科學計算與中學數學

李錦鎣

國立彰化師範大學 數學系

壹、前言

隨著時代發展，中學數學教育逐漸強調「素養」、「跨領域」與「實作」這些要素，這也是 108 年數學新課綱強調的幾個新元素。在本文中筆者將適度結合此三要素，選擇一個不同的切入角度與教學方式，提供第一線中學教師經由結合「中學數學」與「科學計算」內容這一策略來發展新的教學模組，可以作為 108 年數學新課綱實施後，高中選修課程及加深加廣課程的另一種選擇。在此，作者特別感謝教育部師藝司對完成本文所提供的各項協助，以及對彰化師範大學數學領域教學研究中心的支持與鼓勵。

貳、Maxima 軟體與科學計算示例

在中學數學中有許多計算問題，例如：二次或三次函數畫圖、十分逼近法或二分法求方程式的解、整數解的線性規劃，甚至是涉及大量數據的統計問題，在缺少計算工具與科學計算方法協助時，這些重要實作過程在課堂上很難順利進行。另一方面，雖然這些電腦時代中成長的學生，將來未必每一個人都會進入資訊產業，但是他們仍然應該具備基本現代資訊能力。上述兩個問題是可以經由適當課程設計來解決。我們可以嘗試在中學階段，藉著提出數學及其他學科中的具體問題，運用適當數學策略與科學計算方法，利用電腦與數學軟體進行模擬、實驗與觀察，進而提出假設與猜測。這不但能培養數學發現與實作之能力，減少過往在課堂上告知式教學的比例，更能夠逐步培養學生程式設計基本資訊能力。在此特別強調，此課程內容是配合新課綱的跨領域及實作目標，並著重於奠定基本科學計算與程式設計觀念與能力，所以此課程內容不要求教師在課堂上立刻給予大量相關數學論證，反而要以數學實驗與數學模擬於具體事例中實現，作為培養數學發現能力與素養的一種切入方式。這不是傳統課堂教學所能進行的教學方式，這提供高中數學教師一種不同於傳統課堂教學的新選擇。

然而，資訊學科常用的 C 語言，對多數數學專業的中學教師來說並非擅長之工具，而且對多數中學生來說，認識 C 語言嚴謹架構，並不是理解科學計算的第一要務，因為有許多專業數學軟體可以幫助我們解決此困境。

與 mathematica 及 matlab 這一類需付費的數學軟體相比，網路上有一套完全免費的

maxima 軟體可以下載。它包含電腦版與手機版，這將能大幅度解決所有學校與同學在經濟上不必要的採購負擔 。maxima 是依據麻省理工學院所研發的 Macsyma 軟體所發展出來，具有極強符號運算與科學計算之功能， 非常適合數學老師在科學計算上使用，也方便讓中學生與大學生快速理解數學策略如何與演算法結合，進而完成必要程式設計與大量科學計算。這對於將來不使用 C 語言，只需使用 matlab 或 mathematica 這一類更強大數學軟體的理工或商管科系同學來說，不僅能順利銜接未來專業科學計算之需求，更可以提早培養中學生應有的資訊素養與基礎能力。

現在為讀者介紹電腦版 maxima 下載與安裝方法，讀者只需在電腦上開啟 Google Chrome，搜尋 maxima，就會看到 「maxima,a Computer Algebra System」再依下列步驟進行 ：

(1) 進入該頁後，點選左上角的 Downloads ,(請注意:使用 Windows IE 似乎無法完成剩下的步驟)。
(2) 進入該頁後，再點選中間 Windows 旁的藍字 Installation。
(3) 進入該頁後，然後點選畫面左方第 1-6 點中的第 1 點中央的藍字 5.42.2-Windows(這是此時的最新版本)。
(4) 最後進入該頁後，再點選藍字 maxima-clisp-sbcl-5.42.2-win64.exe 來下載。
(5) 等左下方下載提醒時間結束後，就可以開啟進行安裝了。
(6) 安裝完後，點選那個中間有紅字 M 的圓形圖示就可開啟軟體。
(7) 輸入 1+1 ，再同時按 shift 與 enter 鍵，(如果不想同時按 shift 與 enter 鍵，那就直接按鍵盤最右方數字鍵盤區的那個 enter) 如果跳出 2，那你應該安裝成功了。

另外，讀者也可以在安卓手機的 Google Play 下載完全免費的手機版 maxima， 即使在沒有電腦的教室中，老師與全體學生都能使用手機版完成所需要的科學計算。

以下，我們將舉幾個使用 maxima 的例子，來具體說明結合中學數學與科學計算的簡單教案。

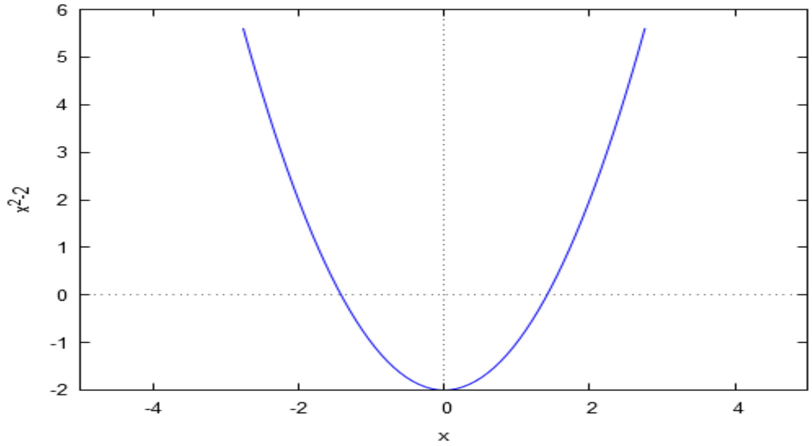
例一：以二分法求方程式 x^2-2=0 的近似解到小數第 6 位

問題：求方程式 $x^2-2=0$ 的近似解到小數第 6 位：

首先，我們將以二分法求方程式 $x^2-2=0$ 的近似解為例子說明 maxima 基本功能與操作。

(1) 先輸入 f(x):=x^2-2；再同時按 shift 與 enter 鍵執行這個指令，然後再輸入一次 f(x)，你看到程式已經記得 f(x)了。
(2) 在上方工作列中選 「繪圖(P)」的二維繪圖，在第一行數式中輸入 f(x) ，並將第三行

的變數 y 的範圍從 0 到 0 ，改成 從-2 到 15， 最後按下確認，你就看到電腦如何快速實現國中數學課本中用描點來畫函數圖形， 當然你也可以在電腦或手機的 maxima 中，直接輸入 wxplot2d(x^2-2,[x,-5,5],[y,-2,15]) ，或 plot2d(x^2-2,[x,-5,5], [y,-2,15]) 就可以看到圖形



(3) 由圖形，我們看出方程式 $x^2-2=0$ 的只一個正解介於 1 與 2 這兩個相鄰整數之間。

(4) 我們再利用數學中「連續函數 f(x)在 a,b 兩點的函數值一正一負時，亦即 $f(a)*f(b)<0$ 時，那麼在 a,b 之間必有$f(x)=0$之解」這個性質與「二分法 」這個演算法，合力完成以下三行程式：

```
f(x):=x^2-2$    a:1.0$    b:2.0$
for  n:1  thru  30  do(
c:(a+b)/2 , if  f(a)*f(c)<0  then  (b:c)  else  (a:c),  print([a,b])       )
```

執行後，不需一秒鐘，你會看到 [a,b] 一次又一次縮小成原有的一半，最後告訴你 $x^2-2=0$的解 $\sqrt{2}$ 是介於 1.414213561452925 與 1.414213562384248 這兩個數之間，所以 $\sqrt{2}$ 約為 1.41421356。

在此，順便說明這三行程式，是如何能夠快速地替你完成這麼單調卻繁複的計算：第一行的第一句 f(x):=x^2-2 這是讓程式認識 f(x)，第二句與第三句是「指定」剛開始時的 a 與 b 分別為 1.0 與 2.0 ，在 maxima 程式中「指定」是用「冒號：」 ，而兩個指令之間用「$」號來斷句。

在第二行中 for n:1 thru 30 do( ) ，這是告知程式我們將重複做一些相似的動作或實驗，每一次的編號是用整數 n 來表示，而整數 n 將從 1 變動到 30 就結束程

式。

在第三行中 c:(a+b)/2 , if f(a)*f(c)<0 then (b:c) else (a:c) ，表示先指定 c 為 a,b 中點，再檢查 f(a)*f(c)<0 是否成立。 若 f(a)*f(c)<0 成立，那新的 b 指定為這個 c 點，否則就將新的 a 指定為這個 c 點，最後印出這一階段由二分法得到的新 [a,b] 區間。

(5) 讀者可以試試用以上方法，求出 $2^x - 3 * x - 2 = 0$ 或是其他方程式解的小數第六位近似值。利用二維繪圖功能，教師在教授三角函數，指數函數，對數函數時，無論在對稱、平移或週期的觀察與討論上也能獲的不同的成效。

以上這個例子，是結合數學課本知識與資訊課程基本程式設計過程的跨領域實作範例。在課堂中用電腦或手機版的 maxima 中輸入短短的這三行指令都可以完成此一教案，這與一般在軟體中直接使用內建解方程式的教學方式不同。 同學此刻不只是看到方程式的解，而是親手結合數學上解方程式的方法與程式設計過程，完成過去在課堂上所無法進行的大量繁複計算，這是數學理論與資訊實作的極佳組合。其他在數學課本上提及的規律性解題策略與操作， 例如輾轉相除法求最大公因數，或是十分法求方程式的近似解，都可以模仿上述過程，在數學課堂上的對應單元進行教學與實作。

例二：模擬、實驗與猜測

問題：超級樂透博弈公司設計一款遊戲如下：

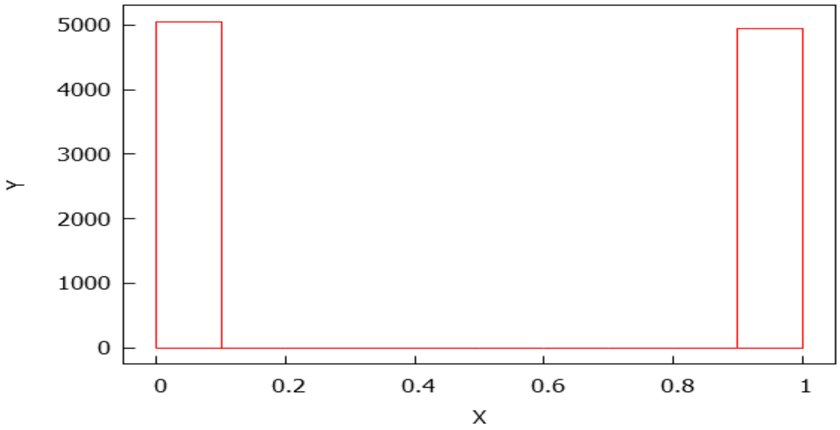
賭徒交了入場費後，他得到基本分數為 0 分，然後賭徒利用丟一個公正銅板來改變分數，正面加 1 分，反面減 1 分，只要他的分數到達 5 分或是-5 分， 遊戲就結束，並由公司依照賭徒在這過程中丟銅板的總次數，付給他相同數量獎金，請問該公司要如何設計入場費金額以維持公司長久的營運。

我們可以藉著 maxima 來模擬這家公司的遊戲經營過程，藉由進行大量實驗，看看該公司如果進入大量賭徒，例如 1000 位賭客時，平均每位賭徒領走多少獎金，作為該公司設計入場費金額之依據與參考。

首先，我們讓丟一個銅板出現正面就用 1 表示，出現反面則用 0 表示，既然是隨機事件，那就使用 maxima 中的 random 指令，讓我們輸入 random(2)後 反覆執行幾次發現隨機出現 0 與 1，但是 random(2)雖然看起來是隨機出現 0 或 1，但是是否符合我們說的「公正」銅板需求呢？也就是 0 與 1 出現機會相同嗎？

我們可以這樣做，輸入並執行 w:makelist(random(2),n,1,10000) ，也就是我們指定 w 是 10000 個隨機的 0 與 1 所成數列，然後點選工作列中 View 那個下拉選單中的「統計」

選項，再點選「統計」面板區中的 「直方圖」 ，並在第一列輸入 w ，你就會看到 0 與 1 出現次數幾乎相同，如下圖，因此確認我們所要的銅板是公正的。



現在，為了看看這 1000 位賭客平均每位賭徒丟銅板的總次數是多少，也就是平均領走多少獎金呢？ 我們可以在 maxima 輸入以下六行：

```
w:[]$
for   m:1   thru   1000   do(
z:0,   for n:1   thru   100000   do(
a:random(2)   , if a=1   then   z:z+1   else   z:z-1,
if ((z=5) or (z=-5))   then   (w:append(w,[n]))   and   (n:100001) ));
mean(w*1.0);
```

不需要多久，畫面跳出數字來，約為 25 ，這表示平均每位賭徒丟銅板的總次數是 25 ，也就是領走 25 元獎金，這可作為該公司設計入場費金額之重要依據與參考。如果該公司負責人不是數學專業人員，經由參考這樣大量實驗所獲得的數據，他應該可以估計出至少要向每一位賭客收 25 元的入場費吧！

在此，我們簡單說明這六行程式是如何能夠快速地替你完成這麼單調卻繁複的模擬實驗：

第一行指定 w 剛開始是一個空白字串。

第二行 for m:1 thru 1000 do ( ) 表示將 1000 個賭徒用 m 來編號，分別進行相同遊戲程序。

第三行的第一句 z:0, 表示當選好第 m 號賭徒上場之後，初始分數 z 指定為 0。

第二句 for n:1 thru 100000 do ( ) 表示讓這位賭徒有儘量多的機會丟銅板， 例如最多讓他丟 100000 回。

第四行的第一句 a:random(2)表示電腦模擬丟一回銅板，可能出現正面 1 或反面 0 ，第二句 if a=1 then z:z+1 else z:z-1, 表示在丟出正面時，指定新的分數 z 是舊分數 z 加 1 ，否則就指定新的分數 z 是舊分數 z 減 1。

第五行 if ((z=5) or (z=-5)) then (w:append(w,[n])) and (n:100001)) 表示分數 z 到達 5 或-5 時，就將這位賭徒所投擲的次數 n 加到原有 w 字串的最後位置， 來造出新的 w 字串，並且讓實驗編號 n 變成 100001 ，由於這個編號數字 100001 超過程式原來設定的合法編號 n 的上限 100000， 所以程式就會自動結束這位賭徒的遊戲了。

第六行 mean(w*1.0) 表示 1000 位賭徒都結束遊戲後，計算它們所丟擲次數的 1000 個數字所形成字串 w 的平均數。

這六行程式讓我們在一個模擬情境中，利用 maxima 的機率與統計工具進行科學計算實驗，也得到這些限制條件下，平均每位賭徒領走 25 元獎金這個實驗結果 如果老師將終止條件改成「z=7 或 $z = -7$」，或是「 $z = 10$ 或 $z = -10$ 」，這些不同條件時，那麼平均每位賭徒又會領走多少獎金呢？ 此時同學可以立刻修改程式並做實驗，讓同學自己觀察實驗結果，並自己提出可能的猜測，這不但會引同學好奇心，也可以避免傳統課堂上告知式教學， 使得同學將來修習相關課程時，不只有具體事例與經驗，將會有更強學習動機。

例三：大量進行同一個互不干擾的隨機實驗

問題：在「1000 層的平行時空」這部科幻電影中，有甲乙兩個賭徒利用丟一個公正銅板來對賭，丟出正面甲就贏乙 1 元，丟出背面甲就輸給乙 1 元， 請問投擲 100 次之後，在這 1000 層的平行時空中，甲的輸贏狀況如何？

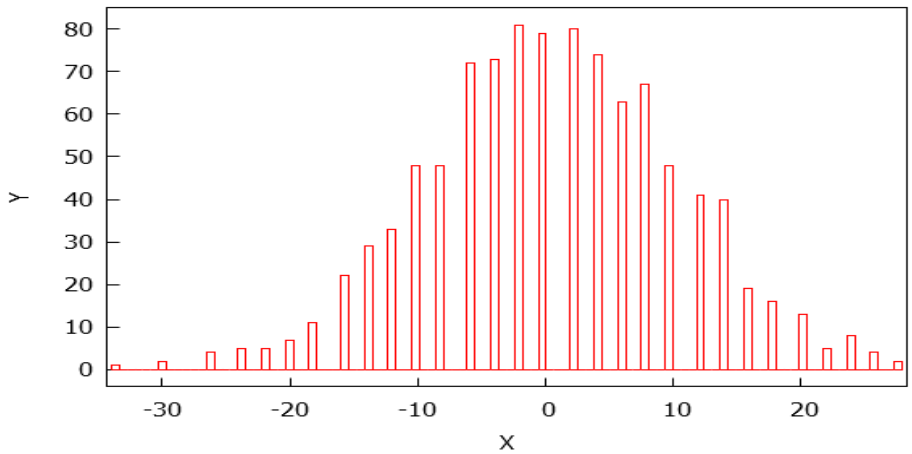
現在我們用 maxima 來模擬這 1000 層平行時空中對賭的過程，
輸入以下 5 行：

```
w:[]$
for i:1 thru 1000 do (
  z:makelist(2*random(2)-1,n,1,100),
  v:sum(z[k],k,1,100),
  w:append(w,[v*1.0]));
```

上方程式只有兩點需稍加說明，其一，2*random(2)-1 表示 1 與 -1 隨機出現且機會相同，所以 z 表示投擲 100 次公正銅板的正反面狀況所成數列；其二，z[k]表示數字字串 z 的第 k 個數字，所以 v 表示投擲 100 次後甲的輸贏金額之總和，因此 w 集合表示這 1000

個不同時空中甲的輸贏總金額集合。

然後在「統計」面板區， 按「直方圖」，並在第一列輸入 w ，就立刻可以看到這 1000 次中甲方的輸贏金額與次數，如下圖，整個圖形像是以 x=0 為對稱軸的鐘形曲線。



如果分別執行「統計」面板區中的平均 mean(w)與偏差 std(w)，大概就會的到平均數為 0 ，而偏差大概為 10 ，此時我們可以算一算在 w 的 1000 筆數字中， 與平均數的差距小於一個標準差的有多少筆數字？占全部多少比例？輸入以下四行：

```
t:0$
for i:1 thru 1000 do (
  if (w[i]-mean(w)<std(w)) and (w[i]-mean(w)>-std(w)) then t:t+1 );
t/1000.0
```

得到 0.685。我們也可以算一算在 w 的 1000 筆數字中，與平均數的差距小於二個標準差的有多少筆數字？占全部多少比例？ 輸入以下四行：

```
t:0$
for i:1 thru 1000 do (
  if ( w[i]-mean(w)<2*std(w)) and (w[i]-mean(w)>-2*std(w)) then t:t+1 );
t/1000.0
```

得到 0.957。此時可以提醒同學 這是相同隨機事件互不干擾地反覆大量進行時，可能會經常看到的圖形與集中分散比例，這可能比直接對同學宣告「常態分布」,「68%」及「95%」這些訊息來的更可親近些。

例四：從規律到混亂

問題：在中學數學課程我們曾經學到等差數列，也就是 給 a_0 以及 $a_{n+1}=a_n+d$,其中 d 為已知數，稱$\{a_n\}$為一個等差數列。我們也可以換另一種表示方式：給 a_0 以及 $a_{n+1}=f(a_n)=a_n+d$，其中 d 為已知數，亦即 f(x)=x+d 這個一次函數是相鄰兩項 a_n 與 a_{n+1} 的關係。同樣地，後來我們又學到的等比數列 $a_{n+1}=f(a_n)=a_n*r$，其中 r 為已知數，就可以用 f(x)=x*r 這個一次函數來表示相鄰兩項 a_n 與 a_{n+1} 的關係。 對於等差數列與等比數列，同學們可以輕易地由公差 d 或公比 r 猜測出這個數列最後將發生什麼情況。然而，如果連接相鄰兩項 a_n 與 a_{n+1} 的關係 f(x)不是一次多項式呢？例如，改成二次多項式呢？此時我們是否依舊可以猜測出這個數列最後將發生什麼情況呢？

例如，我們取 $a_0=0.2$ 且 f(x)=0.3*x*(1-x)，看看 n 夠大，例如 n 介於 1000 與 1020 時，a_n 將發生什麼情況？我們可以讓 maxima 進行實驗：

```
f(x):=0.3*x*(1-x) $    a:0.2$
for n:1 thru 1020 do(
a:f(a) , if n>1000 then print(a))
```

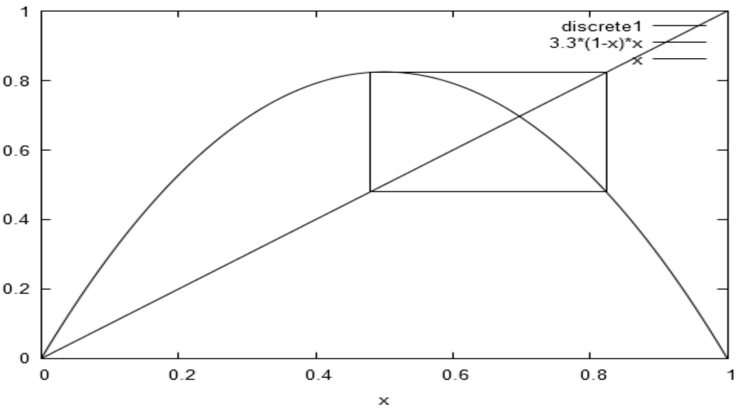
我們看到這個例子中，a_n 最終很接近 0。若將 f(x)改成 f(x)=1.3*x*(1-x) 或是 f(x)=2.3*x*(1-x)，我們會看到 a_n 最終分別接近 0.2307 及 0.5652，這好像暗示連接相鄰兩項 a_n 與 a_{n+1} 的關係 f(x)是二次多項式時，數列 a_n 最終就是會很靠近一個數。但是，如果將 f(x)改成 f(x)=3.3*x*(1-x)：

```
f(x):=3.3*x*(1-x) $    a:0.2$
for n:1 thru 1020 do(
a:f(a) , if n>1000 then print(a))
```

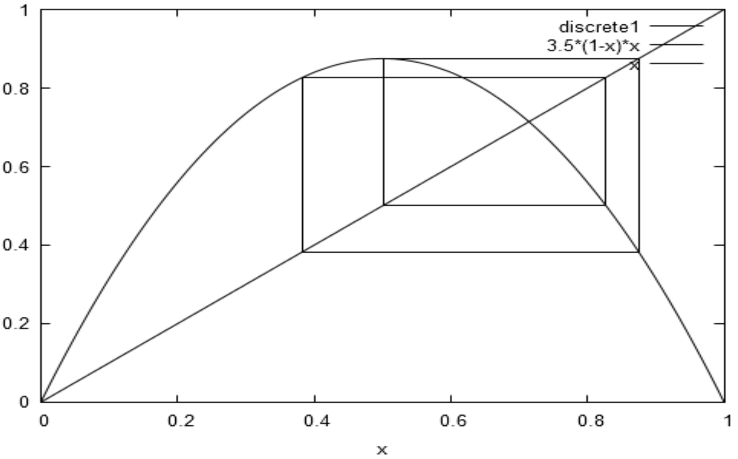
我們將看到 a_n 最終會在 0.4794 與 0.8236 這兩個數字來回不斷跳動。我們也可以利用 maxima 收集 a_n的終極軌跡圖：

```
f(x):=3.3*x*(1-x) $    a:0.2$    w :[]$
for n:1 thru 1020    do(
a:f(a) , if    n>1000    then    (w:append(w,[ [a,a] ] ) ) and (w:append(w,[ [a,f(a) ] ]) ));
wxplot2d([ [ discrete,w] ,f(x),x ] ,[x,0,1],[y,0,1],[style,[ lines,1,5,7 ] ]    )
```

我們從圖形上看到數列 a_n 是如何在 0.4794 與 0.8236 這兩個數字來回跳動。



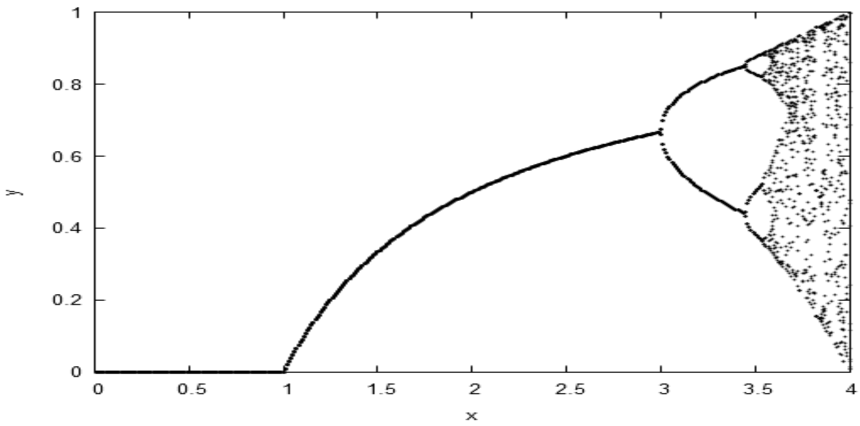
如果將 f(x)改成 f(x)=3.5*x*(1-x) 我們看到卻是：



也就是說，此時數列 a_n，在 n 很大時，是在 0.3828 ，0.8269 ，0.5008 與 0.8749 這四個數字來回走動。如果將 f(x)改成 f(x)=3.55*x*(1-x)，此時數列 a_n ，在 n 很大時，是在 0.5404，0.8816，0.3703，0.8278，0.5060，0.8873，0.3548，0.8126 這 8 個數字來回走動。

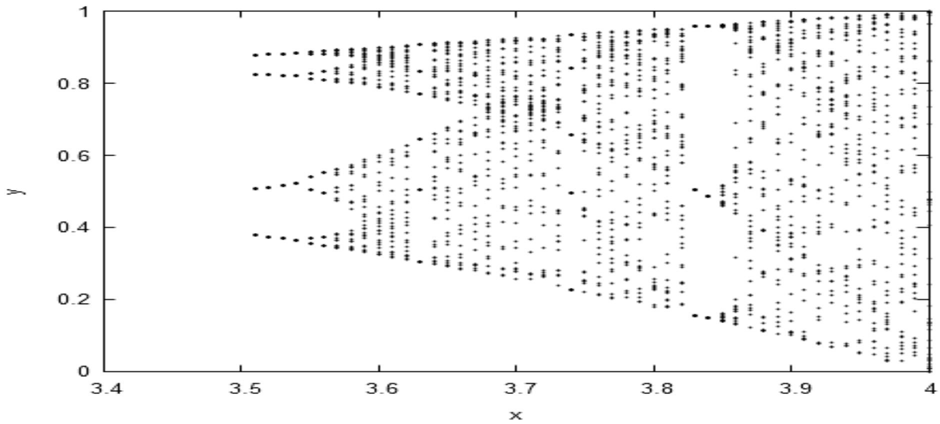
經過這幾次試驗，我們發現連接相鄰兩項 a_n 與 a_{n+1} 的關係 f(x)=k*x*(1-x)的係數 k 會影響數列 a_n 最後將發生的情況，所以我們可以使用 maxima 來觀察 k 連續改變時，對應的數列 a_n 的終極狀態：,

```
f(k,x):=k*x*(1-x) $    w:[ ] $
for m:1 thru 400
  do(     a:0.2,
    for n:1 thru 1020
      do( a:f(0.01*m ,a), if    n>1000    then    (w:append(w,[[0.01*m ,a]]))));
  wxplot2d([discrete,w], [x,0,4],[y,0,1],    [style,    [points,0.3,5,7]]         );
```



在此圖中橫軸為 k 座標，而曲線上的點，就是固定 k 值時，數列 a_n 的終極數字或狀態。例如 k 為 2.3 時，數列 a_n 的終極狀態為 0.5652；又例如 k 為 3.3 時， 數列 a_n 的終極狀態為 0.4794 與 0.8236 這兩個數字來回跳動。我們從圖形中看到 k 介於 3.5 與 4 之間時，圖象很混亂。我們用 maxima 放大這個區塊：

```
f(k,x):=k*x*(1-x) $    w:[ ] $
for m:1 thru 50
  do(     a:0.2,
    for n:1 thru 1050
      do( a:f(3.5+0.01*m ,a), if    n>1000    then (w:append(w,[[3.5+0.01*m ,a]]))));
wxplot2d([discrete,w], [x,3.4,4],[y,0,1],   [style,   [points,0.3,5,7]]        );
```

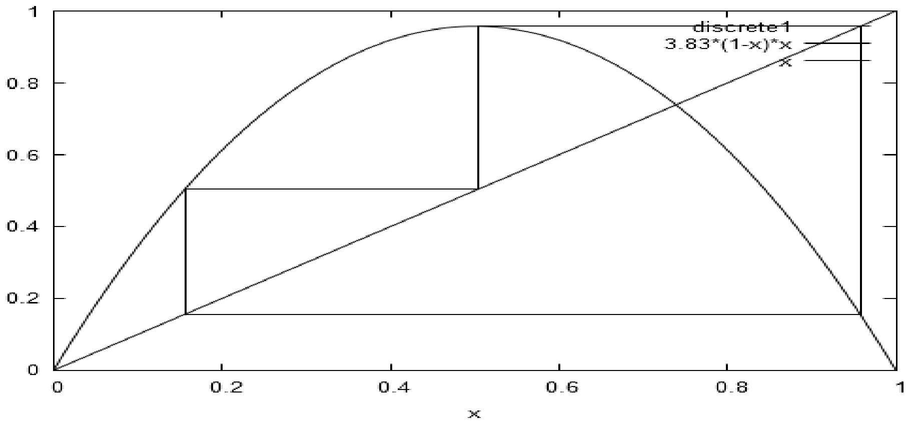


這張圖顯示： k 大於 3.7 時，多數的數列 a_n 的終極狀態，經常是密密麻麻的一些混亂數字，也就是這些數列 a_n 的終極狀態似乎失去規律性了，進入「混沌狀態」。但是，這張圖又提醒我們，在 k 約為 3.83 時，會出現一個特殊的數列 a_n，它的終極狀態將會是只在三個數字來回跳動，我們可以讓 maxima 進行實驗：

```
f(x):=3.83*x*(1-x) $    a:0.2$
for n:1 thru 1020 do(a:f(a) , if n>1000 then print(a))
```

我們就可以找到 a_n 的終極狀態會是在 0.1561，0.5046 與 0.9574 三個數字來回跳動，此時，我們還是可以利用 maxima 收集 a_n的終極軌跡圖:

```
f(x):=3.83*x*(1-x) $    a:0.2$    w :[]$
for n:1 thru 1020    do(
a:f(a) , if    n>1000    then    (w:append(w,[ [a,a] ] ) ) and (w:append(w,[ [a,f(a) ] ]) ));
wxplot2d([ [ discrete,w] ,f(x),x ] ,[x,0,1],[y,0,1],[style,[ lines,1,5,7 ] ]    )
```



這是一個有趣的探索實驗，由國、高中數學的等差數列與等比數列出發，向前跨出一步，經由在數學軟體上進行簡單的程序設計，就能發現許多奇奇怪怪的現象。也許同學會問：那數列最終可能會在五個數字間跳動嗎？如果不是二次函數又會如何呢？甚麼是「混沌狀態」？這就是引發好奇心與學習動機，讓他們試，讓他們觀察，讓他們提出猜測，讓他們研究，也許不急著給答案去限制他們的想法，也是現代老師的另一種選擇。

參、結語：

以上我們舉了四個例子，說明在中學數學教學時，配合適當數學軟體，進行科學計算，不只是可以落實課本中所提到的規律計算方法，也可以進行實驗與模擬，這對跨學科教學與培養學生資訊能力都有很大助益，更符合新課綱所強調的實作精神。

參考文獻

教育部(2018)。**十二年國民基本教育課程綱要國民中小學暨普通型高級中等學校-數學領域**。台北：教育部。