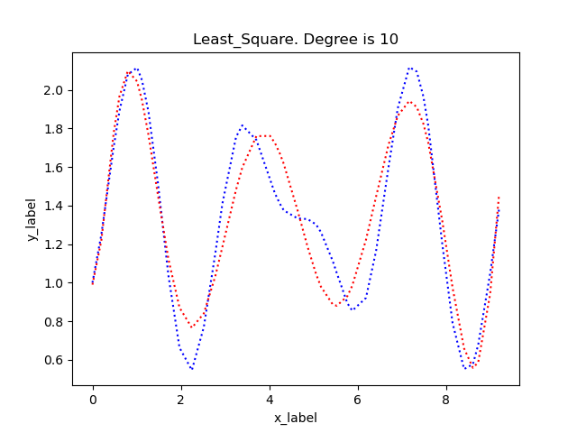
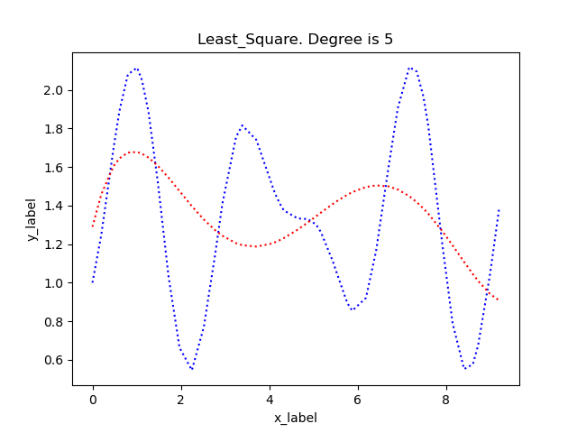
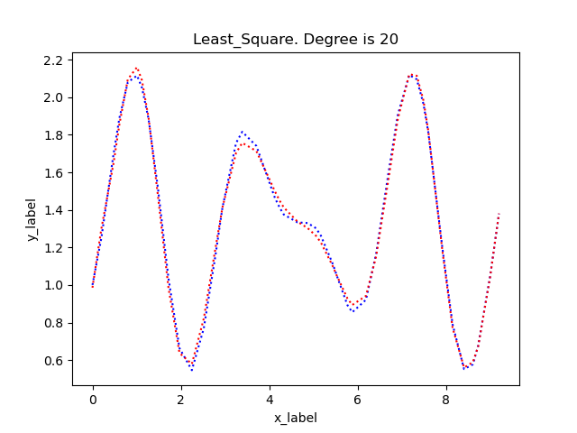
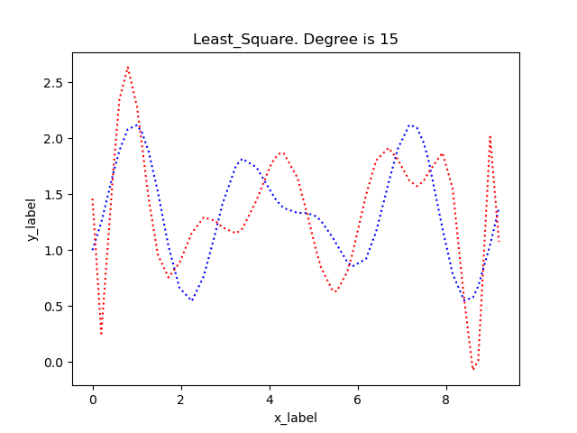
**數值方法HW4**

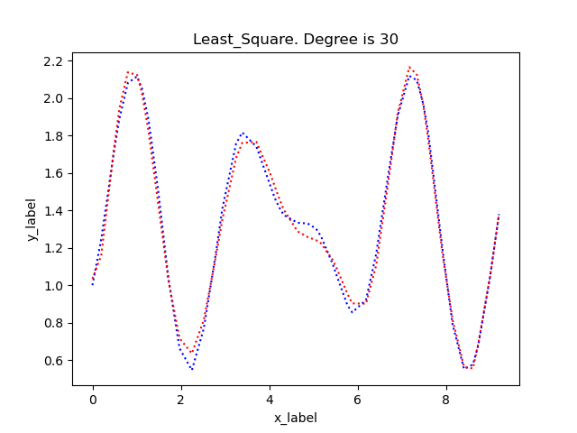
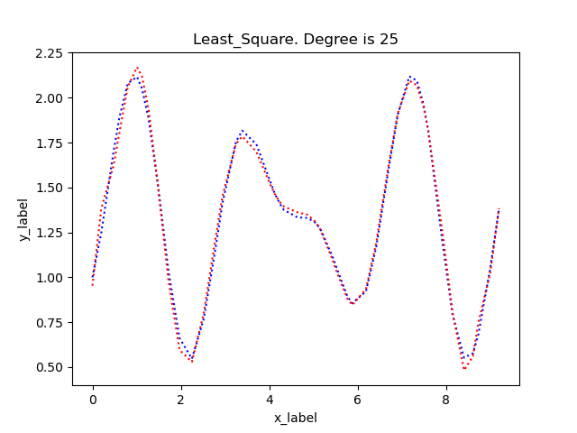
406262292資工三乙 梁鈞韋

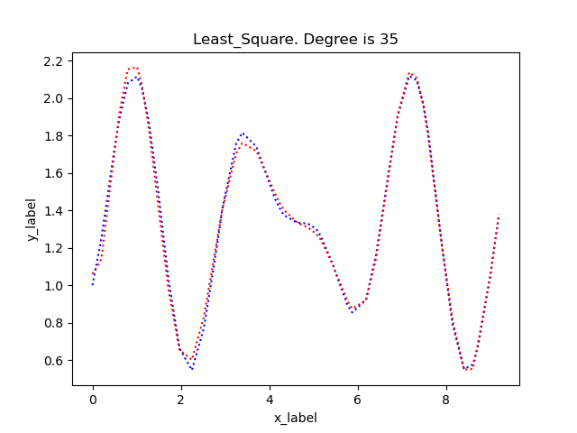
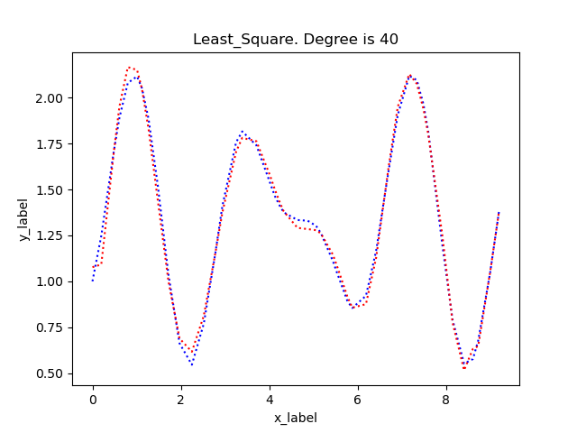
2020/06/27完成

**圖形**







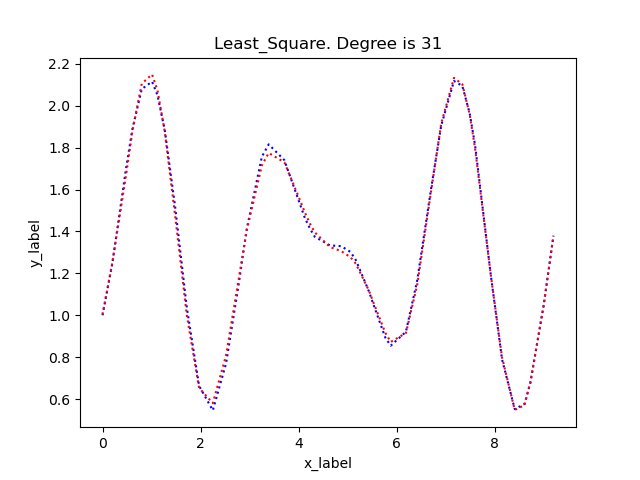
**畫圖形的方式**

Python套件matplotlib, 就是好用!, 藍線的部分是原本的f(x),紅色部分是後來寫出來的P(x)

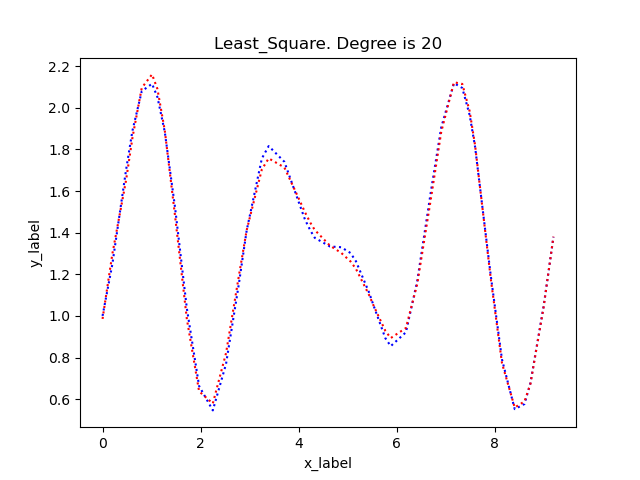
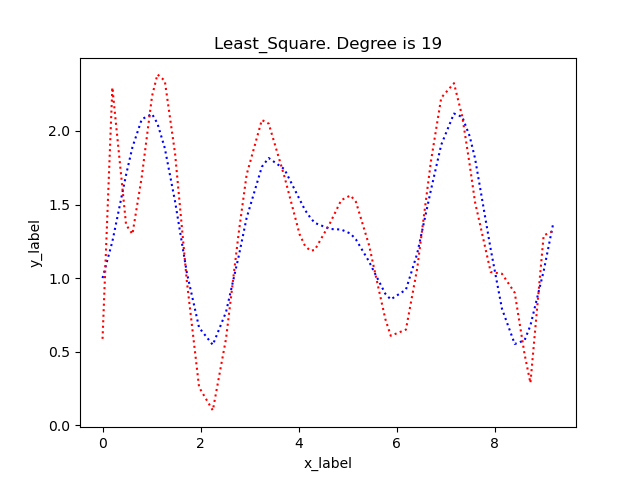
**程式架構**

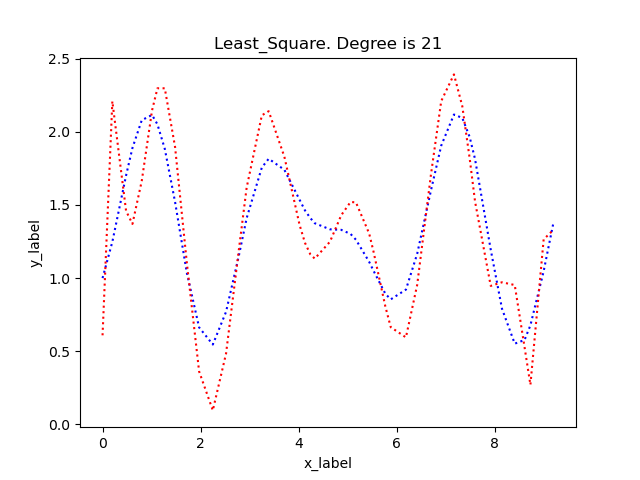
Picture資料夾下有各個degree的實驗照片, 執行least\_square.py即可知道the best choice是哪個degree的polynomial

**分析**



以往都用c++寫作業的我, 後來受不了matrix的計算, 所以這次改用python, 算p(x)那條方程式的時候, 把I,j搞混, 導致debug花了一些時間, 實驗發現在Degree is 31的時候, P(x)跟f(x)的圖形誤差最小, 再算error標準差的時候, 要小心分母等於0的情形。





實驗發現假如在degree是19跟21時, P(x)的曲線變得非常的不科學, 整個亂偏掉。