatrix},$$

(a) What is the rank of **A?**

(b) Write down the eigenvalues

(c) Write down the eigenvectors with respect to each eigenvalue.

2. (50%) Name your Jupyter notebook `gauss_seidel.ipynb` and Python script `gauss_seidel.py`. Write a Python program to solve the equations by using the Gauss–Seidel method.

$$Ax = y$$

$$\begin{bmatrix} a_{1.1} & a_{1.2} & \cdots & a_{1.n} \\ a_{2.1} & a_{2.2} & \cdots & a_{2.n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m.1} & a_{m.2} & \cdots & a_{m.n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

We can assume all values of $x^{(0)}$ as 0 and using following the equation to substitute $x^{(1)}$ in the first iteration.

$$x_i = \frac{1}{a_{i,i}} \left[y_i - \sum_{j=1, j\neq 1}^{j=n} a_{i,j} x_j \right]$$

After obtaining $x^{(1)}$, we continue to iterate until the difference between $x^{(k)}$ and $x^{(k-1)}$ is smaller than a predefined threshold ε = 0.0001 or the maximum iterations is reached.

Below is the running example

Sample 1

```
a = np.array([[8, 4, -3], [-2, -8, 5], [3, 5, 10]])
y = np.array([14, 5, -8])


Iteration results
 k,    x1,      x2,       x3
 1   1.7500   -1.0625   -0.7938
 2   1.9836   -1.6170   -0.5866
 3   2.3385   -1.5762   -0.7134
 4   2.2706   -1.6385   -0.6619
 5   2.3211   -1.6190   -0.6868
 6   2.3019   -1.6298   -0.6757
 7   2.3115   -1.6252   -0.6809
```

```
 8   2.3073   -1.6274   -0.6785
 9   2.3092   -1.6264   -0.6796
10   2.3083   -1.6268   -0.6791
11   2.3088   -1.6266   -0.6793
12   2.3086   -1.6267   -0.6792
13   2.3087   -1.6267   -0.6793
14   2.3086   -1.6267   -0.6792
Converged!
```

Sample 2

```
a = np.array([[12, 3, -5], [1, 5, 3], [3, 7, 13]])
y = np.array([10, 6, 3])


Iteration results
 k,    x1,      x2,      x3
 1   0.8333   1.0333   -0.5179
 2   0.3592   1.4389   -0.6269
 3   0.2124   1.5337   -0.6441
 4   0.1816   1.5501   -0.6458
 5   0.1767   1.5521   -0.6458
 6   0.1762   1.5522   -0.6457
 7   0.1762   1.5522   -0.6457
Converged!
```