

1) 方法描述 - 演算法原理與實作方式描述

這次的作業就是要實作出Mandelbrot Set，他的數學原理並不會太困難，最主要的公式就是 $f(x) = x^2 + c$ ，在每個iteration的過程中，都會有一些點跑出了半徑為2的圓，我們就將這些點用顏色標示出來，每個iteration都分別標示出不同的顏色。接著我們會發現iteration越多次之後，會越難看到邊際上的顏色變化，如果我們在此時放大圖片，就能再把邊際的變化看清楚。如此一直跑下去，就會形成一個無窮自動跑的影片。

實作的方式其實就是利用plot這個函式，把每輪跑出去的點用不同的顏色標記上去，要記得使用hold on，否則前面的顏色就會消失。要做到可以放大的話，情況就會相較比較複雜一些，要處理的問題會變多。像是說因為剛開始是使用(-2,2)這個區間下去取1000個點，那當放大到一定程度的時候，就會發現前面的點點都分開了，不再是連續的顏色，此時就需要再把全部的iteration以較小間距的情況下重新跑一遍。另外因為要看清楚他的邊界變化，隨著放大的過程也需要增加iteration的次數。

2) 執行方式 - 執行的函數名稱、參數設定等

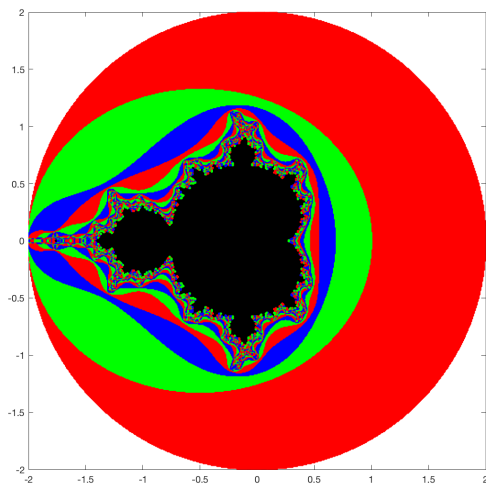
這次的作業我寫了兩個不同的版本，一個是基本的20次遞迴，一個是會放大的遞迴。

在基本的遞迴中(資料夾HW2_origin)，只使用了一個副函式Mandelbrot用來計算他遞迴的結果，Origin存這個值最剛開始是在哪個點，方便遞迴以及後面標記點時使用，另外C的話就是上一輪中留下來的點，這輪只需針對這些點做運算。整體並不會太複雜，只需要注意好inside和outside的各個點要標記清楚即可。

至於可以放大的部分(資料夾HW2)，就比較複雜一些了，目前他其實也只是個半成品，概念都有了，但實在是時間不夠把完整的影片做出來。在這部分除了main function外，還有4個副函式，分別是Check、Controlmove、Divide、Mandelbrot。check的部分是檢查一個點是否有在半徑為2的範圍內，Controlmove的部分是用來控制圖片的放大以及移動方式，Divide是為了讓圖片在放大的時候，可以重新選取1000個點而設計的，所以這裡也會使程式多了limx、limy的參數，做為確認目前要在哪些範圍中取點。最後Mandelbrot和其他參數就跟上一版的原理是一樣的。

而兩個程式在運行之後，都會自動在目錄下產生一個avi的影像檔，已確認執行結果。

3) 實驗結果 - 每一個階段的圖片、數據結果



整個的產生過程可以參考資料夾中final_100、final_500、final_1000三個影片，後面的數字代表的是每個區間中取了幾個點。而可以放大版本的結果請看資料夾中的Mandelbrot這部影片。

4) 結果討論 - 對於實驗結果的一些解釋和討論

我想先討論一下有關顏色著色這部分的問題。剛開始還很疑惑matlab不是只能一個點一個點點上去嘛，要怎麼做到像Mandelbrot這樣直接把整個區域上色呢？原來靠的就是密集的点，其實這就跟照片的原理一樣啊，用很多的畫素就能形成一個有輪廓的圖形了。所以把這個半徑為2裡面的點點都塗成黑色，對人眼來說它就是一整塊都是黑色了。如果我們將區域間的取點數改成100，就可以很明顯的發現我們可以很清楚的看到每個點點了。而這也是為什麼我們放大的過程中需要重點的原因。

另外在這次的作業中，我也解決了上次作業遇到的一個問題，執行速度的問題。在剛開始時，我遇到了一樣的問題，程式也是跑得很慢，但心想這次不能再這樣了，必須要嘗試解決他。然後看看matlab他給的warning才知道，原來是因為陣列的問題。matlab上面可以很方便的不宣告陣列大小，要使用才會去延長他。方便歸方便，但這卻會犧牲他的效能，因為他每次放大的原理是把原陣列複製到另一個陣列上，這過程的cost是很高的。所以這次修改了這個問題之後，程式的運行速度就非常非常明顯的提升了。

5) 問題討論 - 作業撰寫中遭遇的演算法問題與實作的困難

剛開始在想遞迴的運作方式時就花了一點時間，多紀錄一個origin值是嘗試了幾次之後覺得比較好的方式，原本忘記了遞迴之後就不會是原本的點了，而直接把結果值畫出來，可想而知圖案一定是錯的，但除此之外基本版就不太有什麼問題了。

比較大的困難是在寫可以放大的過程中，像是最早遇到的就是放大之後顏色會變點點的狀況。這也是為什麼會把原本寫在main中的切割獨立出來的原因。但切

割開來之後又遇到了新的問題，如果只將還在範圍中的點做新切割，那外圍的顏色就還會是一樣呈現點點狀，整張圖片就是一半連續、一半離散，會變得很奇怪。所以解決辦法應該就是每次做了新切割，就要重新做前面的遞迴，但要如何不讓這些重新遞迴的過程被看到也會是一個困難點。在下一個遇到的情況就是我要朝那裡放大下去，才會有好的效果呢？原本嘗試往正中間直接放大，但這後來發現是絕對不可行的，因為正中央的點(0,0)永遠都會在半徑2的圓中，他附近的點也是一樣，所以根本不可能被畫上新的顏色，會發現遞迴下去他永遠都是黑色的。

不過雖然知道這些問題點，但最大的問題還是在「時間」，因為這次的作業剛好在期中考前，只能利用一些零碎時間來做這個作業，所以最後也來不及解決這些問題。不過既然知道問題點了，等到期中考之後要解決應該也是很有機會的。