Python Week 5

• Review hw4

– /home/teachers/fortran_ta/data/PSC2020/hw4/

- Q&A:plt.contour
- 程式設計技巧-加速、How to Debug?
- HW5

Map of Contour

Mount Fuji's contour lines



https://en.wikipedia.org/wiki/Contour_line https://gisgeography.com/contour-lines-topographic-map/

Map of Precipitation Contour (or pressure, temperature, ...)



Map of Contour

- 1. choose a **contour interval** (e.g., interval of 10)
- 2. Add markers at transition points of ten (30, 40, 50, etc) by interpolation



https://gisgeography.com/contour-lines-topographic-map/

Map of Contour

3. Connect the interpolated markers with smooth contour lines for each interval



https://gisgeography.com/contour-lines-topographic-map/

Q&A: 2D plot (plt.contour)

- Prepare data x軸位置的陣列 (2D) y軸位置的陣列 (2D) 要畫的資料陣列 (2D) (也可以是從資料檔案讀進來)
 Prepare data x=np.linspace(-1,1,25) y=np.linspace(-1,1,25)
 xx,yy=np.meshgrid(x,y)
 z=np.exp(-np.sin(2*xx)**2np.cos(2*yy)**2)-0.5
- 畫出等值線,設定等值線數值、設定等值線顏色
 CS = plt.contourf(x, y, z, levels= np.linspace(-0.5,0.5,11),cmap=cm.jet)
 CS2 = plt.contour(x, y, z, levels=np.linspace(-0.5,0.5,11),colors='r')
- 加上color bar

plt.colorbar(CS,orientation='vertical')

- 設定座標軸、文字...
- 存檔、顯示

程式設計技巧-加速

運算速度比較(1)

- 測試「一維陣列內積」: c = a[0]*b[0]+a[1]*b[1]+a[2]*b[2]+...
 Numpy內建函數 vs. 簡單迴圈
- /home/teachers/weitingc/lecture_ex/innerproduct.py
 start1=time.time() <-- 記錄程式執行到這行的時間

Numpy內建 c1=np.dot(a,b) (0.2



簡單迴圈 for i in range(n): c2=c2+a[i]*b[i] (81.8 s)



運算速度比較(2)

- 測試「條件搜尋」: 找出array a有幾個元素>0.5
 Numpy whole array mask vs. 簡單迴圈+if
- /home/teachers/weitingc/lecture_ex/search.py

Numpy mask c1=np.sum(a>0.5) (0.2 s)



簡單迴圈 for i in range(n): (50.1 s) if a[i]>0.5 c2=c2+1

利用 迴圈+if 計算



利用 whole array mask 取代x-y的迴圈計算

```
def calPBL_theta05_layer(theta, qv, zc):
    th_pbl = theta[0,:,:]+LV
    z_pbl = -1*np.ones((1024,1024), dtype='f')
    theta_xm = np.ma.array(theta, mask= (qv == 0))
    for z in range(70):
        theta_z = np.array(theta_xm[z, :, :])
        aa = (theta_z == th_pbl) & (z_pbl==-1)
        z_pbl[aa] = zc[z]
        bb = (theta_z>th_pbl) & (z_pbl==-1)
        z1, z2 = zc[(z-1, z),]
        th1, th2 = np.array(theta[(z-1,z), :, :])
        z_pbl[bb] = z1 + (z2 - z1) * (th_pbl[bb]- th1[bb]) / (th2[bb] - th1[bb])
    return(z_pbl)
```

程式設計技巧 -- Debug

工程師通常花10%的時間寫程式,90%的時間抓蟲

- 程式語言常見的錯誤種類:
 - 語法錯誤(例如:a=[1,2,3<mark>;</mark>4,5] , if (a>0)<mark></mark>)
 - 邏輯錯誤(例如: if ((a>0) and (b<0)):... 打成if ((a>0) or (b<0)):
 - 執行階段錯誤(例如: a=100/b, 當b=0時a=Inf)
- 經常活在python程式中的「蟲」:
 - 變數名稱大小寫混淆(例如 Count vs count)
 - 輸入函數或指令的變數類型錯誤(例如 plt.xlabel(3x2), np.exp('a'))
 - 純量變數與陣列變數混淆,或弄錯陣列的大小(或維度)
 邏輯判斷相等時,誤用=(正確語法應該是 if a==0).
 沒有先import需要的工具庫
 - 迴圈、邏輯判斷、自訂函數沒有正確縮排

一般除錯技巧(Debugging)

- 秘訣零:避免bug產生
 - 程式不要一口氣寫到尾,寫一小段就進行測試,正確再繼續往下- 好好面對錯誤訊息
- 秘訣一: 在程式中間加入print指令, 顯示變數執行到此處的 結果, 檢查是否如預期
- 秘訣二:將某段程式用#暫時變成註解,觀察程式執行的情況 (可以搭配print指令)
- 秘訣三:debug自訂函數時,用#把第一列的函數宣告列(def xxx(...))暫時註解掉,加入臨時的程式碼直接設定要從主程式接收的變數值,

- 此時就可以獨立執行這個自訂函數(即不需要透過主程式呼叫)

• 秘訣四:使用 Spyder的除錯器 debugger。

Spyder除錯器(Debugger)

- 圖形介面的除錯器(debugger工具列)。
 - 進入Debug mode
 - 一行一行地執行 .py 檔案(Step)
 - 進/出自訂函數(Step Into/Step Return)
 - 設立/清除中斷點

Debugger toolbar

🕸 Spyder (Python 3.6)		
<u>File Edit Search Source Run</u>	<u>Debug</u> C <u>o</u> nsoles <u>P</u> rojects <u>Tools</u> <u>V</u> i	ew <u>H</u> elp
· · · · · · · · · · · · · · · · · · ·	Debug	Ctrl+F5
Editor - C:\Users\weitingc\Drop ox\C	🕻 Step	Ctrl+F10
🗅 temp.py 🗉 test_pl_t.py 🖬	🔄 Step Into	Ctrl+F11 Cons
1#class	🚍 Step Return	Ctrl+Shift+F11
2	Continue	Ctrl+F12
2	Stop	Ctrl+Shift+F12
3import numpy a	Set/Clear breakpoint	F12
4 import matplot	Set/Edit conditional breakpoin	Shift+F12
5	Clear breakpoints in all files	
	List breakpoints	
6# read file	Debug with winndh	
7P, T, Q=np.loadt , 1 VI, Ltt, SKIPIOWS-		
8		IPython con
6# read file 7P,T,Q=np.loadt	List breakpoints	IPython con

除錯功能:設立中斷點

- 將游標移到要放置中斷點的那行程式,用滑鼠點取工具列的 set breakpoint,就會在行編號的右邊出現紅點,代表中斷點。
- 或者直接用滑鼠在行編號左邊點擊,設定中斷點
- 設定中斷點後,按F5(或Run file按鈕),執行到該處就會暫停
- 連點兩次紅點就可清除中斷點。

```
6# read file
7P,T,Q=np.loadtxt('P_T_Q1.txt
8
9# plot Q vs. P using blue di
10plt.plot(Q,P,'b--d')
11
12# set x axis, yaxis
13plt.xticks(np.linspace(0,0.0
14
15plt.xlim([0,0.02])
16plt.ylim([1000,10])
17
```

下週期中考10/26(一)

- 範圍:與python相關的上課、影片與作業內容
- 個人上機考試(3:30PM~5:20PM)建議提早到達準備
- OPEN BOOK (NOTES, INTERNET, HOMEWORK, SOLUTIONS...)
- 使用自己的筆電,請攜帶電源線、網路線
 (需要借用筆電請在<u>週六17:00前</u>先預約)
- 考試期間<u>不可與旁人討論或傳遞訊息、檔案</u> (違規者取消考試資格)
- 佔原始總成績11%

期中考準備

- 瞭解範例檔與作業程式
- 重點觀念:
 - 變數種類(實數vs整數vs字串)
 - 邏輯判斷
 - 迴圈(counter在迴圈執行過程中如何變化)
 - List
 - numpy陣列(陣列的元素、subscript的語法、迴圈 與陣列的搭配、陣列maks、陣列內建函數)
 - Numpy讀取檔案、格式化輸出到檔案
 - 畫圖 (1D plot, 2D plot)
 - Function (與主程式之間的變數傳遞)