Question1

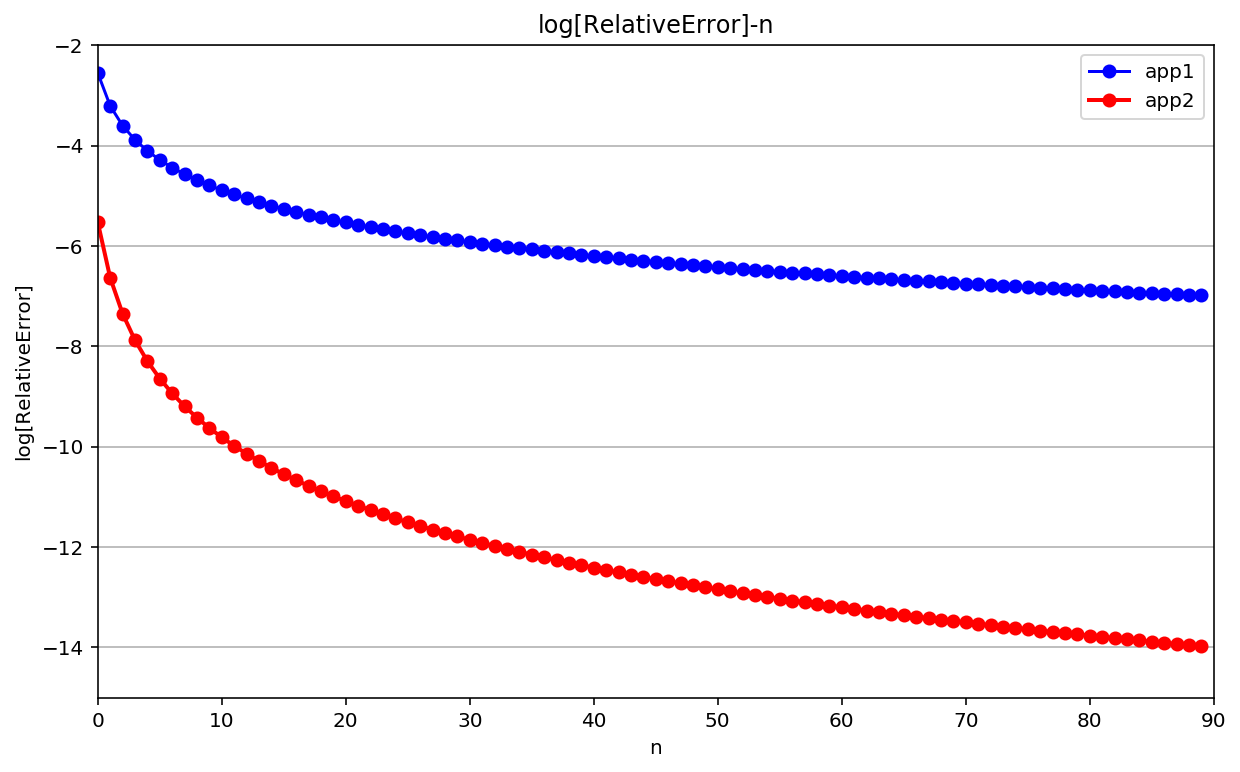
首先，從numpy 引進 exp,log,pi

接著，以n為自變數，定義app1、app2、r(真實值)，

app1為、app2為、r為n!

我進行相對誤差的運算後，以app1l為app1 log[RelativeError]的list，範圍從1到90；以app2l為app2 log[RelativeError]的list，範圍從1到90。

然後，我要進行繪圖，設定好圖名、x範圍從n=0到n=90、y範圍從-14到-2、設定app1的log[RelativeError]為藍色的線，設定app2的log[RelativeError]為紅色的線，最後，把圖例放到右上角，完成繪圖如下。

由本圖可以看出，app1的誤差小於app2的誤差，所以app2是一個對階乘函數比較好的近似。

接下來，要分別使用梯形法和辛普森法，去尋找gamma function積分的近似解，所以，我定義gf=做為要被積分的函數，之後再想辦法將之切成n等分，從0積分到b。

我定義梯形法的函式Tn，參數為n,b，n為總共切成幾等分，b為積分的右端點。首先，設定f為要被積分的函數:f=gf，以n建立各個取點的lista，num=n+1的原因為有n+1個點。接著，設定初始值a=0以待之後進行運算用。我先用for算中間的值，並加入a中，再把左右端點的值加入a中，最後再將a值return，完成梯形求值法的運算。

再來，說明辛普森法的撰寫過程，我也定義辛普森法的涵式Sr，參數為n,b。因為辛普森法的n被限制只能是偶數，所以，我在最外面放了一層if，

如果n mod 2 =0的話，才會進行運算，其他為奇數的情況則return None。接著，我也建立各個取點的lista，設定a=0，為初始值。為了方便，我設定

，然後先計算不含兩端點的函數值，寫一個for，使用兩個if篩選奇數和偶數，若該點在lista中的編號為奇數: x1,x3,x5,......,xn-1，則在a中加入f(lista[i])\*c\*4；若該點在lista中的編號為偶數: x2,x4,x6,......,xn-2，則在a中加入f(lista[i])\*c\*2。最後，將端點值加入a中，return a，完成辛普森法的求值。

接下來則要說明如何找到最小的b，再來尋找與之相應的n。我設定found這個函式，參數為f ，f是要帶入的積分方法:梯形法、辛普森法。我先建立一個空的list c。接著，我疊了兩層for，第一層for i in range(1,101)，設定b的範圍；第二層for j in range(1,201)，設定n 的範圍，如果該運算方法的值會等於真實值20!的話，我會把[n,b]的值加入c中，最後按照list的編號取最小的b，及其對應的n，即為本題的答案。我得到的答案在梯形法為n=70、b=80，在辛普森法為n=128、b=79。接著要計算跑了幾步，按照梯形法來說，就是(b-1)\*200+70，依照辛普森法來說，就是100\*(b-1)+int(n/2)+n%2(在不考慮n為奇數的情況下所走的步數，也就是return None的情況下不計入步數)。我得出結果梯形法跑了15870步，辛普森法跑了7864步，就結果看來，辛普森法是一個較為快速的方法。

Question 2

本題要使用pandas讀取資料，再以matplotlib.pyplot進行繪圖。首先我設定path為路徑，再指定df為讀取路徑後的格式，接著，我用df.columns讀取各個欄位的標題，使用先前獨到的標題設定A=Contact Area (nm^2)、P=Indent Load (µN)、c0=24.5、h=Contact Depth (nm)、H=Hardness (GPa)等行的資料。接著就設定標題、x、y軸標題、輸入要畫的資料(H、、、-)，即完成作圖，不過在、的部分要乘以1000，才能使單位一致。最後，要把四張圖合併，我使用fig = plt.figure()的方法，它可以輸入左下角的網格點以利擺放圖表，也可以設定圖片的大小，至於顏色和標題則自然可以設定。以繪製H為例，我輸入的函式為fig.add\_axes([0.1,0.5,0.3,0.3])，(0.1,0.5)為此圖左下角的網格點，該圖為的大小，最後依此方法將四張圖都設定好，即完成本次的作業。

