

數學的設計藝術-1

講者：李華倫

2021/7/12

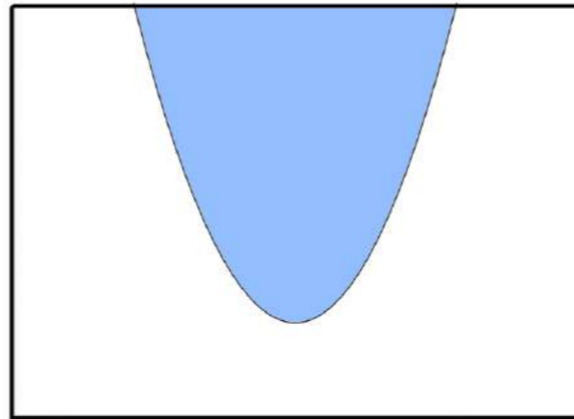
將數學想法寫成程式

不等式畫圖

想像平面上每個點皆可著色，我們用不等式來判斷哪些點該著哪個顏色。

Example:

$$y > x^2$$



程式想法：

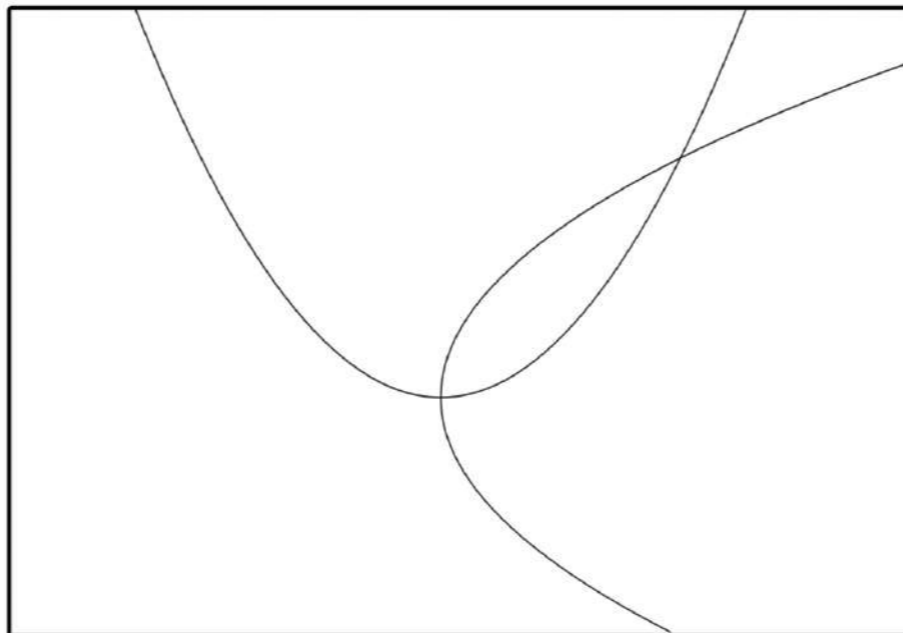
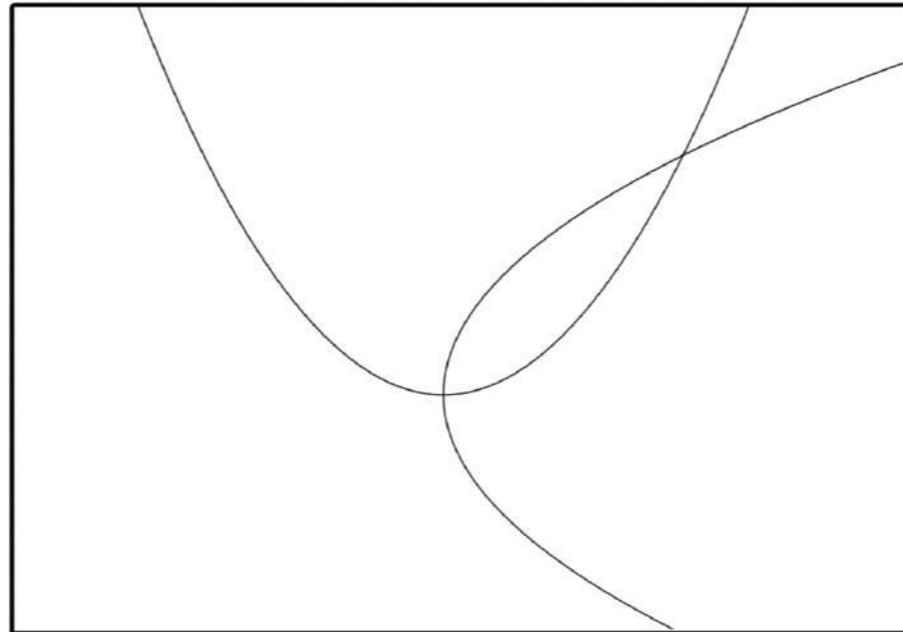
```
if y > x**2:
```

```
    在(x,y)點著上藍色
```

Example:

$$y > x^2$$

$$x > y^2$$



程式想法：

如果 $y > x^2$:

在 (x,y) 點著上藍色

如果 $x > y^2$:

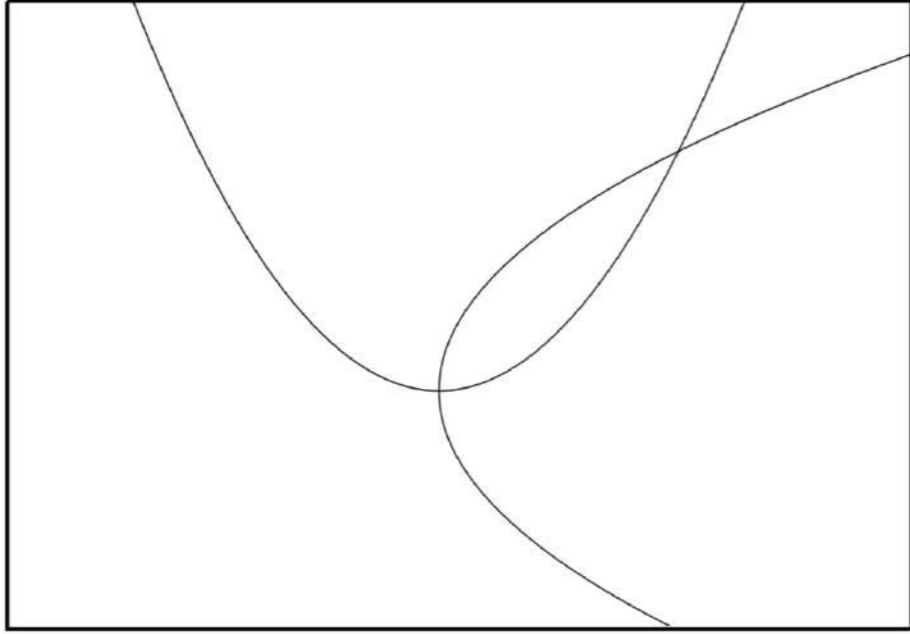
在 (x,y) 點著上黃色

如果 $y > x^2$:

在 (x,y) 點著上藍色

其他情況如果 $x > y^2$:

在 (x,y) 點著上黃色

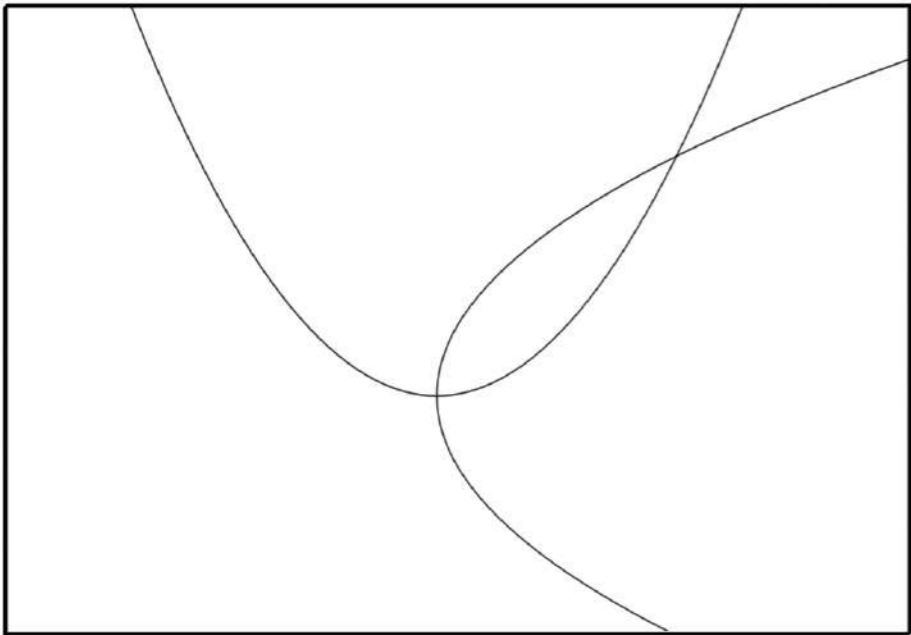


如果 $y > x^2$ 且 $x > y^2$:
在 (x,y) 點著上紅色

其他情況如果 $y > x^2$:
在 (x,y) 點著上藍色

其他情況如果 $x > y^2$:
在 (x,y) 點著上黃色

其他 :
在 (x,y) 點著上綠色



如果 $y > x^{**2}$ 且 $x > y^{**2}$:
在(x,y)點著上紅色

其他情況如果 $y > x^{**2}$:
在(x,y)點著上藍色

其他情況如果 $x > y^{**2}$:
在(x,y)點著上黃色

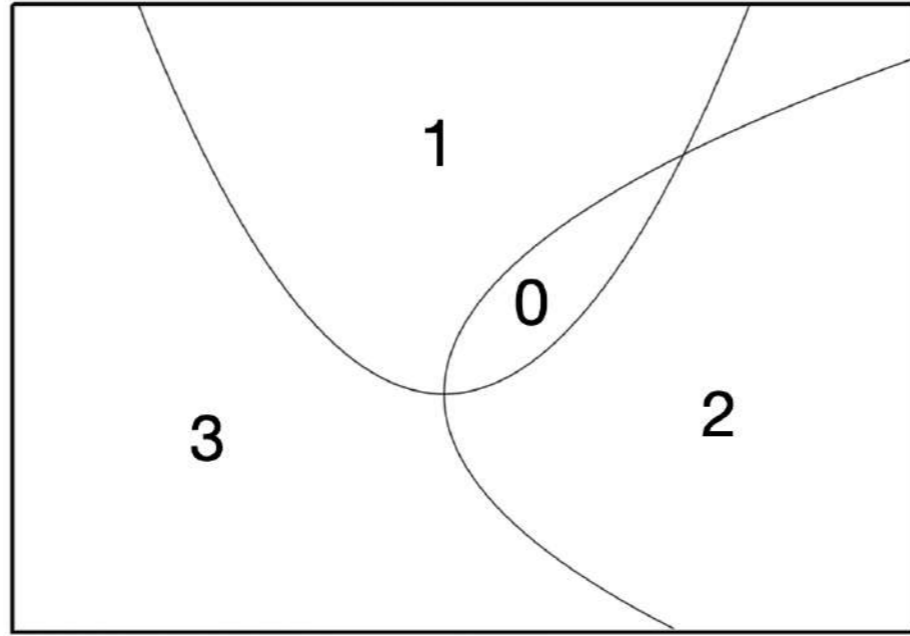
其他 :
在(x,y)點著上綠色

if $y > x^{**2}$ and $x > y^{**2}$:

elif $y > x^{**2}$:

elif $x > y^{**2}$:

else :



CC = [紅色 , 藍色 , 黃色 , 綠色]

設計函數 : $F = \mathbb{R}^2 \rightarrow \{0, 1, 2, 3\}$

須依曲域得對應值 .

數學想法：

1. 將圖像格點pixel足碼 (i, j) 轉換成 $(x, y) \in [-1, 1] \times [-1, 1]$
2. 依每個點 (x, y) 是否符合不等式來著色。
3. 函數的定義域及值域。
4. 應用模(modulo)，絕對值函數，高斯符號，max，min 及週期函數。
5. Sequence 序列(Python的List)
6. 應用線性參數做出漸層色

程式部分：條件句

1. if, elif, else

2. and , or

3. >, <, ==, >=, <=, !=

程式部分：迴圈

```
from PIL import Image
from math import *
m=1000
n=1000
new=Image.new('RGB', (m, n), (255,255,255))
for i in range(m):
    x=2*i/(m-1)-1
    for j in range(n):
        y=-(2*j/(n-1)-1)
        new.putpixel((i,j),(int(255*i/(m-1), 0, 0))
new.save('a.png')
print('OK')
```

程式部分：List

1. 將多個顏色依序寫入List。

```
cc=[ (255,0,0), (0, 0, 255), (255,255,0),(0,255,0)]
```

2. 用 `cc[i]` 來取出第 `i` 個顏色 (`i` 從 0 開始)

3. 指令 `len(cc)` 得List `cc`的元素個數，用 `cc[len(cc)-1]` 來取出最後一個顏色。

程式部分：函數

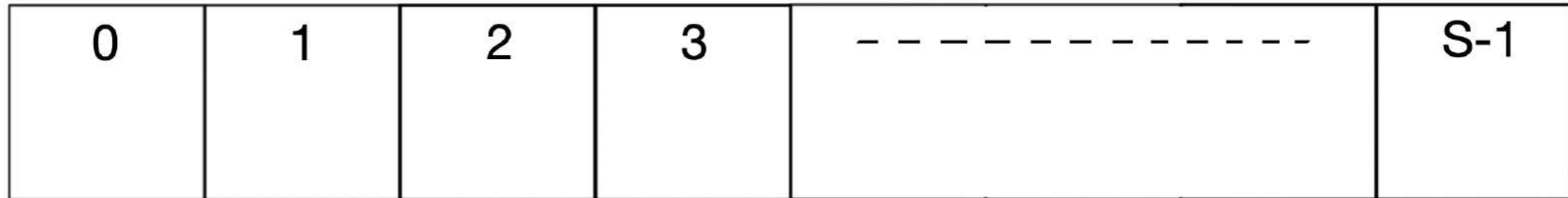
數學絕對值可寫成條件句，如下：

```
def f(x,y):  
    if x>0:  
        return x  
    else:  
        return -x
```

```
def f(x,y):  
    if x>0:  
        return 0  
    elif x>-0.5:  
        return 1  
    else:  
        return 2
```

不等式切割圖像，區塊著色

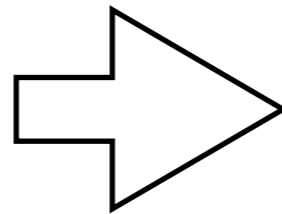
假設已分成 S 塊



已有函數 $f: \mathbb{R}^2 \rightarrow \{0, 1, 2, 3, \dots, S-1\}$

現有一新的不等式 $A(x, y) > 0$ ，它將平面分成兩部分。

```
def f(x,y):  
    if A(x,y)>0:  
        return 0  
    else:  
        return S
```



令 $F(x,y) = f(x,y) + h(x,y)$

則

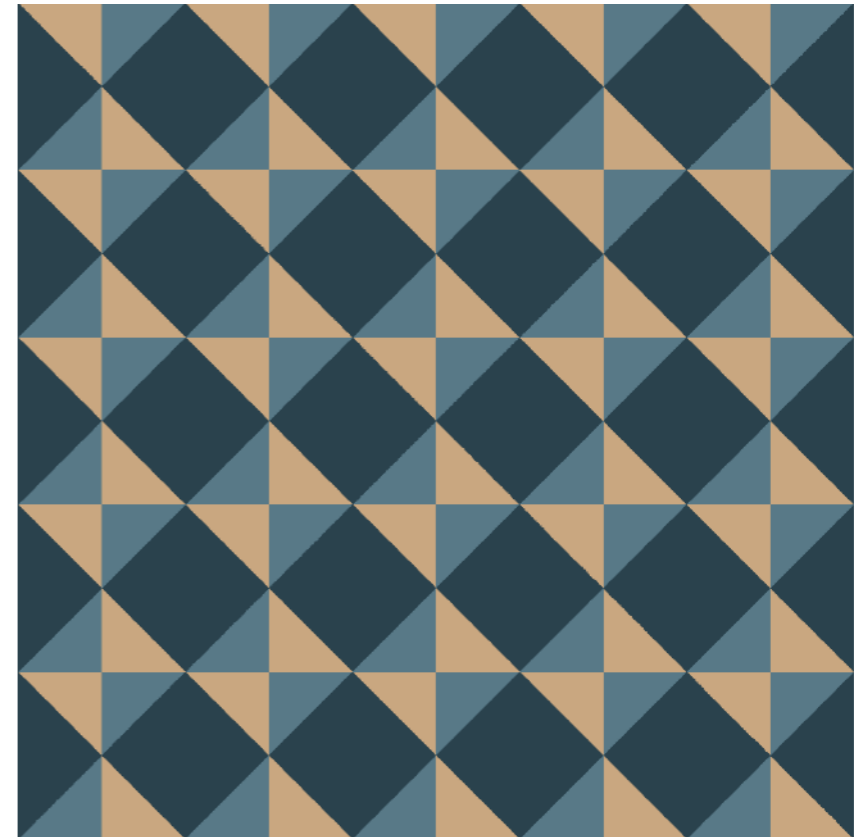
$F: \mathbb{R}^2 \rightarrow \{0, 1, 2, \dots, 2S - 1\}$

範例：

```
from PIL import Image
from math import *
m=1000
n=1000
cc=[(192, 237, 255),(42, 66, 77),(88, 121, 135),
    (125, 88, 45),(201, 167, 128),(255, 245, 135)]
new=Image.new('RGB',(m,n),cc[1])

for i in range(m):
    x=2*i/(m-1)-1
    for j in range(n):
        y=-(2*j/(n-1)-1)
        if sin(5*pi*x-pi/2)<sin(5*pi*y-pi/2):
            new.putpixel((i,j),cc[2])
            if sin(5*pi*(x+y)-pi/2)<sin(5*pi*(x-y)-pi/2):
                new.putpixel((i,j),cc[4])

new.save('A_2.png')
print('OK')
```



範例：

```
from PIL import Image
from math import *
m=1000
n=1000
new=Image.new('RGB',(m,n),(255,255,255))
cc=[(208, 63, 71),(248, 118, 156),
     (57, 131, 181),(240, 180, 67),(245, 212, 197)]
def f(x,y):
    return y-0.125*sin(4*pi*x)
for i in range(m):
    x=2*i/(m-1)-1
    for j in range(n):
        y=-(2*j/(n-1)-1)
        k=floor(f(x,y)*5)
        t=f(x,y)*5-k
        k0=k%len(cc);k1=(k+1)%len(cc)
        R=t*cc[k0][0]+(1-t)*cc[k1][0]
        G=t*cc[k0][1]+(1-t)*cc[k1][1]
        B=t*cc[k0][2]+(1-t)*cc[k1][2]
        new.putpixel((i,j),(int(R),int(G),int(B)))
new.save('A_4.png')
print('OK')
```

