中華民國第四十八屆中小學科學展覽會
作品說明書

高職組　電子、電機及資訊科

最佳團隊合作獎

091004

當黎曼愛上了不等式(任意封閉曲線求面積法)

學校名稱：臺中市私立宜寧高級中學

作者：	指導老師：
職三 黃偉國	程瑋翔
職三 林洸曄	吳嘉鴻
職三 林汶亮	
職三 劉忠易	

關鍵詞：　任一邊形面積、黎曼和、不等式

作品名稱：當黎曼愛上了不等式 (**任意封閉曲線求面積法**)

摘要：

作品定位：方格紙法求任意封閉曲線所圍的面積。

理論依據：線性規劃與黎曼積分及無窮級數之夾擠性質。

首要目標：完成單位圓之方格面積法計算，並由誤差章節可以證明出，透過方格面積窮極細化計算，無論多小精度的誤差皆可達到，讓研究方法得以驗證。

第二目標：完成任意三邊形方格法面積，透過三點面積法比對誤差率。

第三目標：完成任意 N 邊形面積，此區域代表 N 條不等式所圍之區域面積。

第四目標：完成任意封閉曲線面積，方法沿著封閉曲線取 N 點，任意封閉曲線區域便可以 N 條不等式所圍之區域面積來詮釋。

最後目標：完成方格紙面積法求任意封閉曲線面積之誤差分析 (透過 N 多邊形模擬圓面積之誤差控制，確保任意封閉曲線可由 N 多邊形估計面積)。

壹、 研究動機：

有一天在上數學課的時候，我們這節課是上黎曼和的課程，老師在課堂上問我們說：「各位同學，問你們一個小小的問題，請問你們，假如我們要求圓的面積，但是我們不用“πR^2”這個方法，那我們該怎麼求圓的面積呢？」

突然有一位學生舉手說：「老師，我們可以用正方形的格子來算圓的面積，將圓細分成很多正方形，然後再算出正方形的總和面積，約能算出圓的大約面積。」

老師說：「OK！ 這個方法不錯，但是我們要如何控制誤差？ 」

另一位學生就說：「我們可以把格子切割的更小一點點，透過將圓細分成更細的正方形，這樣一直細分，誤差也會離真正面積越來越少，細分到最後就能把圓真正的面積求出來。」

老師說：「那你們學過 VB，該如何由電腦幫您計算出呢？」

貳、 研究目的：

一、基於好奇心與好勝心解答yahoo知識問答，尋找生活問題的電腦解決方法。

二、找出簡單的估算不規則的圖形面積的方法。

三、數學多加一些有效電腦估算方法，問題解決不一定要使用複雜的公式。

四、透過程式的寫作，增加學習數學的有趣度和實用度。

五、小學看過老師用方格紙來計算圖形面積，現在我們期許用電腦程式來完成。

六、提供實用軟體讓小學生們更容易瞭解方格求面積方法，更願意公開程式碼供教學研究。

參、研究設備及器材：

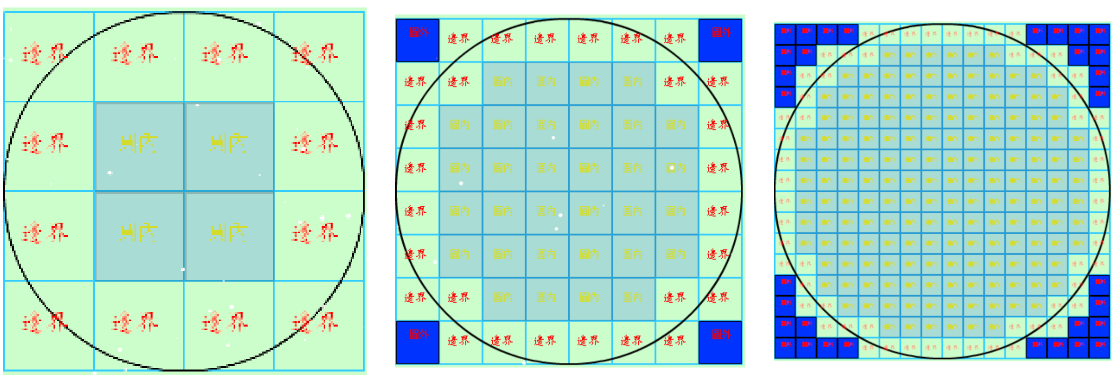
我們使用的平台 XP SP2 以及 P4 雙核心電腦，1GB RAM；使用軟體有 2 個，一個是” Microsoft Visual Basic 6.0”另一個是” Macromedia Flash 8”這兩套程式語言。

肆、研究過程或方法：

用正方形小格子求圓的面積，我們用方格數來估算出圓曲線內部區塊的大約面積，同時也在尋找有效且簡易的適當的方式來模擬任意不規則圖形，然而誤差要如何控制？比如說把格子變得更小，變成跟奈米一樣小，這樣算圓的面積一定會更精確，這個方法也可以利用在不規則形狀上面，利用格子的細化去算不規則形狀面積，努力研究去，GO~！

以圓面積爲例：由區塊的產生透過已知圓方程式 $f(x,y)=x^2+y^2-R^2$ 可以很容易判定出區塊之四頂點落於曲線的內部與外部或邊界。真實的圓面積爲 πR^2，透過方格紙的計算，圓面積介於[*註] 內部區塊和 (下和) 與內部區塊與邊界區塊和 (上和) 之間，當方格紙取得越細，則越接近真實面積。透過電腦窮極運算，便可知得到

$$\sum_{\Delta x\to 0}(\Delta x)^2 number \quad (\text{內部區塊}) \ \le \pi R^2 \le \sum_{\Delta x\to 0}(\Delta x)^2 number \ (\text{內部區塊+邊界區塊})$$



[*註] 註解：本文自行定義物件：點與區塊的形式如下

1、內部點　2、邊界點　3、外部點 4、內部區塊 5、邊界區塊　6、外部區塊 7、$\Delta x=$ 方格子邊長

製作流程

對於任一封閉曲線並不存在既成方程式，以傳統黎曼和求面積法，透過切割長方形，得由已知函數方能計算出封閉曲線面積，我們是試著在圓邊界上切割 10 等份，取出 10 個點，用正 10 邊形來模擬圓面積，欲得到更精確面積可在細切割，以 100 邊形或更多邊形來模擬計算圓面積。然由正多形模擬圓計算曲線所含方格面積尚有許多問題待解決，圓是有既成方程式 $f(x,y)=x^2+y^2-R^2$,可以容易判區塊位於圓的內部、邊界或外部，在此正 10 邊形所在區域，電腦如何計算出封閉曲線內方格的數目呢？

諸多新問題產生，原本試著將正 10 邊形，切成 10 塊三角形計算面積和，然此舉和原本規劃方格面積法相違背。而且對於凸多邊形多易於解決面積計算，但對於凹多邊形以及任意形狀的切割與計算便相形困難。切割成三角形，由面積公式求之對於最後章節的誤差分析給了相當大的幫助。

製作過程指導老師更提出了，如下情況電腦如何著手計算？

1、當於小畫板隨意畫一個封閉曲線，面積如何求出？
2、透過方格面積法、如何計算出已知方程式圓錐曲線的面積？
3、給定三個點所圍成的三角形封閉區間如何用方格計算面積？
4、三角形區域如何用課堂所學的的知識求出其方程式？

解決方法：

1 、三角形區域代表三不等式所圍之區域面積。
2 、透過直線兩點式便可求出直線方程式。
3 、透過內部區域的選擇，便可求出聯立不等式。
4 、任意一個封閉曲線，可以透過取得更多點達到多點聯立不等式。
5 、透過 Flash 導引線的寫作，動點可以沿著導引線繞一圈。
6 、如同圓的切割方法將導引線切等份取出多點。
7 、封閉曲線透過多點模擬，產生多直線之聯立不等式模擬封閉區間面積。

問題產生：

1、一直線將平面切成兩等份；兩直線切成四等份；三直線與多直線更甚。多直線產生平面，面積方格所在區塊的判斷隨多聯立方程式而日形困難。
2、Flash 程式語言使用 Action Script 語法。撇開 VB 重新學習 Action Script 產生無限的壓力與困難。
3、指導教師要求全程自行完成，但 Flash Action Script 的學習真的相當困難。
4、原本以為簡單的紙筆數學。聯立不等式化成程式表示卻錯誤連連！甚至不知為何解決？

一、已知函數格子面積求法(圓面積為例描述黎曼和)

程式的寫作與發展得定義如下物件：

1、**內部點**：落在封閉區間邊界內之點

2、**邊界點**：落在封閉區間邊界上之點

3、**外部點**：落在封閉區間邊界外之點

4、**內部區塊**：當四邊界點皆爲內部點時稱內部區塊

5、**邊界區塊**：非全爲內部點(亦非爲全外部點)或其中一個爲邊界點

6、**外部區塊**：區塊四頂點皆外部點稱外部區塊

7、**所求面積**：其中 $\Delta x=$ 方格子邊長

$$\sum_{\Delta x\to 0}(\Delta x)^2\,number\ \text{(內部區塊)} \le \pi R^2 \le \sum_{\Delta x\to 0}(\Delta x)^2\,number\ \text{(內部區塊+邊界區塊)}$$

※以圓爲例，判別函數爲 $f(x,y)=x^2+y^2-R^2$；平面上任意點 $P(x_0,y_0)$

1、內部點： $f(x_0,y_0)<0$

2、邊界點： $f(x_0,y_0)=0$

3、外部點： $f(x_0,y_0)>0$

4、內部區塊：例如 A

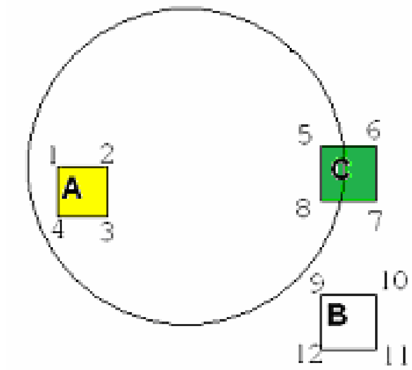
條件:四邊界點皆爲內部點。

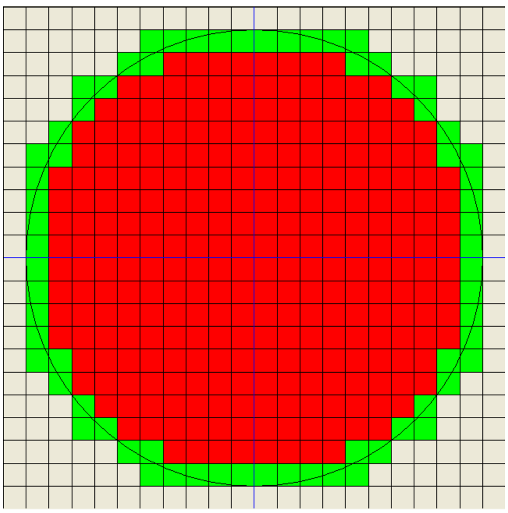
5、邊界區塊：例如 C

條件:(非全爲內部點亦非爲全外部點)或其中一個爲邊界點。

6、外部區塊：例如 B

條件: 全部爲外部點。





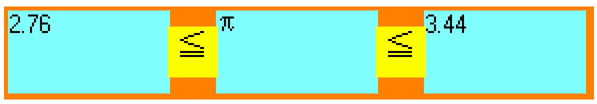
半徑爲 1 的圓爲例

我們利用 0.1×0.1 方格求出來的面積是

(下和)$2.76\le\pi\le 3.44$(上和)

誤差率

12.1%(下和)~9.554%(上和)。



二、三角形格子面積求法(線性規劃法求封閉曲線範圍)

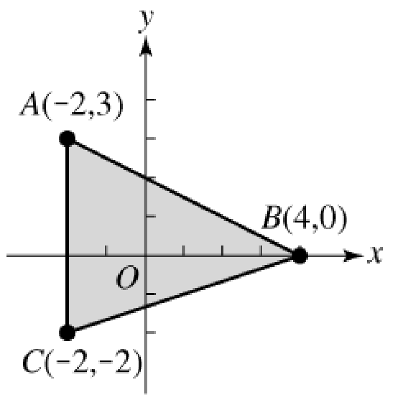
(一)給定三點產生聯立不等式

上一章中求圓格子面積,透過點代入 $f(x,y)=x^2+y^2-R^2$ 判定內部點或邊界點進而決定正方格區塊是否爲內部區塊,然三角形(多邊形)例子,電腦如何透過分析判別出位在封閉區間的相對位置?於是我們引入了高職數學所學的線性規劃以不等式來描訴封閉三角形,透過聯立不等式可以判定方格位於封閉曲線的相對位置,進而可以準確求出三角形之面積。

誤差分析則透過標準三角形三點面積公式,亦由誤差分析中,可驗證出任意誤差精度要求,只要藉由細化方格即可達到。以下是數學產生三角形區域的限制函數範例。

[範例] $A(-2,3),B(4,0),C(-2,-2)$ 試以 x,y 的不等式組表△ABC 所圍成的區域。

解析:



給定平面上兩點 $P(x_1,y_1),Q(x_2,y_2)$

直線 PQ 的兩點式: $\dfrac{y-y_1}{x-x_1}=\dfrac{y_2-y_1}{x_2-x_1}$

A(-2,3)B(4,0)帶入兩點式: $\dfrac{y-(3)}{x-(-2)}=\dfrac{(0)-(3)}{(4)-(-2)}$

可得 $\overleftrightarrow{AB}:x+2y-4=0$

B(4,0)C(-2,-2)帶入兩點式: $\dfrac{y-(0)}{x-(4)}=\dfrac{(-2)-(0)}{(-2)-(4)}$

可得 $\overleftrightarrow{BC}:x-3y-4=0$

因爲 $(0,0)$ 在△ABC 的內部

故包含 $O(0,0)$ 的區域已連立不等式表示爲 $\begin{cases} x+2y\le 4 \\ x+2\ge 0 \\ x-3y\le 4 \end{cases}$

(二)程式語言如何產生聯立不等式?

步驟 1：輸入三點座標並連接三邊。

步驟 2：給定之三點座標透過兩點式產生三條直線方程式：

$$\begin{cases} f_1: x + 2y - 4 = 0 \\ f_2: x + 2 = 0 \\ f_3: x - 3y = 0 \end{cases}$$

步驟 3：用滑鼠點選欲求之區域內部，取出內部點之參考點；比方指到(0,0)

步驟 4：將參考點(0,0)帶入聯立方程式產生聯立不等式

$(0)+2(0)-4<0$

因為 x 係數為正故在圖形的左方

y 係數為正故在圖形的下方

故 $x+2y\le 4$

$(0)+2>0$

因為 x 係數為零故無法判斷方向

y 係數為正故在圖形的上方

故 $x+2\ge 0$

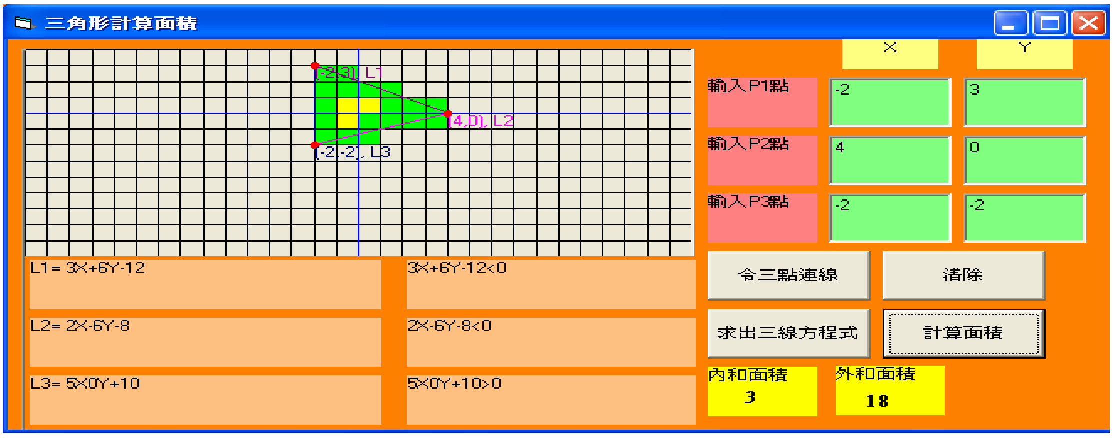
$(0)-3(0)-4<0$

因為 x 係數為正故在圖形的左方

y 係數為負故在圖形的上方

故 $x-3y\le 4$

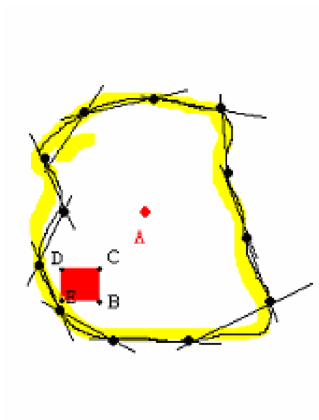
故聯立不等式 $\begin{cases} x+2y\le 4 \\ x+2\ge 0 \\ x-3y\le 4 \end{cases}$



(三)程式語言如何判斷凸多邊形內部點與內部區塊？

1.由參考點 A ,P 點之回傳值判斷 P 所在區域

$$\underset{k=1\ to\ n}{\text{回傳值}(\mathrm{P})}=\begin{cases}0(\text{邊界點})\ \mathit{if}\ \exists \mathrm{i}\ \ f_{i}(\mathrm{P})=0, \\ \qquad\qquad f_{k}(A)f_{k}(\mathrm{P})>0,\mathrm{k}\neq \mathrm{i} \\ 1(\text{內部點})\ \mathit{if}\ \forall \mathrm{k}\ \ f_{k}(A)f_{k}(\mathrm{P})>0\ ; \\ 2(\text{外部點})\ \mathit{if}\ \exists \mathrm{i}\ \ f_{k}(A)f_{k}(\mathrm{P})<0\end{cases}$$



2.判斷區塊 BCDE 之回傳值

(1)內部區塊：$\mathit{if}\quad B\cdot C\cdot D\cdot E=1$

(2)邊界區塊： $\mathit{if}\quad B\cdot C\cdot D\cdot E=0\ \ \mathit{or}\ \ \mathit{if}\ 1<B\cdot C\cdot D\cdot E<16$

(3)外部區塊：$\mathit{if}\quad B\cdot C\cdot D\cdot E=16$

1、**內部點**：點落在封閉區間邊界內

例如：邊界點 5, 6, 7, 8, 9,10

2、**邊界點**：邊界點落在封閉區間邊界上

例如：點 15

3、**外部點**：邊界點落在封閉區間邊界外

例如：點 1, 2, 3, 4,11,12,13,14,16

4、**內部區塊**：四邊界點皆爲內部點

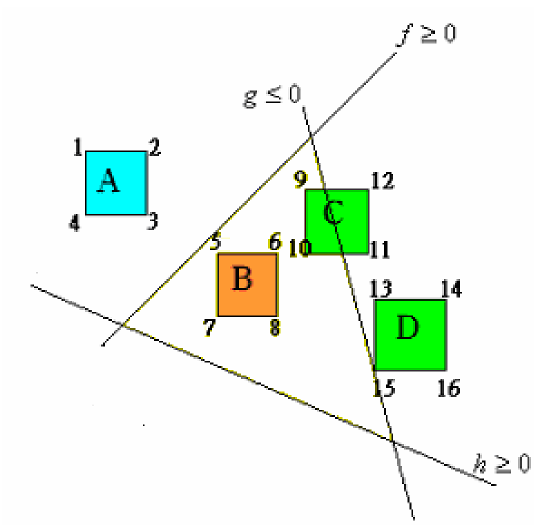
例如：內部區塊 B

5、**邊界區塊**：非全爲內部點亦非爲全外部點)或其中一個爲邊界點

例如：**邊界區塊** C,D

6、**外部區塊**：區塊四頂點皆外部點

例如：**外部區塊** A



7、**所求面積**：其中 $\Delta x=$ 方格子邊長

$$\sum_{\Delta x\to 0}(\Delta x)^2\,number\ \ (\text{內部區塊})\ \leq \pi R^2\leq \sum_{\Delta x\to 0}(\Delta x)^2\,number\,(\text{內部區塊+邊界區塊})$$

（四）線性凸多邊形面積

已知凸多邊形的頂點坐標依序為 (x_1,y_1) , (x_2,y_2) , …… , (x_n,y_n)，其面積為

$\overline{P_1P_2},\overline{P_2P_3},\overline{P_2P_3},\cdots,\overline{P_{n-1}P_n},\overline{P_nP_1}$ 所圍區域面積 。

步驟 1：輸入 N 點座標並連接 N 邊。
步驟 2：給定之 N 點座標透過兩點式產生 N 條直線方程式：

$$\begin{cases} f_1:\ a_1x+b_1y+c_1=0 \\ f_2:\ a_2x+b_2y+c_2=0 \\ f_3:\ a_3x+b_3y+c_3=0 \\ \quad\cdots \\ f_N:\ a_Nx+b_Ny+c_N=0 \end{cases}$$

步驟 3：用滑鼠點選欲求之區域內部，取出內部點之參考點 A；比方指到(x_A,y_A)
步驟 4：將參考點(x_A,y_A)帶入聯立方程式產生聯立不等式

$$f(x_A,y_A)=a_i(x_A)+b_i(y_A)+c_i\ ;\ i=1\ to\ N$$

$$f(x_A,y_A)=a_i(x_A)+b_i(y_A)+c_i>0$$

CASE1: $a_i>0$ 則 $a_i+b_i+c_i>0$

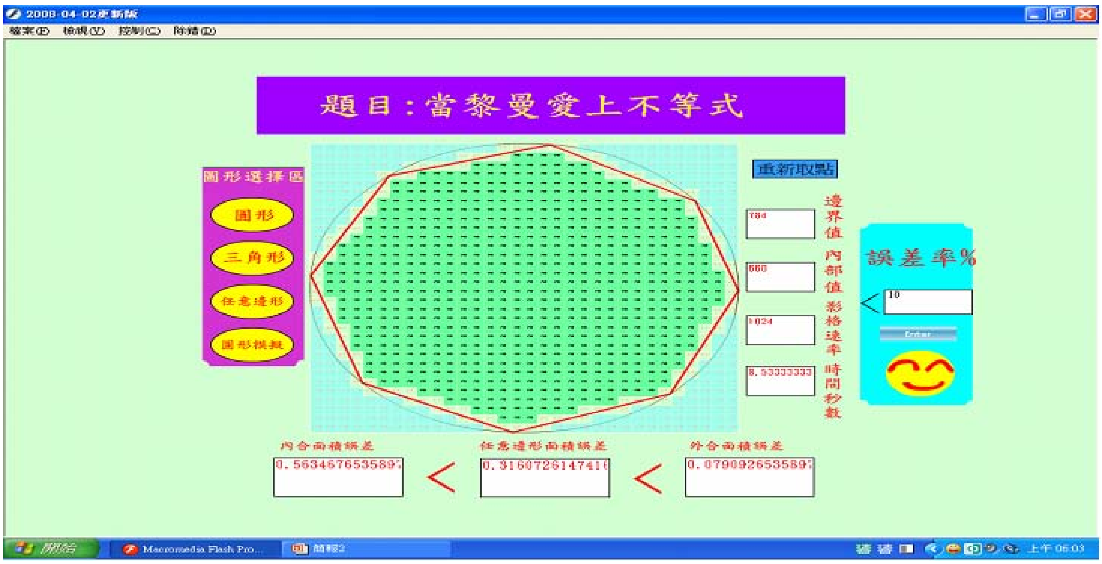
CASE2: $a_i<0$ 則 $a_ix+b_iy+c_i<0$

CASE3： $a_i=0$且$b_i>0$ 則 $b_iy+c_i>0$

CASE4： $a_i=0$且$b_i<0$ 則 $b_iy+c_i<0$

故聯立不等式可由此產生

$$\sum_{\Delta x\to 0}(\Delta x)^2 number(\text{內部區塊})\le Area\le\sum_{\Delta x\to 0}(\Delta x)^2 number(\text{內部區塊}+\text{邊界區塊})$$



(五)不規則圖形之面積求法：

有邊界之圓滑封閉曲線圖形,邊界 $x \in [a,b]$，$y \in [c,d]$，則**封閉曲線面積(Area)**

$$\sum_{\Delta x \to 0} (\Delta x)^2 number(內部區塊) \le Area \le \sum_{\Delta x \to 0} (\Delta x)^2 number(內部區塊+邊界區塊)$$

然如何判定方格與封閉區線之相對位置？方法透過圓滑封閉曲線邊界取點產生 N 多邊形；如同多邊形模擬圓封閉曲線一般，計算出多邊形面積便能估算任意不規則的封閉曲線面積。

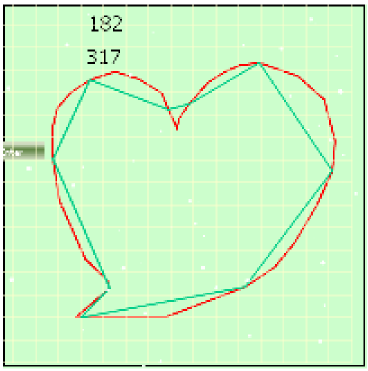
步驟一：**產生一電子畫板**，於板上畫出一封閉曲線，由使用者要求誤差範圍條件下，電腦判定出需要之 N 點座標，再沿著封閉曲線繞一圈，固定的時間間格產生座標點，以 N 條之線段來呈現此任意形狀之封閉曲線區域。

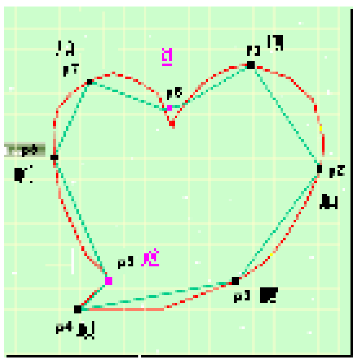
步驟二：**N 條之線段所詮釋之 N 多邊形**可能為凸多邊形或凹多邊形。然線性規劃格子點求面積法僅適用於前者，若遇到凹多邊形必需先將凹多邊形化成多個凸多邊形的面積和。

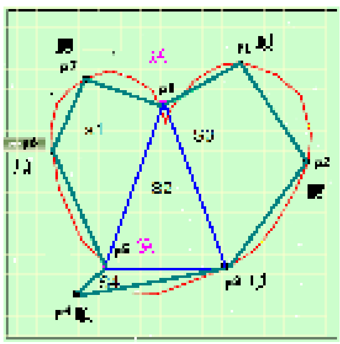
步驟三：**產生凹點與凸點**。假設 N 多邊形之頂點依 $P_1, P_2, P_3, \cdots, P_n$ 順時針排序，如下圖之範例若依序繞一圈時，由 $P_1 \to P_2 \to P_3$ 過程將 $P_1P_2P_3$ 視作獨立三角形，過 P_2 時為順時針旋轉，故 P_2 為凸點；同理 P_3、P_4 為凸點。當 $P_4 \to P_5 \to P_6$ 過程將 $P_1P_2P_3$ 視作獨立三角形因此過 P_5 時為逆時針旋轉，故 P_5 為凹點；以此類推依序可得 P_6、P_7 為凸點，而 P_8 為凹點。

步驟四；**透過面積分割，消滅凹點**，使得各區塊之多邊形皆為凸多邊形，使用者要求誤差自動決定計算方格子大小或直接指定方格大小開始計算，回傳 $S_1 \sim S_k$ 各區塊凸多邊形中內部區塊與邊界區塊個別數目來估計面積總合。

$$\sum_{s_1}^{s_k}\left(\sum_{\Delta x \to 0} (\Delta x)^2 number(內部區塊)\right) \le Area \le \sum_{s_1}^{s_k}\sum_{\Delta x \to 0} (\Delta x)^2 number(內部區塊+邊界區塊)$$







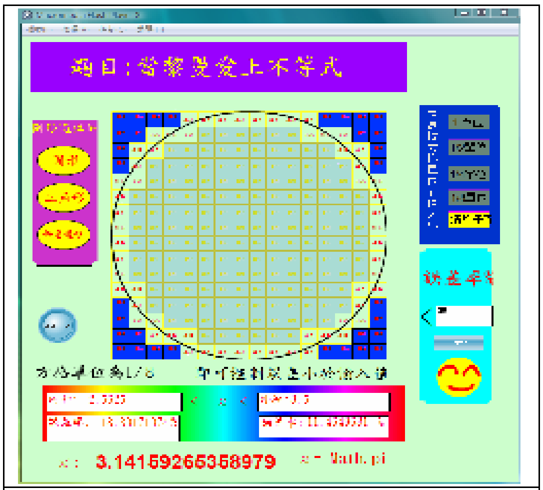
(六)誤差分析

$$\text{誤差率}=\max\left\{\left|\frac{\text{下和}-\text{真實面積}}{\text{真實面積}}\right|\times 100\%,\left|\frac{\text{上和}-\text{真實面積}}{\text{真實面積}}\right|\times 100\%\right\}$$

1. 圓誤差分析(以半徑為 1 的圓面積為例)

控制誤差率<50%時,

方格得達 1/4	時間花費 00:00:7



控制誤差率<30%時,

方格得達 1/8	時間花費 00:00:21

控制誤差率<10%時,

方格得達 1/16	時間花費 00:01:21

控制誤差率<5%時,

方格得達 1/32	時間花費 00:20:28

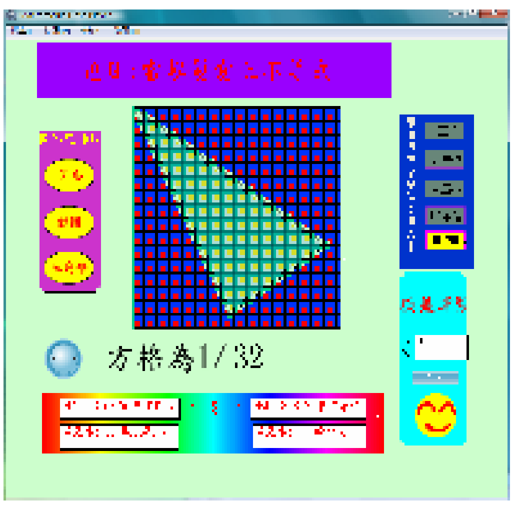
2. N 多邊形誤差分析（以三角形為例）

$$\text{N 多邊形所圍的區域面積}=\frac{1}{2}\left|\begin{array}{l} x_1y_2+x_2y_3+x_3y_4+\cdots+x_{n-1}y_n+x_ny_n \\ -x_2y_1-x_3y_2-x_4y_3-\cdots-x_ny_{n-1}-x_1y_n \end{array}\right|$$

$$\text{誤差率 }=\text{Max}\left\{\frac{\text{下和}-\text{N多邊形面積}}{\text{N多邊形面積}}\times 100\%\,,\frac{\text{上和}-\text{N多邊形面積}}{\text{N多邊形面積}}\times 100\%\right\}$$

控制誤差率<50%時,	
方格得達 1/8	時間花費 00:00:29

控制誤差率<30%時,	
方格得達 1/16	時間花費 00:01:09



控制誤差率<20%時,	
方格得達 1/32	時間花費 00:07:21

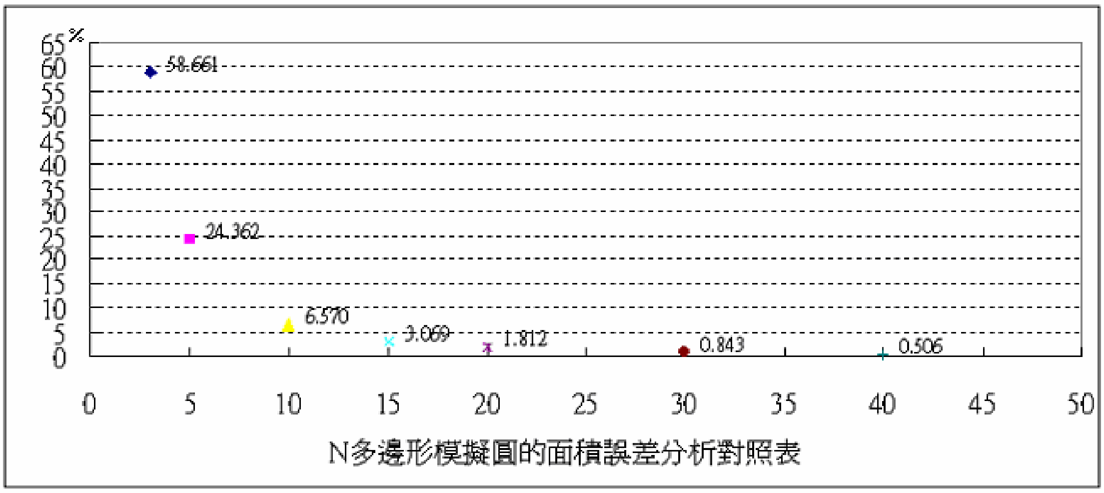
控制誤差率<10%時,	
方格得達 1/64	時間花費 00:57:27

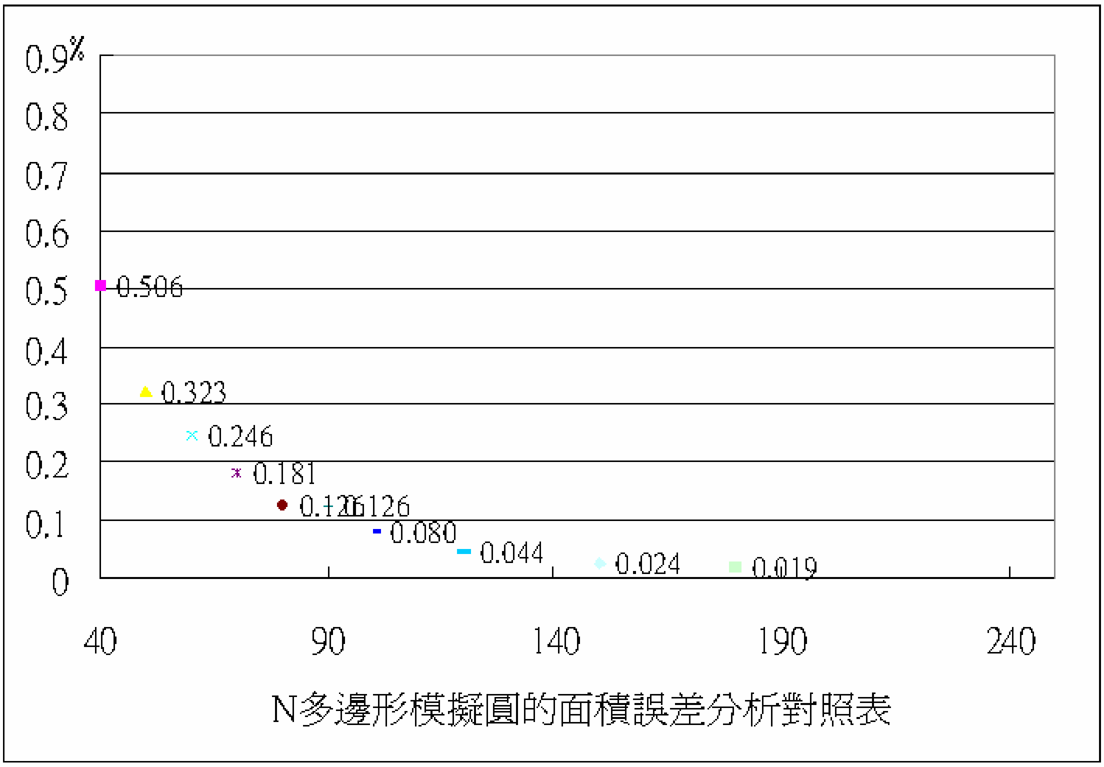
3. 多邊形模擬圓面積之誤差分析

方法將圓沿邊界作切割，每隔固定間隔取點， 輸入需要之 N，因此**等距**產生 N 個座標並將其連接成 N 邊形。以 N 多邊形模擬單位圓的面積,若取得之點坐標依序為 (x_1,y_1) , (x_2,y_2) , $\cdots$, (x_n,y_n)，則其

N 多邊形所圍的區域面積= $\dfrac{1}{2}\left|\begin{array}{l} x_1y_2+x_2y_3+x_3y_4+\cdots+x_{n-1}y_n+x_ny_n \\ -x_2y_1-x_3y_2-x_4y_3-\cdots-x_ny_{n-1}-x_1y_n \end{array}\right|$

N 多邊形相對於圓的面積誤差率 $= \left|\dfrac{\text{N邊形面積}-\pi}{\pi}\right| \times 100\,\%$





4. N 多邊形模擬圓與任意封閉圖形的面積流程與思考

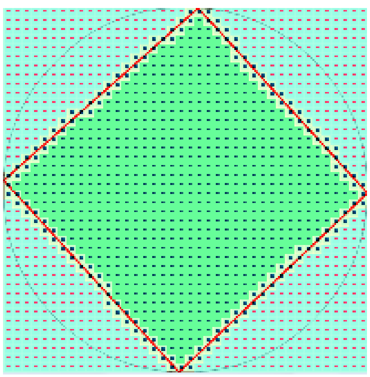
假想某徒弟當下沒有圓公式也不知 N 多邊形的公式與只有畫圓上正 **N 多邊形工具**及和**無限可能細度的方格紙**，希望徒弟藉由 1 公尺單位圓上的正多邊形與方格紙估算出面積約 3 平方公尺；但當下師傅卻擁有「圓的公式」、「正 N 多邊形公式」、「給定正多邊形圓誤差 N 邊形產生器」以及「N 多邊形的方格面積程式」，**請問該徒弟至少該用幾邊形模擬此圓，算出接近師傅心中的 3.~~1415926~~** 呢？

階段 1. 取正三角形可能算出 3 嗎?

解答： 不可能! 因爲由師傅經過思考 3 和圓真實面積 π 誤差真實面積誤差約 4.5%，由誤差對照表便知大約要 15 邊形。

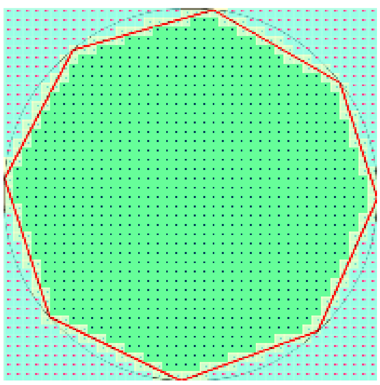
階段 2.\如此徒弟真的取 15 邊形了，開始用方格紙量面積了，師父規定每一塊小方格不能超過多邊形的邊界，請問這徒弟藉由不停的換更細的方格紙由 $1\times1 \to \frac{1}{2}\times\frac{1}{2} \to \frac{1}{4}\times\frac{1}{4}$ $\cdots$您知道大概**要取到多小的方格紙**就可以算出面積超過 3 了呢?

解答： 聰明的師傅心想 15 邊形接近了，因爲正 15 邊形和圓的誤差約 3%，若取方格子來計算面積和圓誤差小於 5%在 15 邊形中就用掉 3%，故師傅便拿起格子點計算多邊形工具。輸入正 15 邊形，誤差<1.5%，經過數分鐘的計算，便查出徒弟算出面積達 3 時，大概使用的方格子大小了。



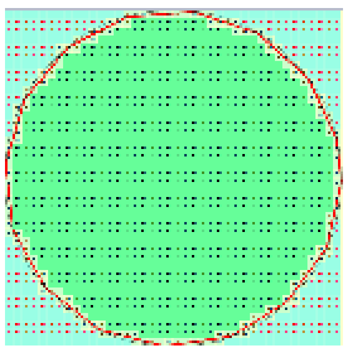
N=4 時面積最大可達 2 平方公尺

真實面積誤差達 63.66%



N=8 時面積最大可達 2. 8284 平方公尺

和真實面積誤差達 10.06 %



N=15 面積最大可達 3.04

誤差<1.5%所需要的 $\Delta x = \frac{1}{?}$

5. 任意封閉曲線之誤差分析與控制

試想當徒弟沒有任意曲線面積公式也不知 N 多邊形的公式與只有能產生 **N 多邊形工具**及和**無限可能細度的方格紙**，希望藉由多邊形與方格紙估算出與真實封閉曲線面積誤差在 5%以下；當師傅傳承了徒弟「圓的公式」、「給定正多邊形圓誤差 N 邊形產生器」以及「N 多邊形的方格面積程式」，**請問該徒弟該如何決定將曲線切成幾份產生 N 多邊形足以模擬此曲線面積，並控制誤差在 5%以下呢？**

Step1：**欲求真實面積誤差< Ei (比方 5%)條件下之面積計算**，徒弟傳承師傅之技能，首先想完成原本就會的**產生 N 多邊形**。藉由取動點沿邊界導引一周，加入時間因子等時間距取點，徒弟心想 N 邊形的產生速度快，便以 $Ei\times(10^{-2})\,\%$**百倍精密度誤差**控制呼叫 N 產生程式，透過多邊形模擬圓之程式以誤差 $Ei\times(10^{-2})\,\%$條件產生 N 值；**因而得以控制和真實面積誤差 0.005%條件下，以 N 多邊形取代原始封閉曲線**。

Step2：**封閉曲線計算之誤差控制**：由輸入之 Ei 值（5%為例），取 N 邊形過程，誤差率控制達到萬分之五決定出之 N 多邊形，足以取代原任意封閉曲線，故徒弟啓動師傅給的「N 多邊形的方格面積程式」，格子計算面積由 $1\times1\to\frac{1}{2}\times\frac{1}{2}\to\frac{1}{4}\times\frac{1}{4}\cdots$，不用再靠人力一塊塊的計算出 N 邊形面積和師傅所給真實面積的誤差趨近於 0.005%遠小於 5%。

N 多邊形模擬任意封閉曲線的面積誤差率分析

$$E_N=\left|\frac{\text{N邊形面積}-\text{真實面積Area}}{\text{真實面積Area}}\right|\times100\%\cong\left|\frac{\text{N邊形面積}-\text{單位圓面積}}{\text{單位圓面積}}\right|\times100\%<Ei$$

$$L_{real-N}=\sum_{\Delta x\to0}(\Delta x)^2\,number(\text{任意曲線下之N多邊形內部區塊})$$

$$L_{real}=\sum_{\Delta x\to0}(\Delta x)^2\,number(\text{任意曲線下內部區塊})$$

$$U_{real-N}\ \sum_{\Delta x\to0}(\Delta x)^2\,number(\text{任意曲線下之N多邊形內部區塊}+\text{邊界區塊})$$

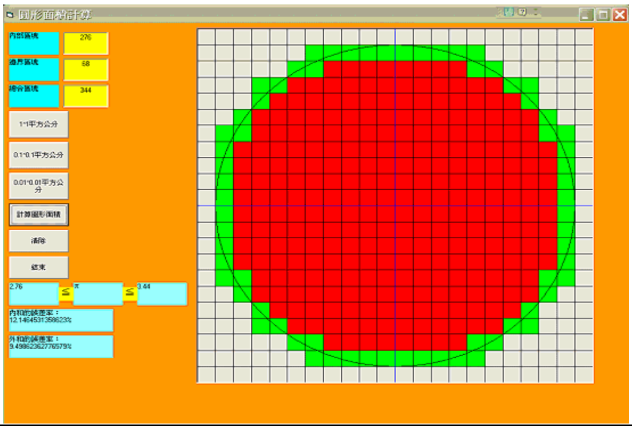
$$U_{real-N}\ \sum_{\Delta x\to0}(\Delta x)^2\,number(\text{任意曲線下內部區塊}+\text{邊界區塊})$$

當 $\Delta x\to0$，$E_N=E_i\times(10^{-2})$，$\max\left|\frac{L_{real-N}-L_{real}}{L_{real}}\right|\to E_N$,$\max\left|\frac{U_{real-N}-U_{real}}{U_{real}}\right|\to E_N$

則 $\left|\frac{Area-Area\_N}{Area}\right|\to E_N<<E_i$

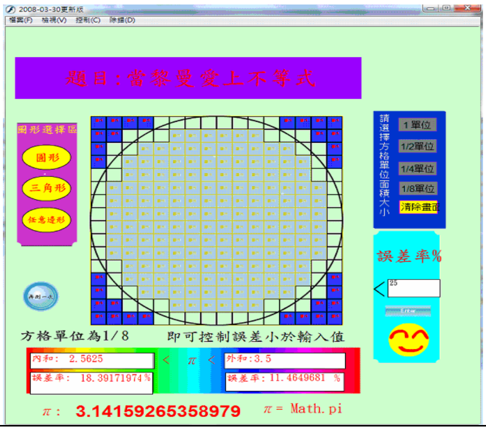
伍、研究結果

(範例一)圓形(V.B.)↓



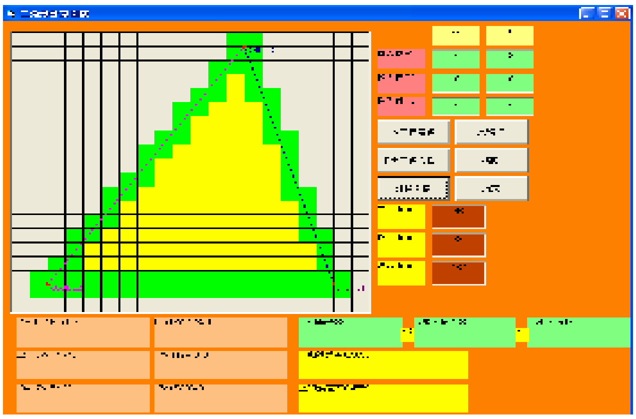
首先點選方格子的大小，有 1*1 平方公分、0.1*0.1 平方公分和 0.01*0.01 平方公分的格子，選完之後她便會幫你畫格子在圓上面，之後請點選"計算圓形面積"此程式就會開始計算圖形的面積，這圓的面積我們把它設成"π平方公分"， 跑完之後下面有，下面有三個格子，左邊的是下和，右邊的是上和，中間的是圓的面積，最下面的兩個格子是內和 and 外和的誤差率，完成之後如果想要繼續測試的話，請按下"清除"重新選擇格子大小，則否請按下"結束"關閉此程式。

(範例二)、圓形(FLASH) ↓



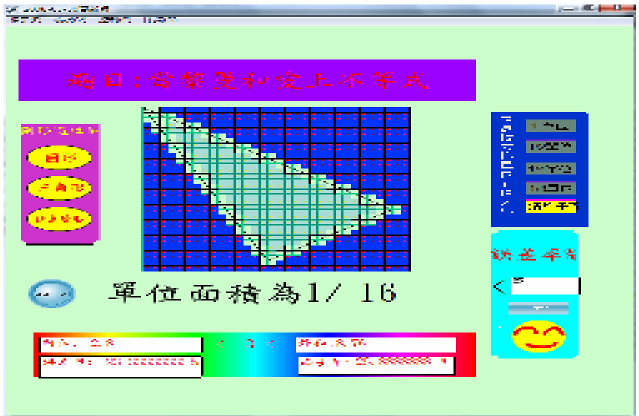
首先請按一下左邊三個選項的"圓形"，就會跑出一個圓形在圖面上，然後選擇則右邊的"方格單位圓面積大小"，選好之後請按下面的"ENTER"開始進行面積的計算，這圓的面積我們把它設成"π平方公分"，下面有四個格子分別為內和、外和、誤差值，這四個格子可以看出計算圓的面積大小和誤差多少，當算完面積時，可以按下"清除畫面"然後再選擇你想要算圓形的面積方格單位大小，我們有 1 單位、1/2 單位、1/4 單位、1/8 單位，要控制誤差率的大小，請在"誤差率%"這個格子填上想要設定的誤差率，方格越小(誤差率越小)，計算出來的面積就會越來越準確!~

(範例三)、三角形(V.B.)↓



首先請先再右上角的六個綠色的格子設三角形的三個頂點(X，Y)(以上設的示範例)，設完之後請按下" 令三點連線" ，再左邊灰色的圖上就會出現一個三角形，然後按下" 求出三線方程式" ，就會出現三條直線的方程式在左下角的 L1、L2、L3 這三條方程式，然後再三角形內點一下滑鼠的左鍵，就會出現判斷的區域，在另外三個格子裡面，然後按下" 計算面積" 就會開始計算三角形的內和面積、外和面積和誤差率，計算完之後，要算更準確面積請按" 更精準" ，則否可以按下" 清除" ，就可以重新輸入你想要算的三角形面積的頂點!~

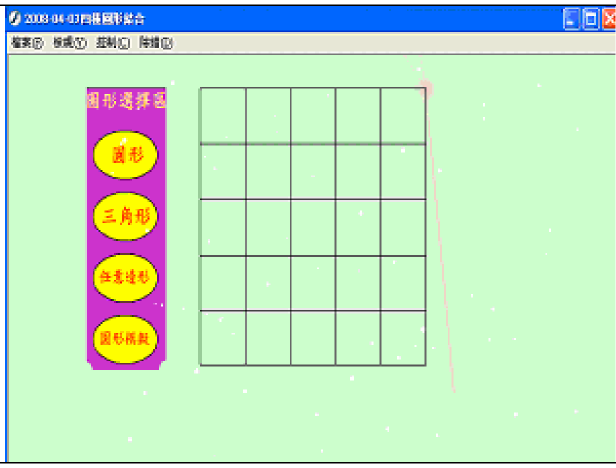
(範例四)、三角形(FLASH)↓



首先請按一下左邊三個選項的" 三角形" ，就會跑出一個三角形在圖面上，然後選擇則右邊的方格單位圓面積大小，選好之後請按下面的" ENTER" 開始進行面積的計算，這圓的面積我們把它設成" 3 平方公分" ，下面有四個格子分別為內和、外和、誤差值，這四個格子可以看出計算圓的面積大小和誤差多少，當算完面積時，可以按下" 清除畫面" 然後再選擇你想要算圓形的面積方格單位大小，我們有 1 單位、1/2 單位、1/4 單位、1/8 單位，如果你要控制誤差率的大小，請在" 誤差率%" 這個格子填上你想要設定的誤差率，方格越小(誤差率越小)，計算出來的面積就會越來越準確!~

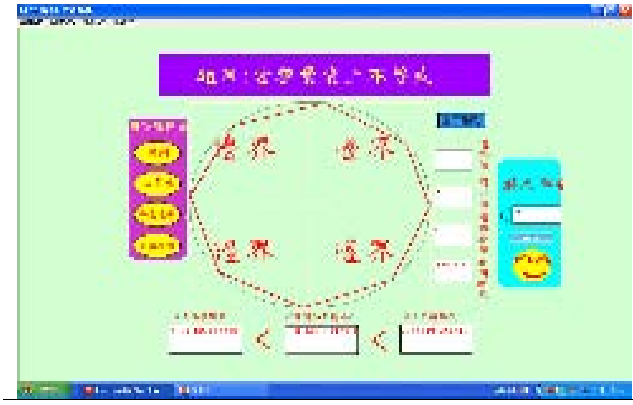
右圖Flash軟體使用：

目前Flash的程式圓形、三角形、任意多邊形、圓形模擬都是由左圖的Flash顯現的，左圖程式是由四個圖形合併的，如果想算圓形的面積請按”圓形”，想要算三角形請按”三角形”，想要算任意邊形請按”任意邊形”，想要算圓形模擬的圖形請按”圓形模擬”。



範例 凸8邊形

格子1x1達誤差10%↓



範例 凸8邊形

格子1/2x1/2達誤差5%↓

範例 凸8邊形

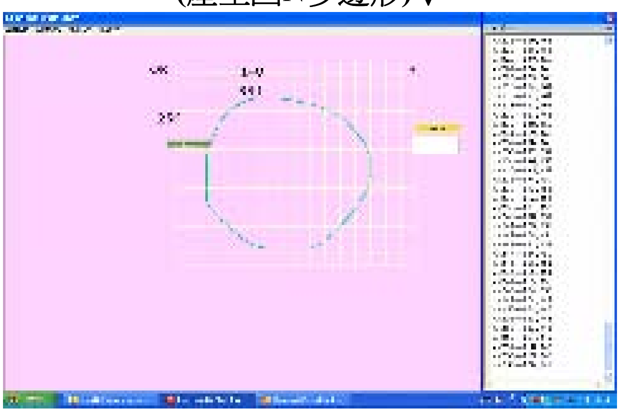
格子1/4x1/4達誤差1%↓

範例 凸8邊形

格子1/8x1/8達誤差0.5%↓

任意曲線範例1

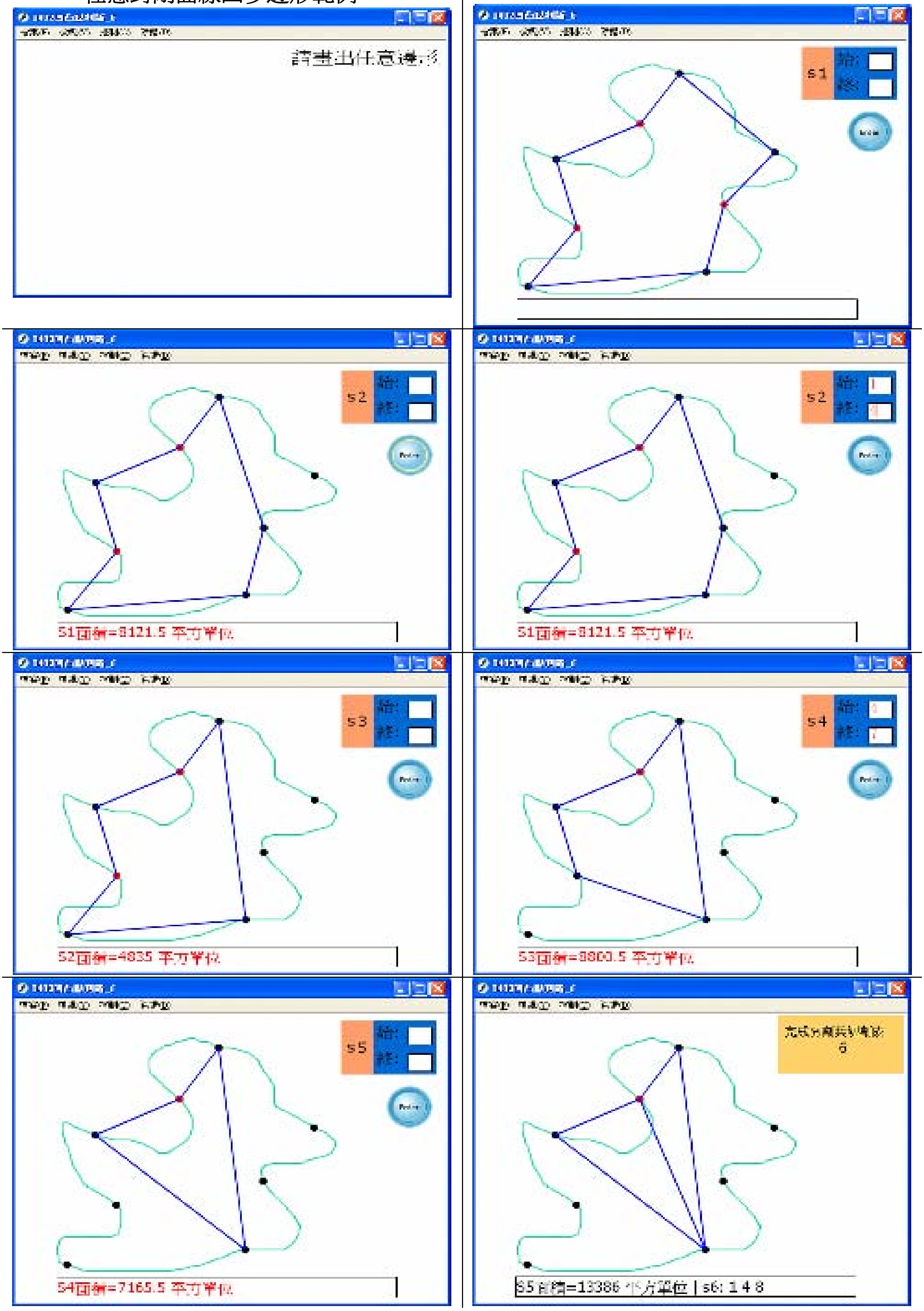
(產生凸N多邊形)↓



任意曲線範例1

(產生面積)↓

任意封閉曲線凹多邊形範例：



陸、討論

一、組員的工作劃分：

我們由一位同學負責用” Microsoft Visual Basic 6.0”軟體來完成此程式，另外二位同學負責用” Macromedia Flash 8” 軟體來完成此程式，最後一位同學負責寫作、搜尋、統整分析資料，每個組員都各自協調選定。

二、問題討論：

(一)問題 1：數學傳播 21 卷 2 期民 86 年 6 月(從醉月湖的面積談起) 提及

在平面上，一條封閉曲線所圍成的區域，例如台大的醉月湖，如何求它的面積呢？

方法：

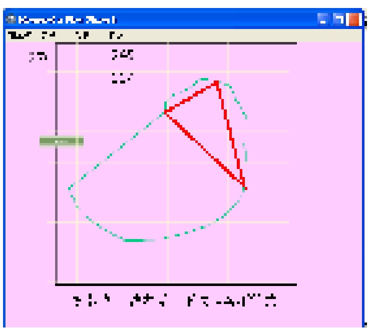
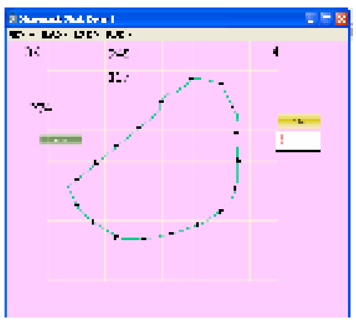
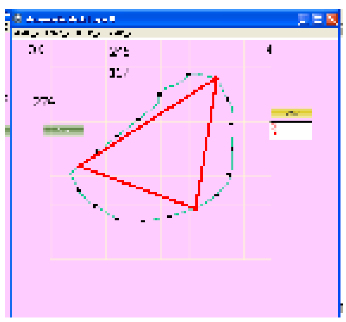
按思考的常理，我們先退到比較簡單的特例，譬如說透過離散化或有窮化，退到多邊形，再退到四邊形乃至三角形。

對於三角形的情形，如果所給的數據是三個邊之長，那麼其面積有 Heron 公式可循。推廣到四邊形的情形，如果所給的數據是四邊之長加上兩對角線或兩個對角，那麼其面積又有 Brahmagupta 公式與 Bretschneider 公式可算。四邊形的面積公式已經有點繁瑣，如果要再推廣到五邊以上的多邊形，其困難是可以想像得到的，甚至根本行不通。一個求面積公式，若只能對付三角形或四邊形，那麼也太侷限了，不合數學追尋普遍的”萬人敵”之道。

(二)問題 2：如何得到代表醉月湖的點座標群 $(x_1,y_1),(x_2,y_2),\cdots\cdots,(x_n,y_n)$?

自製手繪板，於板上畫出一封閉曲線，因滑鼠事件受移動速度產生取點不均現象，電腦取 N 點座標過程，會造成分點之路徑長度會長短不一，故取 N 點之軌跡點模擬封閉曲線會造成過大誤差。

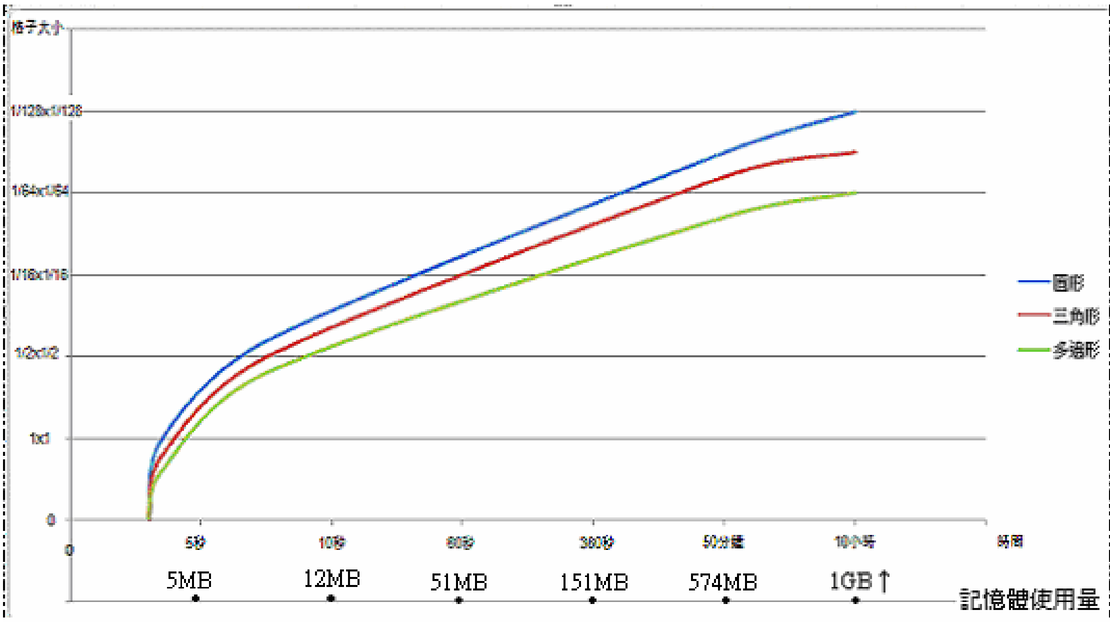
解決方法：放棄手繪板原始紀錄點 $P_1,P_2\cdots,P_K$ 。沿同軌跡透過電腦重新繞手繪封閉區線一圈固定時間距再取點，可得均勻 $P_1,P_2\cdots,\ P_{k2}$ 。

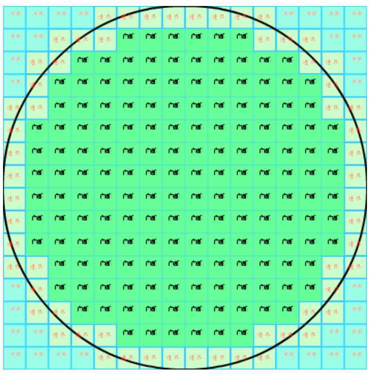
(三)問題 3：為何電腦執行到最後好像都會 Lag？

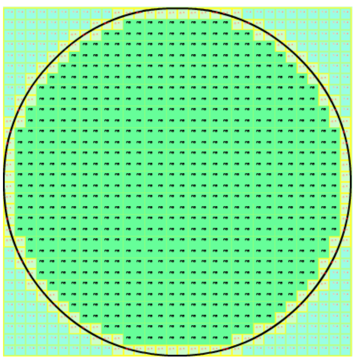
我們給誤差率給的越小，花費的時間也就越多，給 10%誤差率第一次花了 27 分鐘，第二次花了 35 分鐘，所以時間會因電腦執行其他軟體因素有大誤差。

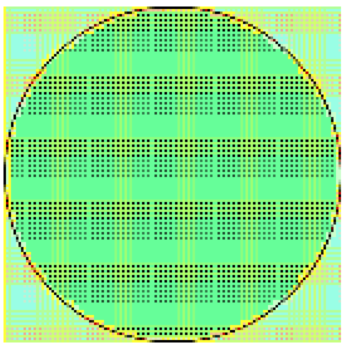
記憶體分析和時間分析



我們發現此圖格子點隨$(\frac{1}{2})^n$變小記憶體需求就會隨著$\sum_{n=1}^{N} k(2)^{2n}$增加，這是因為電腦運算時格子點會吃記憶體，我們使用的電腦記憶體容量約 1024MB，當我們做到格子 1/64x1/64 時，開啓工作管理員看處理程序，發現 FLASH 軟體已經吃了 93%記憶體容量，記憶體容量只剩下 10~5MB 可以使用，記憶體已呈現不足，故得利用硬碟來虛擬記憶體，但因硬碟資料交換速度相當緩慢，故跑格子的速度變得非常的慢。又加上是 vista 作業系統開機需要占大量的記憶體容量，所以要運算更精密的誤差率記憶體容量一定要夠大，CPU 運算能力也要夠快。







(四)問題 4 ：線性規劃之聯立不等式能代表凹多邊形嗎?

當 4 人研究小組，完成任一邊形，歡欣鼓舞之時，因為好心情畫個心算算看，30 日深夜 10 點突然傳來程式錯誤的噩耗，方發覺 N 條之線段來詮釋之 N 多邊形可能為凸多邊形或凹多邊形。方知線性規劃格子點求面積法居然僅適用於前者，若為凹多邊形，則線性規劃格子點面積法，便會產生錯誤。惡耗！！！！

該如何解決？

Setp1: 電子畫板，於板上畫出一封閉曲線，由使用者要求誤差範圍條件下，電腦判定出需要之 N 點座標，沿著封閉曲線繞一圈，固定的時間間格產生座標點，以 N 條之線段來模擬此任意形狀封閉曲線。

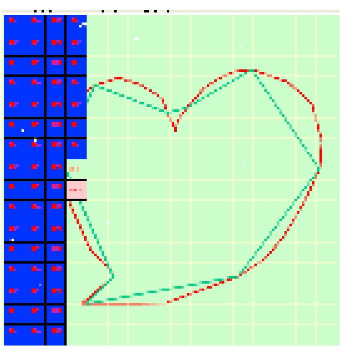
Step2：判斷 N 條之線段來詮釋之 N 多邊形判斷凸多邊形或凹多邊形。

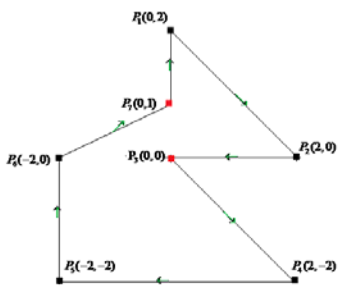
判斷方法：已知假設 N 多邊形之頂點依 $P_1,P_2,P_3,\cdots,P_n$ **順時針**排序，若依序繞一圈比方

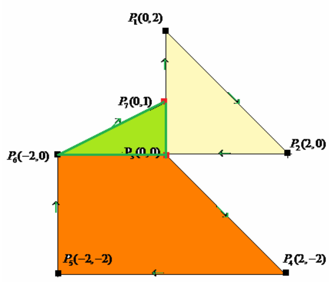
$\mathrm{P}_1(0,2)\to\mathrm{P}_2(2,0)\to\mathrm{P}_3(0,0)$ 過 P_2 時為肉眼判斷順時針旋轉，電腦判斷類似三角形面積公式 $\frac{1}{2}\begin{vmatrix}0&2&0&0\\2&0&0&2\end{vmatrix}$<0 故電腦可知其**順時針**轉到目的，同 $P_1,P_2,P_3,\cdots,P_n$ 走向，故 P_2 為凸點，同理 P_4,P_5,P_6,P_1 為凸點。

$\mathrm{P}_2(2,0)\to\mathrm{P}_3(0,0)\to\mathrm{P}_1(2,-2)$ 過 P_3 時為肉眼判斷逆時針旋轉，電腦判斷方式類似三角形面積公式 $\frac{1}{2}\begin{vmatrix}2&0&2&2\\0&0&-2&0\end{vmatrix}$>0 故電腦可知其**逆時針**轉，和 $P_1,P_2,P_3,\cdots,P_n$ 走向相反故 P_3 為凹點，同理 P_7 為凹點。

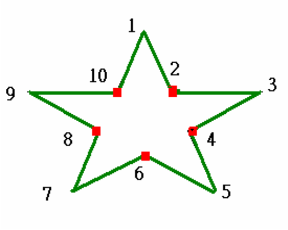
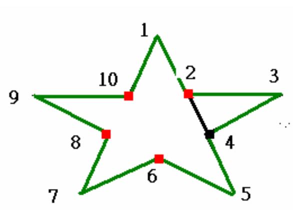
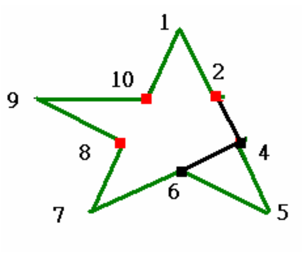
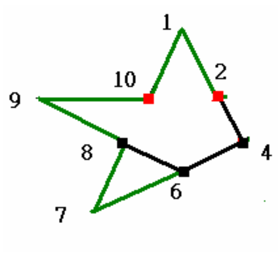
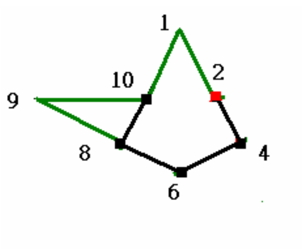
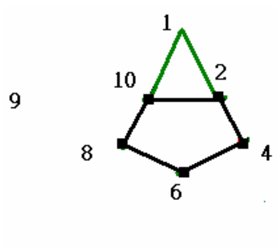
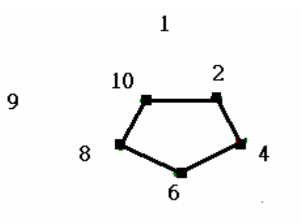
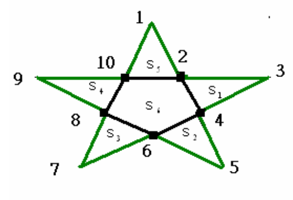
解決方法：若要處理凹多邊形必需先將凹多邊形化成多凸多邊形的組合，若是全為凸點直接點選開始計算面積；若出現凹點必需消滅凹點，連接 $\overline{p_3p_6}$ ，$\overline{p_3p_7}$ ，透過此法將面積分割，使得各區塊之多邊形皆為凸多邊形，使用者要求誤差自動決定計算方格子大小或直接指定方格大小，開始計算各區塊多邊形中格子所表示內部區塊與邊界區塊數目來估計面積。







（五）問題 5 ：凹多邊形電腦如何切割成多個凸多邊形

	星星 由凹點 2 出發 逐一消滅紅點 凹點數目： 5 12345678910		Section1 始點 2 終點 4 S_1={2, 3 , 4} 消滅點 ：3 凹點數目：4 12~~3~~45678910
	Section2 始點 4 終點 6 S_2={4, 5 , 6} 消滅點 ：5 凹點數目：3 12~~3~~4~~5~~678910		Section3 始點 6 終點 8 S_3={6, 7 , 8} 消滅點 ：7 凹點數目：2 12~~3~~4~~5~~6~~7~~8910
	Section4 始點 8 終點 10 S_4={8, 9 , 10} 消滅點 ：9 凹點數目：1 12~~3~~4~~5~~6~~7~~8~~9~~10		Section5 始點 8 終點 10 S_5={8, 9 , 10} 消滅點 ：1 凹點數目：0 ~~1~~2~~3~~4~~5~~6~~7~~8~~9~~10
	Section6 凹點數目：0 自動完成剩餘點 S_6={2,4,6,8,10} 消滅點 ~~1~~2~~3~~4~~5~~6~~7~~8~~9~~10	 $S = S_1 + S_2 + S_3 + S_4 + S_5 + S_6$	

（六）問題 6：凹多邊形經過分割會有更多誤差？

答案是肯定的!但因為凹多邊形的凹點是有限的，切割出的凸多邊形數目也是有限的，但我們格子點理論上是可以無限小的，故此誤差只要格子點夠小便得以已控制。

(七)問題 7：怎知道我們求出來的封閉曲線面積是對的呢？

今天模擬面試，面試老師提出一個疑問，以此方法求任意形狀面積不錯，但你怎麼證明您的面積算出來是對的？您們的理論基礎是什麼？ 傻眼!!!不是 N 夠多就可以嗎？還要什麼理論？

以 N 多邊形模擬任意封閉曲線的面積,若取得之點坐標爲(x_1,y_1) , (x_2,y_2) ,⋯ , (x_n,y_n)；其 N 多邊形所圍的區域=$\frac{1}{2}\begin{vmatrix} x_1y_2+x_2y_3+x_3y_4+\cdots+x_{n-1}y_n+x_ny_n \\ -x_2y_1-x_3y_2-x_4y_3-\cdots-x_ny_{n-1}-x_1y_n \end{vmatrix}$

任意封閉曲線真實面積 $Area$ 爲何？

在此論文本身無法透過任何公式得到真實封閉曲線面積，但對於使用者所要求的任何輸入誤差值，使用格子面積法估算真實面積 Area，

Q1：試問我們怎會知道 N 該取多少？

Q2：沒有真實面積，假設要求誤差 $<E_i=10\%$ **，如何知道格子點要取多小**才能控制真實面積格子和面積誤差在十個百分點？

(八)問題 8：任意形狀可以直接使用此 N 邊形的面積公式進行誤差分析？

答案：可以！

產生凸(凹)N 多邊形過程，由使用者輸入誤差 Ei (比方 5%)條件下，程式透過**呼叫多邊形模擬圓之程式**，以百倍精度 $Ei\times(10^{-2})\times100\%$ 的條件**產生 N 值**，因而封閉曲線得以控制誤差萬分之 5 條件下產生 N 多邊形，取代原任意封閉曲線。

(九)問題 9：電子畫板，畫出曲線若使用者換逆時針畫也可以嗎？

答案：不行！

電腦需要之 N 點座標，沿著封閉曲線**順時針**繞一圈，以 N 條之線段來模擬此任意形狀封閉曲線。判斷凸多邊形或凹多邊形。得依 $P_1,P_2,P_3,\cdots,P_n$ **順時針**排序，透過**判斷逆方向三連續點**的方法找出其**凹點**。

(十)問題 10：能針對較複雜的情形作進一步的分析嗎？

答案：可以！

FLASH 已經解決這個部份，V.B.也已經解決判斷凹點問題的圖形，因此這兩套作品都可以求不規則圖形面積，我們也可以利用圓型、正三角形、正方形、正五邊形等來模擬不規則圖形，只要點取的夠多，利用行列式來進行演算，一樣可以算出真實的面積

(十一)問題 11：是否能展示程式原創性？

答案：可以！

我們願意公開程式碼供教學研究，並可以當場解說程式的演算法，展現程式的原創性。

(十二)問題 12：刻度是否可讓使用者輸入？

答案：可以！

使用者可以自行用滑鼠來畫一個不規則圖形，畫完之後，且可以輸入你想要取幾個點來進行面積的演算，點取的越多，相對的面積就會越精確，這些都可以讓使用者自行進行操作的，增加了使用者的彈性度。

(十三)問題 13：中區科展時，教授提及難道 V.B. 做不到與 Flash 同樣的事情嗎？

答案：可以！

只是礙於畫面無法放大，所以切割越小，肉眼無法看到。V.B.已成功判斷出凹點的部份。

(十四)問題 14：V.B. 程式執行可以做到 AUTORUN 嗎？

答案：可以！

只要把所有點的重心判斷出來，將此點帶入方程式。

(十五)問題 15：Flash 程式該如何判斷凸邊形的內部點？

將各點的 X 座標相加除以總點數，Y 座標相加除以總點數，以求出該內部點。

(十六)問題 16：Flash 程式執行可以做到 AUTORUN 嗎？

答案：可以！

只要把所有點的重心判斷出來，將此點帶入方程式。

(十七)問題 17：Flash 圖形畫的太小，則無法判斷，該如何解決問題？

答案：本來設定一個格子有四個判斷點，現在把格子設成一個格子判斷八個點，增大格子判斷區域，降低誤差率，使得面積更為準確。

柒、 結論

(一)已知函數之封閉曲線面積求法：

大致可利用 $f(x,y)<0$ 之特性，判斷出格子方塊之四頂點，爲內部點 (外部點或邊界點)。加上這些例子 $x^2+y^2=r^2$； $ax^2+by^2=c$ ；$|ax|+|by|=c$ ，可直接利用代數幾何求出真實面積。因此可輕易完成誤差和格子點細度分析。

(二)任意封閉曲線面積求法：

Step1：**產生凸(凹)N 多邊形**。藉由取動點沿邊界導引一周，加入時間因子等時間距取點，由使用者輸入誤差 Ei (比方 5%)條件下，程式透過呼叫多邊形模擬圓之程式，以百倍精度 $Ei\times(10^{-2})\times 100\%$ 條件，產生 N 值，因而任意封閉曲線得以控制誤差萬分之 5 條件下用 N 多邊形取代之。

Step2：**任意封閉曲線之誤差控制**：由輸入之 Ei 值 (5%爲例)，取 N 邊形過程，誤差率控制達到萬分之五決定出之 N 多邊形，足以取代原任意封閉曲線。

(三)礙於非高速電腦採用家用電腦，礙於執行時間過長無法提供更精密的誤差分析數據。原始碼隨小論文呈上，製作過程可說是在嘗試錯誤中一路走過來；承如老師所言「**再縝密的思緒，亦難避免掉掛萬漏一**」。學識知識永無止境，當思考越多變發覺該修改精進的越多，報告到此先作階段的結束。

(四)未來可發展方向：

1. 本作品的程式呈現方式亦可完全的符合傳統黎曼和定義，便可以應用在微積分教學領域；例如數學課堂上的黎曼和定義解說和實作。
2. 若本作品結合地理方面的運用，可以算台灣總面積，各國的領土面積。
3. 雖然當下在雙核心電腦執行此程式，但是還是只有分配到單核心執行我們寫的程式要想辦法分工給各個 CPU 去執行。因應未來的四核心、八核心以上的電腦，希望未來可以將程式碼平行化處理可以獲得更有效的計算效率。

捌、參考資料及其它

1. 劉宇陽(2007 年)FLASH ACTIONSCRIPT 3.0 程式設計入門
2. 林新德(2007 年)FLASH CS3 ACTIONSCRIPT 3.0 應用程式設計
3. 蔡哲明(2006 年)FLASH 8 ACTIONSCRIPT 2.0 與 RIA 應用程式設計
4. 從醉月湖的面積談起: 向量微積分簡介(數學傳播 21 卷 2 期民 86 年 6 月)
5. 工職數學第二冊數列級數；第三冊直線方程式、不等式與線性規劃；第四冊，數列之極限以及定積分與黎曼和。
6. 徐昇多邊形網法之數值計算。
7. 王國榮(1999 年)VISUAL BASIC 6.0 實戰講座。

程式碼(Flash 不規則圖形)：

影格 1：

```
stop();
var z=50;                //方格大小
var r=250;
xmax=0
ymax=0
for (xfirst=1;xfirst<=ain;xfirst++)
{xmax=xmax+nowxarr[xfirst]}
for (yfirst=1;yfirst<=ain;yfirst++)
{ymax=ymax+nowyarr[yfirst]}
gotoandplay(2);
```

影格 2：

```
zx=xmax/ain
zy=ymax/ain
var s=15
var x2=100;y2=100;
var x1=x2;y1=y2;          //封閉區域起點
var leng=500;             //封閉區域長度
var inbox=0;
var mid=0;
var mc1 = createEmptyMovieClip("rect1", 10);
mc1.lineStyle(2,0xFFFFcc,100); //線之粗,顏色,透明度
mc1.moveTo(x2,y2);
mc1.lineTo(x2+leng,y2);
mc1.lineTo(x2+leng,y2+leng);
mc1.lineTo(x2,y2+leng);
mc1.lineTo(x2,y2);
mc1.endFill();
createSquare(x2,y2,z,z);
function createSquare(x2,y2,b,interval){
var mc3 = createEmptyMovieClip("square1", 11);
mc3.lineStyle(2,0xFFFFcc,100);
while(b<leng){
mc3.moveTo(x2+b, y2);
mc3.lineTo(x2+b, y2+leng);
mc3.moveTo(x2, y2+b);
```

```
mc3.lineTo(x2+leng, y2+b);
mc3.endFill();
b=b+interval;
}}
```

影格 3：

```
for (m=1;m<a;m=m+d) {
const=dx[m]*yarr2[m]-dy[m]*xarr2[m];
if(dy[m]<0)
{dy[m]=-dy[m];dx[m]=-dx[m];const=-const;}
if(dy[m]==0 and dx[m] >0)   //dx[m]之前有負號
{dx[m]=-dx[m];const=-const;}
v=dy[m]*zx-dx[m]*zy+const;
if( v >= 0)
{sign[m]=1;
}
else
{sign[m]=-1;
}
if(sign[m]==1) {
}
else
{
}
}
```

影格 4：

```
pos=z/2;
if(x2<x1+leng){
pprt=1;qprt=1;rprt=1;sprt=1;
px=x2;py=y2;qx=x2+z;qy=y2;rx=x2+z;ry=y2+z;sx=x2;sy=y2+z;
tx=x2+(z/2);ty=y2;ux=x2+z;uy=y2+(z/2);vx=x2+(z/2);vy=x2+z;wx=x2;wy=y2+(z/2)
for (m=1;m<a;m=m+d) {
val=dy[m]*px-dx[m]*py+dx[m]*yarr2[m]-dy[m]*xarr2[m];
cc=dx[m]*yarr2[m]-dy[m]*xarr2[m];
if(sign[m]==1){
if(val >= 0 )
{ prt[m]=1 } //不等式成立
else {prt[m]=2 } //不成立
}
```

```
else{
if(val <= 0 )
{ prt[m]=1 }//不等式成立
else{prt[m]=2 } //不成立
}
pprt=pprt*prt[m];
}
if(pprt!=1){pprt=2;}
for (m=1;m<a;m=m+d) {
val=dy[m]*qx-dx[m]*qy+dx[m]*yarr2[m]-dy[m]*xarr2[m];
cc=dx[m]*yarr2[m]-dy[m]*xarr2[m];
if(sign[m]==1){
if(val >= 0 )
{ prt[m]=1 } //不等式成立
else{prt[m]=2 } //不成立
}
else{
if(val <= 0 )
{ prt[m]=1 }//不等式成立
else{prt[m]=2 } //不成立
}
qprt=qprt*prt[m];
}
if(qprt!=1){qprt=2;}
for (m=1;m<a;m=m+d) {
val=dy[m]*rx-dx[m]*ry+dx[m]*yarr2[m]-dy[m]*xarr2[m];
cc=dx[m]*yarr2[m]-dy[m]*xarr2[m];
if(sign[m]==1){
if(val >= 0 )
{ prt[m]=1 } //不等式成立
else{prt[m]=2 } //不成立
}
else{
if(val <= 0 )
{ prt[m]=1 }//不等式成立
else{prt[m]=2 } //不成立
}
rprt=rprt*prt[m];
```

```
}
if(rprt!=1){rprt=2;}
for (m=1;m<a;m=m+d) {
val=dy[m]*sx-dx[m]*sy+dx[m]*yarr2[m]-dy[m]*xarr2[m];
cc=dx[m]*yarr2[m]-dy[m]*xarr2[m];
if(sign[m]==1){
if(val >= 0 )
{ prt[m]=1 } //不等式成立
else {prt[m]=2 } //不成立
}
else {
if(val <= 0 )
{ prt[m]=1 }//不等式成立
else {prt[m]=2 } //不成立
}
sprt=sprt*prt[m];
}
if(sprt!=1){sprt=2;}
if(pprt*qprt*rprt*sprt==1)
{_root.attachMovie("aaa", "liveFace"+s,s);
setProperty("liveFace"+s,_height,z);
setProperty("liveFace"+s,_width,z);
setProperty("liveFace"+s, _x,x2+pos);
setProperty("liveFace"+s, _y,y2+pos);
inbox++;
}
else if(pprt*qprt*rprt*sprt==16)
{_root.attachMovie("c", "liveFace"+s,s);
setProperty("liveFace"+s,_height,z);
setProperty("liveFace"+s,_width,z);
setProperty("liveFace"+s, _x,x2+pos);
setProperty("liveFace"+s, _y,y2+pos);
}
else {_root.attachMovie("middle", "liveFace"+s,s);
setProperty("liveFace"+s,_height,z);
setProperty("liveFace"+s,_width,z);
setProperty("liveFace"+s, _x,x2+pos);
setProperty("liveFace"+s, _y,y2+pos);
```

```
mid++
}
}
var ss=0;
for(m=1;m<=nowdot;m++)
{   h=m+1;
if(h>nowdot) {h=1;}
ss = ss + nowxarr[m]*nowyarr[h];
ss = ss - nowyarr[m]*nowxarr[h];}
inner=inbox*z*z/(r*r)
outter=mid*z*z/(r*r)
total=parseFloat(inner)+parseFloat(outter)
ss=(ss*0.5)/(r*r)
sst=Math.abs(ss-inner)/ss*100
sstt=Math.abs(total-ss)/ss*100
```

影格 5：

```
if(y2<y1+leng and x2<x1+leng){
y2=z + y2;
if (y2==y1+leng)
{x2=z + x2; y2=y1}
s=s+1;   //顯示層
gotoandplay(4);}
else
{
gotoandplay(6);}
```

影格 6：

```
stop();
```

完整程式請看附件光碟

【評語】091004

1. 將資訊軟體設計結合數學運用是個不錯的概念。
2. 四位學生各負責不同的封閉區線求解面積的軌體設計，分工合作相當理想。
3. 四位學生的語言表達及實務操作相當流暢、熟練。