B06507002 材料一 林柏勳 實作一結報

第一題，題目要求要解出泰勒展開式，取某展開點，展開n項，看它對於e^2誤差的影響。

所以，我先引進sympy，定義x這個符號，順便代入diff(微分)，再引進math，定義e^x。

接著，我定義sum=0以利後續求值運算。

我用一個for迴圈去尋找誤差小於1\*10^(-10)時的N值，首先，我在for裡面包一個numerator=exp.diff(x,i)，就是對e^(x)微分i次，再來，我選擇對x=0這點展開，用物件取向的特性，改變numerator，numerator=numerator.evalf(subs={x:0})，代入x=0這點，並用evalf進行數值的運算。接下來，我要處理階乘，定義

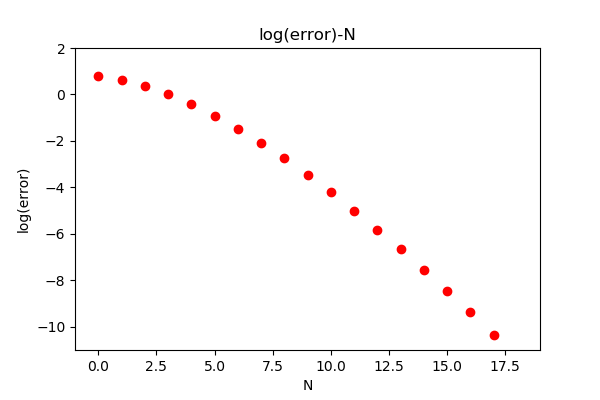
denominator = np.math.factorial(i)，這樣就可以寫出i=多少時的階乘值。

最後寫sums += numerator/denominator\*x\*\*i這就是泰勒展開式在0展開的型態。

而題目問對e^2的誤差，所以我將值(x=2)代入此展開式sums.subs({x:2}。並將估計值和真實值相減後取絕對值，使用if函式輸出在N=多少時，誤差小於1\*10^(-10)，輸出後就停止(break)。

得到在N=17後時，e^(2)用e^(x)在0展開的誤差值小於1\*^(-10)。

使用matplotlib.pyplot將本題目的結果，以log(error)對N做圖出來，如下:



圖中可以看出，N越大的時候，誤差越小。

使用matplotlib的方法在此說明一次，後面不再贅述。

首先，將此模組引進，import matplotlib.pyplot as plt

將前面的資料用for迴圈寫進list中，也就是將log(error)用for寫進一個list(名稱為logy)，用list(range(len(logy))將N寫進另一個list之內。

用plt.plot將將前面list的x,y值寫入，用plt.axis設定x,y的輸出範圍，用plt.xlabel、plt.ylabel、plt.title設定x,y以及圖表名稱，

用fig1 = plt.gcf() 準備截圖，

用plt.show()展示圖片

用fig1.savefig('test.png', dpi=100)儲存這張圖片。

第二題，題目要求sin(x)微分兩次的值，分別用一般的二次微分，以及助教給的近似解。

所以，我先進行一般二次微分的運算，我引進sympy定義符號x，定義目標函數goal(N)，

包含a= diff(sin(x),x,x)，對sin微分兩次的值，而h=pi/N，我又設定一個空list b=[]，再用for將h,2h,……(N-1)h分別代入a中，最後append進b中，完成真實值的定義。

接著，我要處理近似值的部分，我主要以多個定義疊加出最後的結果。

首先，定義近似值函數

df2(u,h,x = sympy.Symbol('x')):

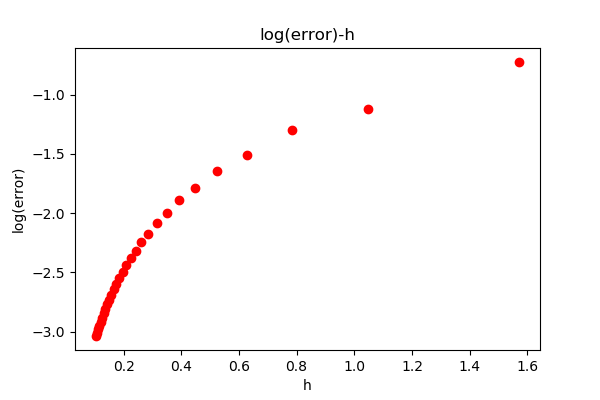
return (u(x+h)-2\*u(x)+u(x-h))/h\*\*2

u是近似解的函數，h在之後設定為pi/N

再定義函數app(f,N)，設定空list a=[]，用for 把df2的函數，將h,2h,3h,……(N-1)h的值代入，並append進a中，最後return a，保持此函數會輸出list。

最後定義app2(N)，裡面再定義一層函數f1(x)，return sin(x)，最後將app(f,N)裡面的f設定為f1，完成近似值函數的定義。

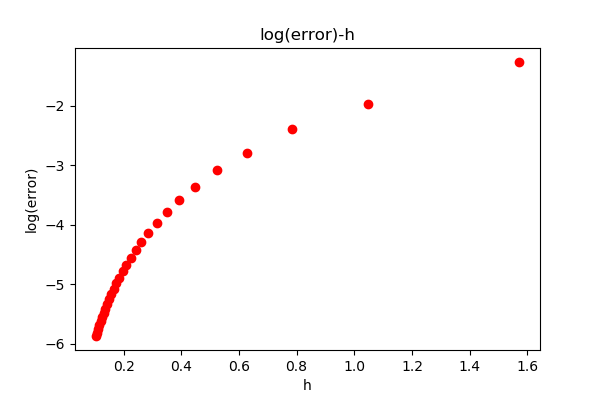
最後，定義error(N)，將前面寫好的app2(N)，goal(N)代入，先寫一個空list c=[]，再使用for將app2(N)跟goal(N)的每一個項相減，寫進c中，最後求出c中的最大值，這就是欲求的error。

用前述方法，將log(error)和h(N)分別寫成兩個list，以matplotlib將之繪出，h的範圍是N=2~30時的h，而圖中可以看出，h越小的時候，誤差越小。

第三題的做法和第二題類似，主要是把公式換成

(-u(x+2\*h)+16\*u(x+h)-30\*u(x)+16\*u(x-h)-u(x-2\*h))/h\*\*2/12

，其餘定義函數稍作命名上的修正就可以輸出結果了。



對比上下兩圖可以看出:下圖的誤差在相同h情況之下，誤差顯得比較小，所以第三題的公式可以提供一個比較好的近似。

額外的數據分析

因為我想知道這條曲線大概是甚麼的方程式，所以我題目三的數據又用pandas處理了。

首先，我將h跟log(error)的資料轉成np.array的格式

y=np.array(errorlist2(30))、x=np.array(h2(30))

(註:errorlist2(30)、h2(30)分別為N=30時，log(error)和h的list)

，再把它轉成dataframe的格式，df = pd.DataFrame({'logerror':y,'h':x})

因為這兩個dataframe的格式目前為object，我使用.astype(‘float’)把他們兩個轉成浮點數，以利進行後續分析。

接著，我執行from scipy.optimize import curve\_fit，引進這個模組，我猜測這個函數是指數函數，所以把它定義成

func(x, a, b, c):

return a \* np.exp(-b \* x) + c ，用它去擬合。

接下來執行擬合的步驟popt, pcov = curve\_fit(func, x, y)

(註:Popt對應的是array、pcov對應的是 2d array)

最後得到係數，轉換成方程式約為:y=-5.467e^(-2.347)-1.374。最後繪圖如下:

可以粗略發現，這個模型在h很小的情況下，擬合程度很高。

