中華民國第 57 屆中小學科學展覽會

作品說明書

高級中等學校組　數學科

最佳創意獎

050401

讓牛頓步上尋根的階梯

—複係數多項方程式的求解與推廣

學校名稱：國立花蓮女子高級中學

作者：	指導老師：
高二 姜亦柔 高二 黃貞瑋 高二 林周郁潔	吳仲奇

關鍵詞：魔術植物栽培算法、牛頓算法

摘 要

一、階梯算法

更改庫恩植物栽培法的限制，使方程式的根形如走階梯般向上攀升，且不下降。

二、牛頓算法的限制

1. 牛頓算法一次只能算一個根，算出多項方程式 $f(z)=0$ 的一個根 ζ 以後，用 $(x-\zeta)$ 除原來的多項式，得到一個階數降 1 的多項式，再用牛頓算法求這個新的多項式根，這樣一次一次降階，可以把全部的根算出來。

2.牛頓算法有時很難控制，且計算是否成功，是沒有保證的。要使牛頓算法成功，重要的是找到一個足夠好的迭代出發點 z_0。

三、比較各個求解方法，結合牛頓算法與階梯算法

階梯算法的優點是保證成功，牛頓算法的長處是進入快速收斂區後收斂極快；將兩種算法的長處結合起來：先使用階梯算法把迅速收斂區域的位置確定，再改用牛頓算法迅速向方程式的根靠近。

壹、研究動機

高中數學課程中，我們使用勘根定理、虛根成對等方法來找出實係數多項方程式的解。但當係數中包含虛數時，上述幾種方法就不適用了，所以我們著手尋找解複係數多項方程式的方法。我們在<數學傳播>月刊中看見了與所學完全不同的方法，這個方法的圖示如栽種植物般，將大籬笆放在培養皿上，然後栽種尋根的種子……，而 n 個新芽長出 n 條藤，並四處攀援，最後向上尋根。我們對此方法十分有興趣，於是將它做為我們專題研究的主題。

貳、研究目的

一、以庫恩植物栽培法為依據，並限制尋根之規則，期能較迅速逼近複係數方程式的零點。

二、證明尋根方法的理論依據。

三、比較牛頓法與階梯法及其他方法的優缺點，並尋找更實在好用的方法。

參、研究設備器材

紙、筆、電腦、計算機、Geogebra。

肆、研究過程與方法

一、多項式方程式求根的階梯算法

(我們所使用的階梯法是由庫恩的論文改良而來，而庫恩法會在之後有詳細介紹)

1. 階梯算法的步驟：

(1)首先我們把一系列複數平面C_k,k=-1,0,1,2⋯層層向上排好, 在上面劃線，層層均分割成三角形。除C_{-1} 分割方式略有不同之外，其餘各層的分割規律完全相同。

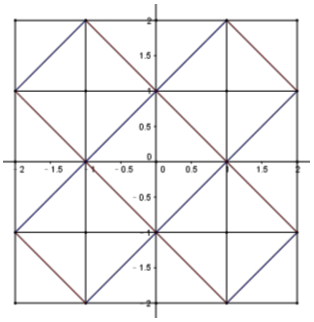
劃分規則如下：

對k=-1：(如圖一)

$\{z \in C : x = p, p \in Z\}$

$\{z \in C : y = p, p \in Z\}$

和$\{z \in C : y = (2p+1) \pm x, p \in Z\}$分割$C_{-1}$平面，



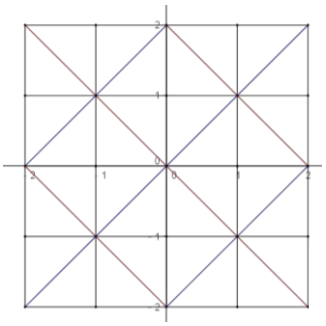
(圖一)

而對 k=0,1,2,3,⋯，用四組平行直線：(如圖二及圖三)

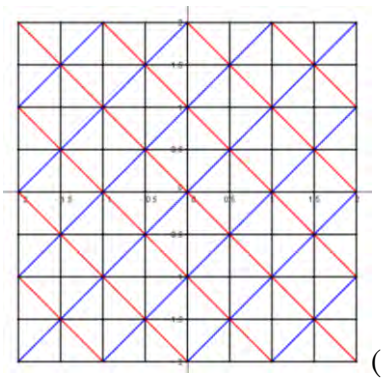
$\{z \in C : x = 2^{-k} p, p \in Z\}$

$\{z \in C : y = 2^{-k} p, p \in Z\}$

和$\{z \in C : y = 2^{1-k} p \pm x, p \in Z\}$分割$C_k$平面。

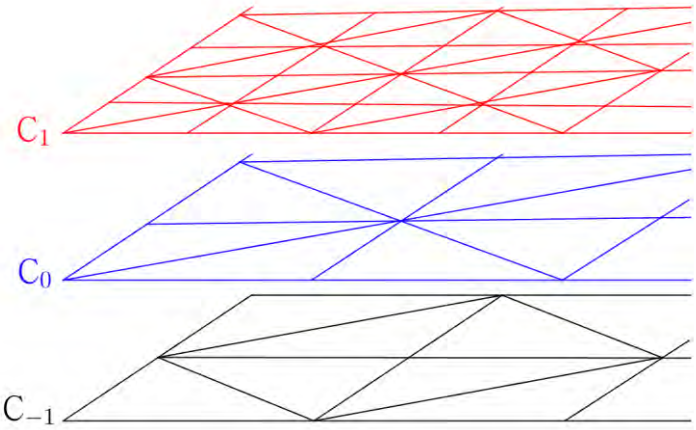


(圖二)



(圖三)

並且從C_0開始，每向上一層， 則線的密度就增加一倍，三角形的邊就縮小二分之一（如圖四）。



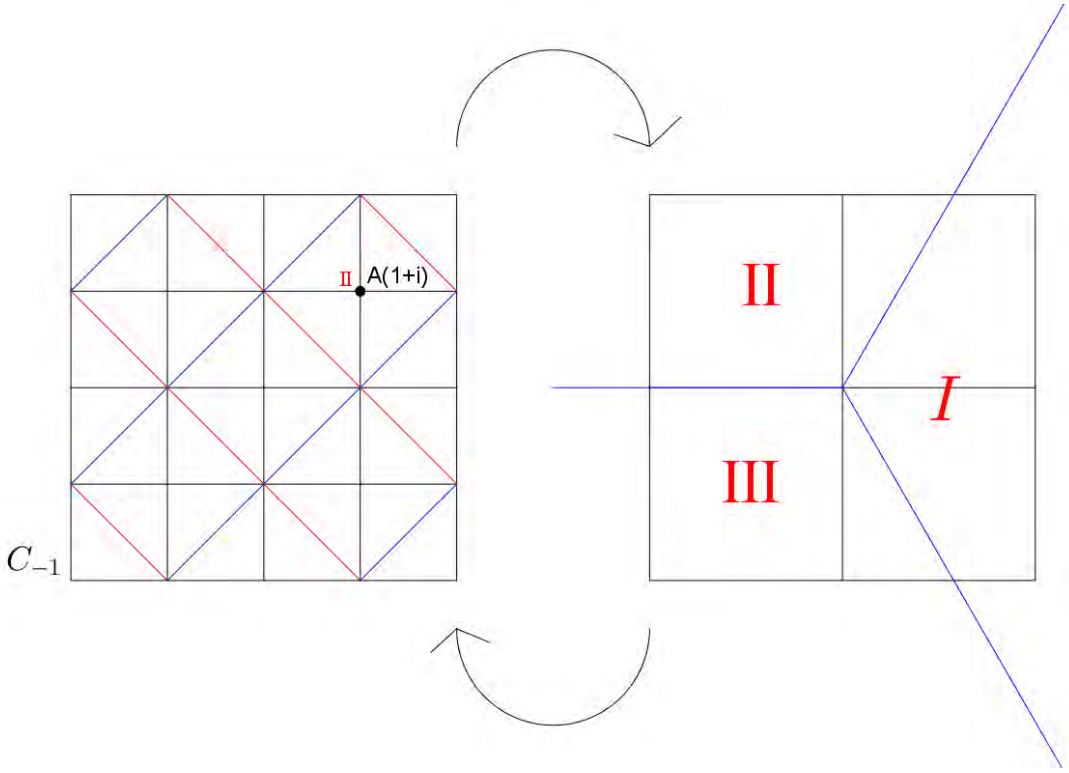
(圖四)

(2)接著我們在C_{-1}上取一個邊長2m格的方塊Q_m（依庫恩算法，取$m=\left\lceil \frac{3(1+\sqrt{2})n}{4\pi} \right\rceil$，n為方程式的次數），再另取一個複數平面c'，用三條射線把複數平面c'分成相等的三個扇形部分I、II、 III區 (如圖五)。對於C_k的每一 層面上的每個點z算出f(z)，

$$g(z)=\begin{cases} \text{I, 若} f(z)=0 \text{或} -\frac{\pi}{3} \le \operatorname{Arg} f(z) \le \frac{\pi}{3} \\ \text{II, 若} \frac{\pi}{3} < \operatorname{Arg} f(z) \le \pi \\ \text{III, 若} -\pi < \operatorname{Arg} f(z) < -\frac{\pi}{3} \end{cases},$$

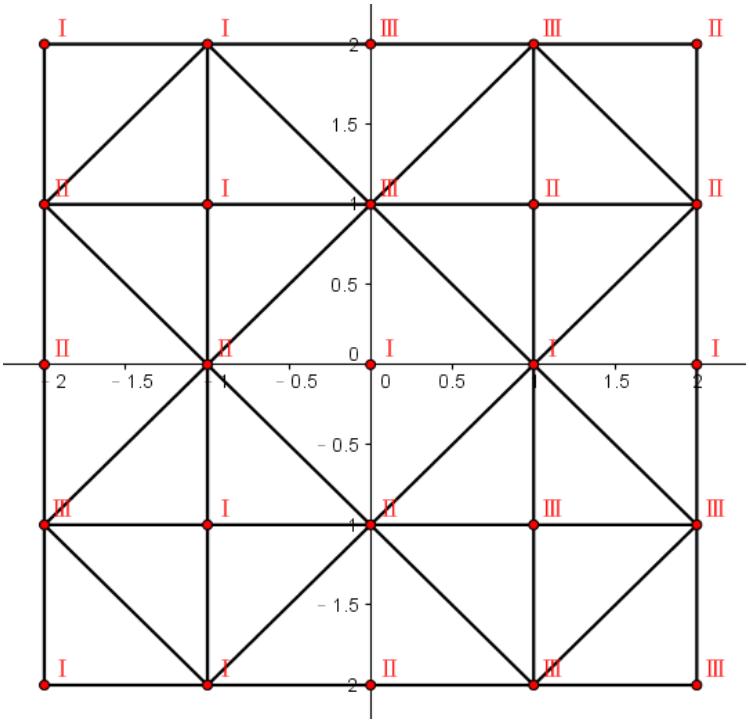
若f(z)落在I區，z就標號I；若f(z)落在II區，z就標號II；若f(z)落在III區， z 就標號III。如此就可以把C_k上Q_m內的全部頂點依此方法完成標號。(如圖六)

將 A(1+i)代入$f(z) = z^3 \Rightarrow f(1+i) = -2+2i$得到A'，再對應A'將落在第 II 區



(圖五)

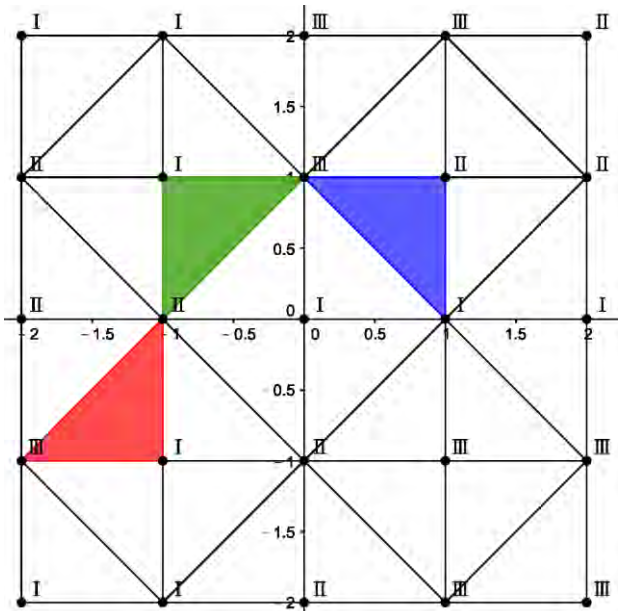
再把 II 標回A(1+i)



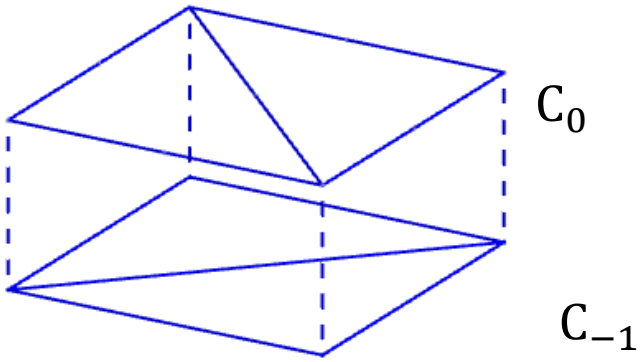
(圖六)

而為什麼要將點分別標號成三區呢？當$f(z) = 0$時，z就是欲求解之方程式的解。我們知道3個點可以圍成一個面，如果我們將分成三區，每區各找一個點，就能形成一個三角形，且這個三角形必可圍住原點，也就是圍住我們所要找的解。

(3)由(圖六)我們可以知道，在Q_m的邊界上有三個I 、II相鄰的稜，我們從這些稜出發，在C_{-1}上尋找下一個I 、II相鄰的稜，直到找到的這個稜是完全標號三角形的其中一稜為止。



(圖七)



(圖八)

在C_{-1}上找到完全標號三角形後， 再把C_{-1}與C_0之間的平面分成許多四面體，依庫恩的方法，在四面體的面上找出有完全標號三角形的那個面，持續此步驟直至到達C_0為止。(如圖七、圖八)

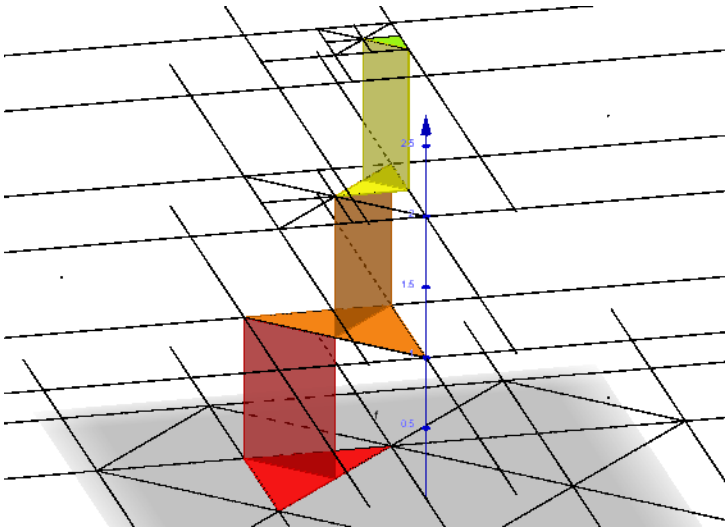
(4) $C_k, k \geq 0$的階梯法則：

我們將庫恩法改良為複數平面層$C_k, k \geq 0$以後，遇到逆時針完全標號三角形，便建構一個三角形階梯， 再往正上方穿過去, 進入另一個較小的逆時針完全標號三角形，繼續以此三角形建構一個三角形階梯，如此持續進行。

為什麼要尋找「逆時針」的完全標號三角形呢？我們發現只有踏在逆時針三角形時才能

穩定的往上，不會往下掉，如此便能省掉計算上的麻煩。

三角形階梯越向上範圍越小，即愈往上愈能逼近根；因此當我們取極限時，這n個階梯就能站在多項式的全部n 個根上了（如圖九）。有關三角形的逼近，我們會在之後詳細說明。



（圖九）

由於階梯法為一逼近根的方法，所以在大多數的情況下皆能找到逆時針完全標號三角形，並往上尋根。但我們會不會有極小的機率，在向上尋根的過程中就直接找到解了呢？

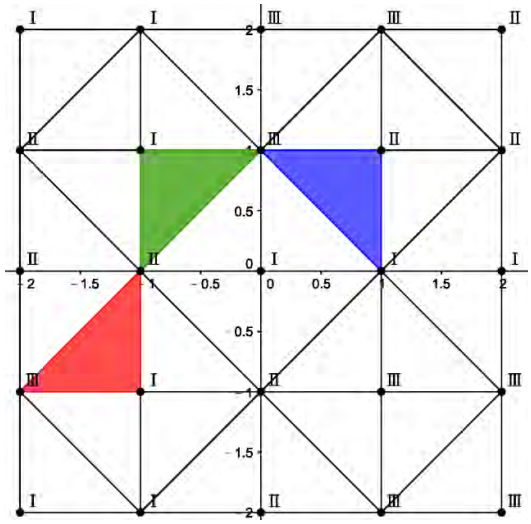
若遇到此種狀況並使用階梯法，在計算時我們只要照著原來的方法，當算到$f(z) = 0$就是此方程式的其中一個解了。此時平面上只會剩下兩個三角形，且其他兩個三角形不受干擾，仍會繼須往上逼近，直到找到方程式的根為止。

2.實例操作

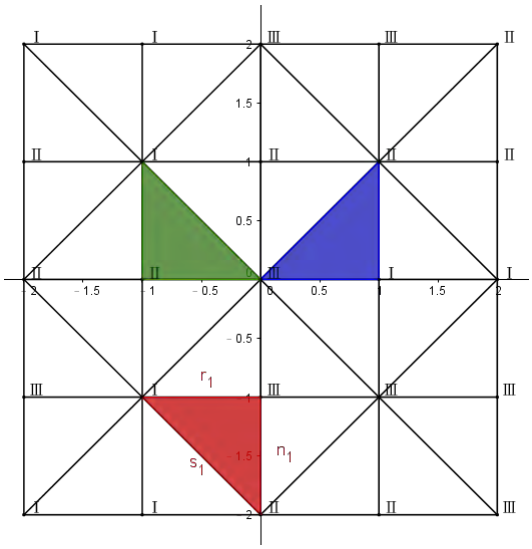
我們以$f(z) = z^3 + \left(\frac{2-i}{3}\right)z^2 + \left(\frac{i}{9}\right)z + \frac{2-11i}{27}$為例，將冗長的計算過程結果製圖如下：

依庫恩算法，對於此三次方程式 n =3，因此我們取$m = \left\lceil \frac{3(1+\sqrt{2})\times 3}{4\pi} \right\rceil$

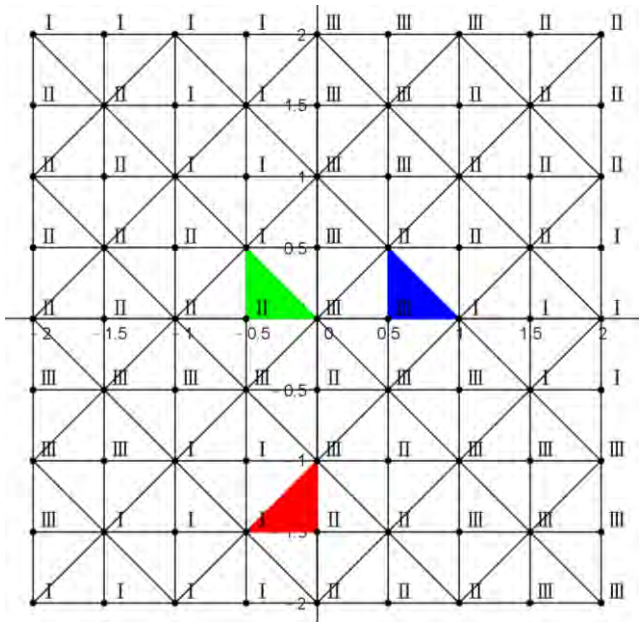
$\fallingdotseq |1.729| = 2$



C_{-1}

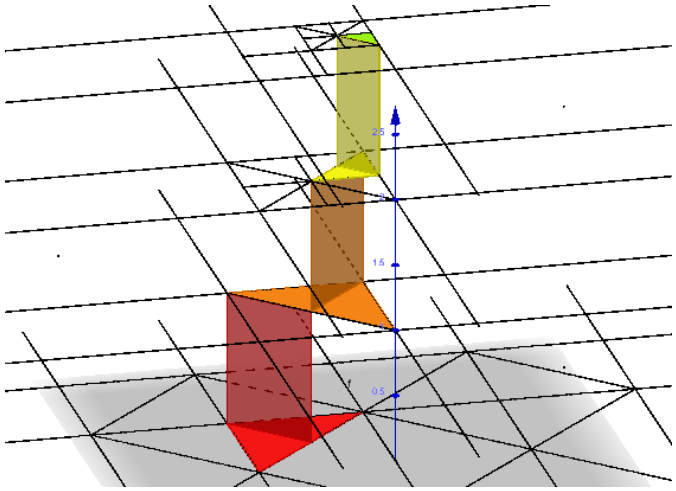


C_0



C_1

(3)步上尋根的階梯，如（圖十）：



（圖十）

3.階梯法的逼近與收斂

在此仍以$f(z) = z^3 + \left(\frac{2-i}{3}\right)z^2 + \left(\frac{i}{9}\right)z + \frac{2-11i}{27}$為例。

首先，我們先定義符號：

$z_{(k,j,i)} = C_k$ 複數平面上第 j 個逆時針完全標號三角形(我們由複數平面右方開始將三角形標號為 j_1 、 j_2 、 j_3 ，若有兩個逆時針完全標號三角形在同一行，則先將上方的三角形標號)標號為 i 的 z 值；

$f_k(j,i) = f(z_{(k,j,i)})$，即 C_k 複數平面上第 j 個逆時針完全標號三角形標號為 i 的 z 值代入函數所得 f(z)之值。

j=1

K	$z_{(k,j,1)}$	$z_{(k,j,2)}$	$z_{(k,j,3)}$	$f_k(j,1)$	$f_k(j,2)$	$f_k(j,3)$
0	1	1+i	0	2.074074074 −1.185185185i	−0.481481481 +2.814814814i	0.074074074 −0.407407407i
1	0.5 + 0.5i	0.5 + i	i	0.379629629 −0.120370370i	0.856481481 − 0.370370370i	−0.148148148 −0.740740740i
2	0.5 + 0.5i	0.5 + 0.75i	0.25 + 0.75i	0.379629629 −0.120370370i	−0.686342592 0.392939814i	0.623842592 −0.244212963i
3	0.375 + 0.625i	0.375 + 0.75i	0.25 + 0.75i	0.079716435 +0.049623842i	−0.141420717 +0.086082175i	0.623842592 −0.244212963i
4	0.375 + 0.625i	0.375 + 0.6875i	0.3125 + 0.6875i	0.079716435 +0.049623842i	-0.023853443 +0.074607566i	−0.035572193 −0.029206452i
5	0.34375 + 0.65625i	0.34375 + 0.6875i	0.3125 + 0.6875i	0.018877947 +0.013524938i	−0.014620180 +0.018881338i	−0.035572193 −0.029206452i
6	0.34375 + 0.65625i	0.328125 + 0.671875i	0.328125 + 0.65625i	0.018877947 +0.013524938i	−0.009168554 +0.007034584i	0.001112478 −0.006854304i
7	0.3359375 + 0.6640625i	0.328125 + 0.6640625i	0.328125 + 0.65625i	0.004925586 −0.0039921i	−0.008410859 +0.005287629i	0.001112478 −0.006854304i

j=2:

k	$z_{(k,j,1)}$	$z_{(k,j,2)}$	$z_{(k,j,3)}$	$f_k(j,1)$	$f_k(j,2)$	$f_k(j,3)$
0	−1 − i	−i	0	1.962962962 −0.962962962i	−1.037037037 +0.592592592i	0.074074074 − 0.407407407i

1	$-0.5-0.5i$	$-i$	$-0.5i$	0.842592592 $-0.601851851i$	-1.037037037 $+0.592592592i$	0.314814814 $-0.365740740i$
2	$-0.5-0.5i$	$-0.25-0.75i$	$-0.25-0.5i$	0.842592592 $-0.601851851i$	-0.000578703 $+0.151620370i$	-0.060763888 $-0.243055555i$
3	-0.375 $-0.625i$	$-0.25-0.75i$	$-0.25-0.625i$	0.057581018 $+0.055700231i$	-0.000578703 $+0.151620370i$	-0.124276620 $-0.059968171i$
4	-0.375 $-0.625i$	-0.3125 $-0.6875i$	-0.3125 $-0.625i$	0.057581018 $+0.055700231i$	-0.029821325 $+0.037308304i$	0.037281177 $-0.057743778i$
5	-0.34375 $-0.65625i$	-0.3125 $-0.6875i$	-0.3125 $-0.65625i$	0.013384783 $-0.018443919i$	-0.029821325 $+0.037308304i$	0.033815737 $-0.012465865i$
6	-0.34375 $-0.65625i$	-0.328125 $-0.671875i$	-0.318125 $-0.65625i$	0.013384783 $-0.018443919i$	-0.008908024 $+0.014702408i$	0.003498925 $-0.010227627i$
7	-0.3359375 $-0.6640625i$	-0.328125 $-0.671875i$	-0.328125 $-0.6640625i$	0.003440609 $-0.004617390i$	-0.008908024 $+0.014702408i$	-0.008624006 $-0.002949838i$

j=3:

k	$z_{(k,j,1)}$	$z_{(k,j,2)}$	$z_{(k,j,3)}$	$f_k(j,1)$	$f_k(j,2)$	$f_k(j,3)$
0	$-1+i$	$-2+i$	-2	1.518518518 $-1.037037037i$	-1.481481481 $-8.148148148i$	-5.925925925 $-0.851851851i$
1	$-0.5+0.5i$	$-1+0.5i$	-1	0.452148437 $+0.267578125i$	0.049804687 $+1.740234375i$	-1.094726562 $+0.353515625i$
2	$-0.5+0.5i$	$-0.75+0.5i$	$-0.75+0.25i$	0.452148437 $+0.267578125i$	-0.076967592 $+2.223379629i$	-0.307870370 $-0.501157407i$
3	-0.625 $+0.375i$	$-0.75+0.5i$	$-0.75+0.375i$	0.062355324 $-0.013744212i$	-0.076967592 $+2.223379629i$	0.020688657 $-0.412398726i$
4	0.34375 $+0.21875i$	$-0.75+0.5i$	-0.75 $+0.21875i$	0.374156810 $-0.343367965i$	-0.076967592 $+2.223379629i$	0.411734121 $-0.340889259i$
5	-0.359375 $+0.359375i$	-0.75 $+0.359375i$	-0.75 $+0.21875i$	0.120731212 $-0.207267196i$	-0.038282323 $-0.101924613i$	0.411734121 $-0.340889259i$
6	-0.359375 $+0.359375i$	-0.5546875 $+0.359375i$	-0.5546875 $+0.6890625i$	0.120731212 $-0.207267196i$	-0.0364663860 $+0.846456174$	0.002854417 $-0.110356825i$
7	-0.52734375 $+0.359375\ i$	-0.5546875 $+0.359375i$	-0.52734375 $+0.39453125i$	0.088636475 $-0.105701526i$	-0.0364663860 $+0.846456174$	-0.056395376 -0.024771800

由上述表格可以發現，當$C_k, k \geq 0$，k 值愈大時，其值 f(z)會愈趨近於零，而當我們取極限時，則可得到此方程式的根。

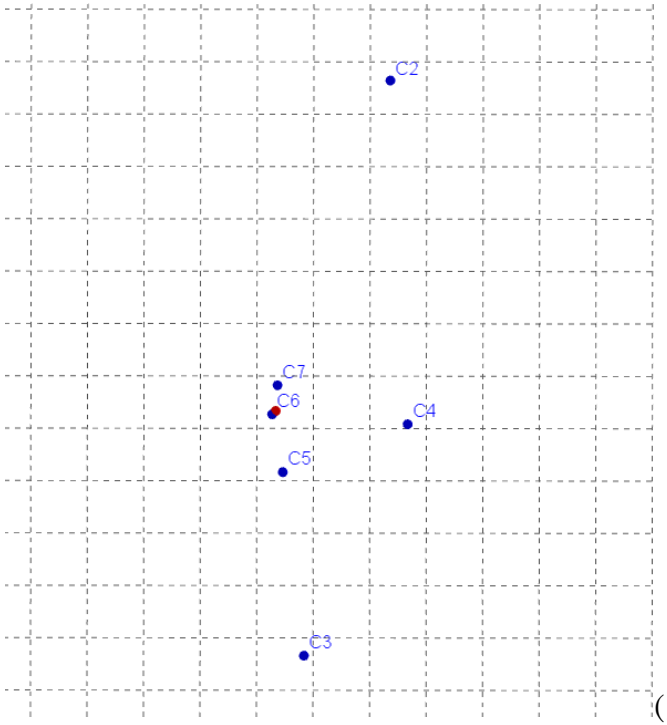
從另一方面來看，我們也能由預設根和實際根的位置或數值來觀察根的逼近。我們將預設根的位置定在完全標號三角形的外接圓圓心處(因其和三頂點等距)，並比對和計算在每一層平面中，預設根和實際根的相對位置及數值上的差異。

以$f(z) = z^3 + \left(\frac{2-i}{3}\right)z^2 + \left(\frac{i}{9}\right)z + \frac{2-11i}{27}$的其中一根$(\frac{1}{3}, \frac{2}{3})$為例，可得出如下表的結果：

R_k	第 k 層近似根誤差範圍圓半徑	預設根位置
R_2	$\frac{2-\sqrt{2}}{4}$ (約為 0.14644660)	$(\frac{\sqrt{2}}{4}, \frac{1+\sqrt{2}}{4})$
R_3	$\frac{2-\sqrt{2}}{16}$ (約為 0.0366116)	$(\frac{4+\sqrt{2}}{16}, \frac{10+\sqrt{2}}{16})$
R_4	$\frac{2-\sqrt{2}}{32}$ (約為 0.01830582)	$(\frac{10+\sqrt{2}}{32}, \frac{20+\sqrt{2}}{32})$
R_5	$\frac{2-\sqrt{2}}{64}$ (約為 0.00915291)	$(\frac{20+\sqrt{2}}{64}, \frac{42+\sqrt{2}}{64})$
R_6	$\frac{2-\sqrt{2}}{128}$ (約為 0.00457645)	$(\frac{44-\sqrt{2}}{128}, \frac{84+\sqrt{2}}{128})$
R_7	$\frac{2-\sqrt{2}}{256}$ (約為 0.00228822)	$(\frac{84+\sqrt{2}}{256}, \frac{168+\sqrt{2}}{256})$

註：R_k 表實際根與預設根之誤差範圍圓半徑

我們將上述的點繪製於下圖中，可以發現當 k 值愈大時，其預設根的位置或數值有愈逼近實際根(紅點處)的現象(如圖十二)。C_4和C_2、C_3的預設根相比，雖然數值相較之下偏離實際根，但位置上卻大大逼近了與實際根的距離，這和使用牛頓法逼近複數方程式的狀況非常雷同(稍後在牛頓法的部分會有詳細說明)；C_7和C_5、C_6的預設根相比，雖然位置稍稍偏離了實際根，但在數值上仍較接近實際根。

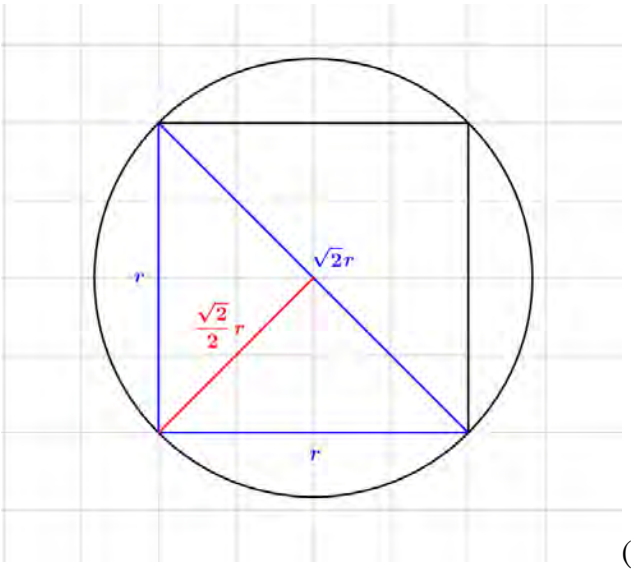


(圖十一)

由上述兩種不同面向的討論，我們可以得知：階梯法是可以逼近根的，並能準確的找到根的位置。

4.根的逼近與誤差值計算

當 $C_k, k \geq 0$，我們取半徑為 $\frac{\sqrt{2}}{2} \cdot 2^{-k}$ 的圓，也就是單位方格的外接圓，並假設欲求方程式的根皆在此圓中(如圖十二) (後稱此種圓為單位圓)。由此我們可知在 C_k 上的單位圓面積為 $2^{-2k-1}\pi$，並推導出對於 C_k 和 C_{k+1} 單位圓的面積差為 $3\pi \cdot 2^{-2k-3}$ 個單位長。



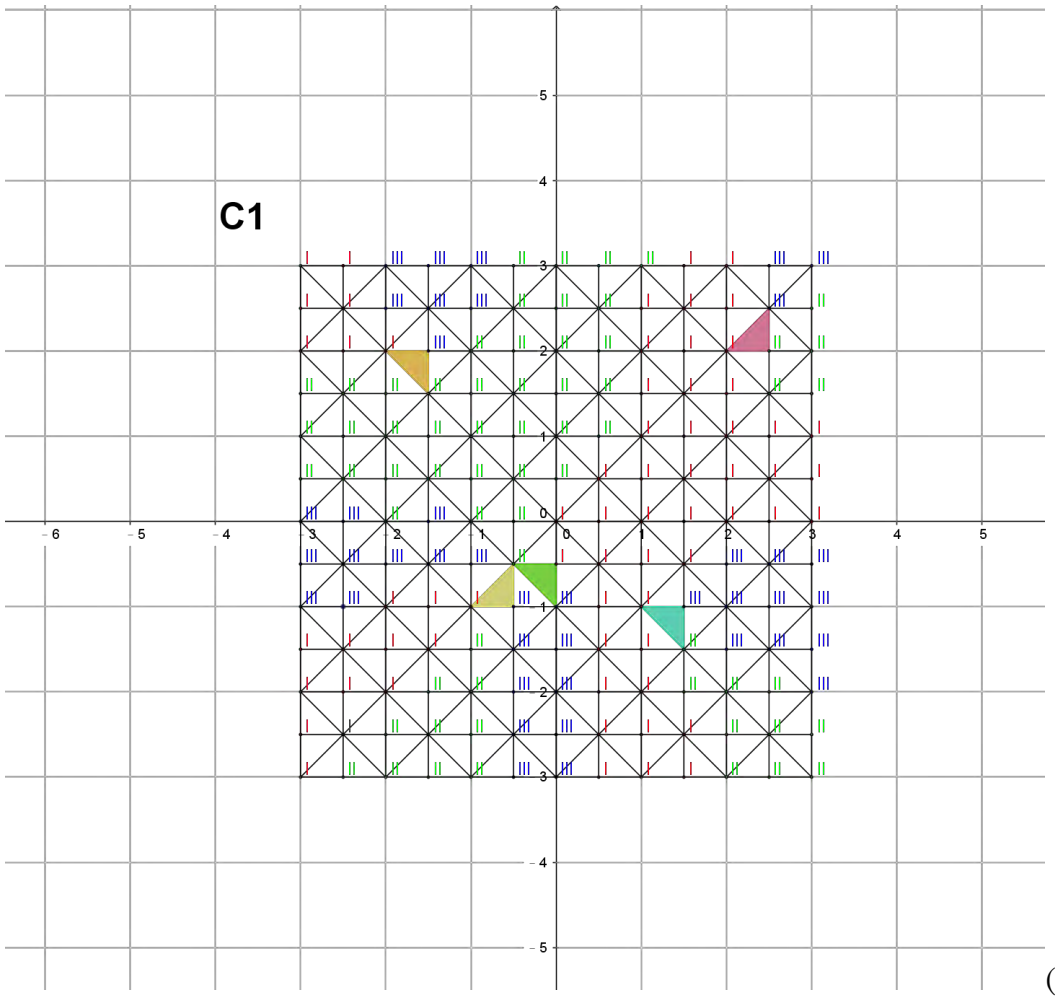
(圖十二)

接著我們要定義所謂的逼近速率。我們知道當三角形階梯越向上範圍越小，即愈往上愈能逼近根，它所在的方格及單位圓也有同樣的性質。若以圓上每一個點所組成的圓，其圓面積代表在單位圓上所有根的可能性，則逼近速率即為C_k和C_{k+1}的圓面積差。

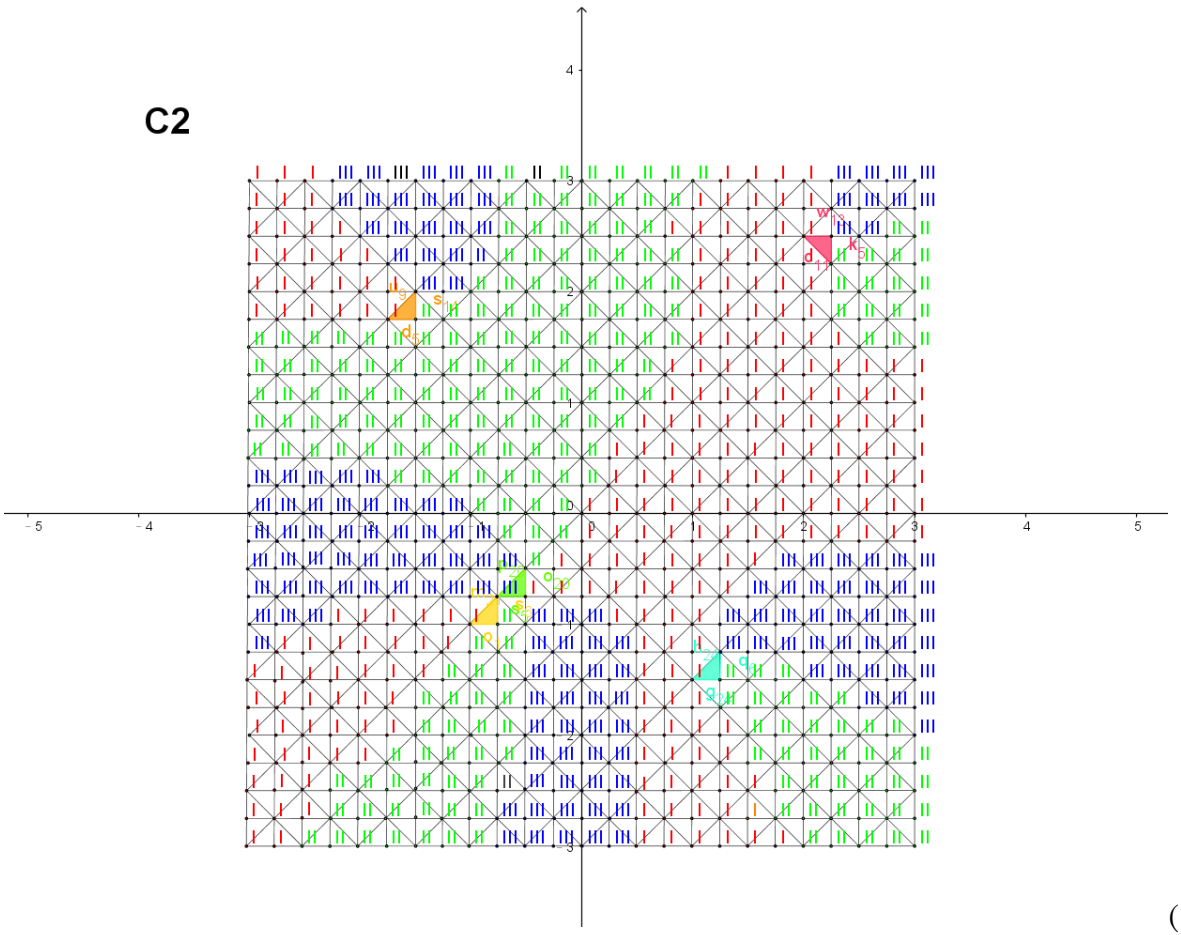
又C_k和C_{k+1}的圓面積差以及C_{k+1}和C_{k+2}圓面積差的比值：$\dfrac{3\pi\cdot 2^{-(2+1)-3}}{3\pi\cdot 2^{-2k-3}}=\dfrac{1}{4}$，則逼近速率每往上一層就減少$\dfrac{1}{4}$。

5.方程式有重根的討論

首先，我們以已知解的五次方方程式$f(z)=\left[x-(-\frac{3}{5}-\frac{4}{5}i)\right]\left[x-(\frac{3}{5}-\frac{4}{5}i)\right]\left[x-(\frac{6}{5}-\frac{7}{5}i)\right]\left[x-(-\frac{8}{5}+\frac{9}{5}i)\right]\left[x-(\frac{11}{5}+\frac{2}{5}i)\right]$進行實例操作：(如圖十三、圖十四)



(圖十三)



(圖十四)

如上述的舉例我們可以知道: 根據庫恩法，無論是否有重根，在 C_{-1}層時皆會有 n 個起始點。而向 C_0以上的平面發展時，重根的兩個藤會分別進入極接近的兩個三角形。當我們取極限時，此兩個根會在較上方的平面重合，進而我們可以知道其為兩重根。

6.階梯法之引理證明

從 ∂Q 的(1,2)稜開始計算,在方塊 Q_n 內平面搜尋到完全標號三角形後，就向空中發展，在層層三角形階梯中踏行，每踏上一個三角形階梯，就要計算等分點的 f(z)，尋找縮小四分之一的完全標號三角形階梯。計算的進行，可以看作一個一個向上縮小的階梯，當其踏過數階時，便能非常逼近方程式的根，我們利用下述引理，可知道其根必在完全標號三角形階梯內。

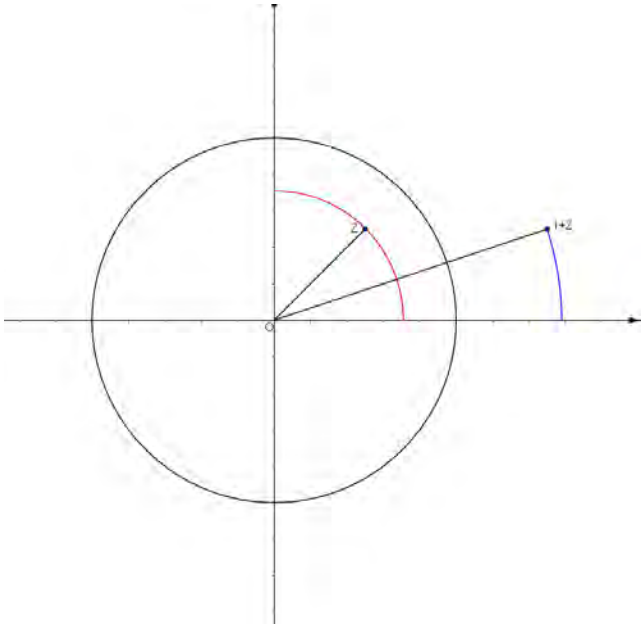
首先，必須利用下述命題證明二引理：

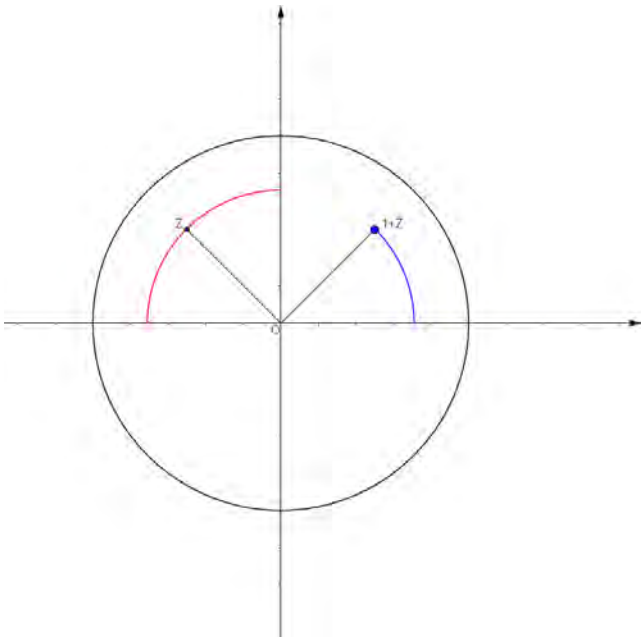
(命題)

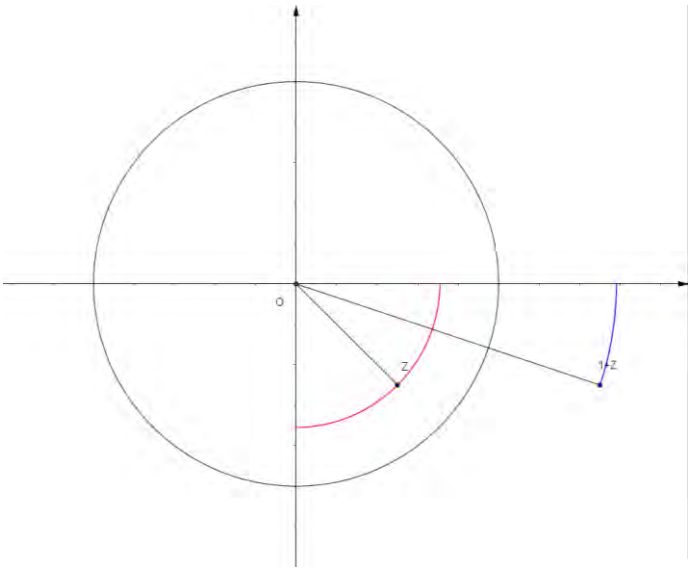
若複數 z 符合 $|z|<1$,且幅角取值範圍為 $(-\pi,\pi]$ ，則 $\mathrm{Arg}(1+z)\le\frac{\pi}{2}|z|$ 。

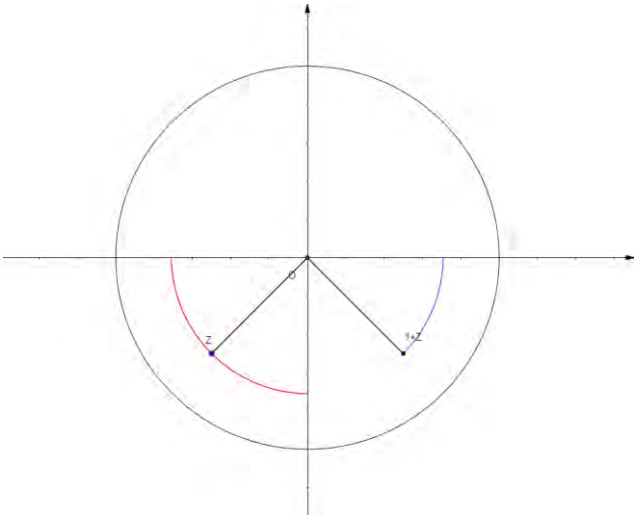
證明：以下列圖示即可證明：

Z 在半徑為 1 的圓內，有下列四種情形：









由上列四個圖示觀之，明顯可知此命題為真。

〔引理 1〕

令 $f(x)=x^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$

$a=\max\{|a_0|,|a_1|,\cdots,|a_{n-1}|\}$

$$r=\max\left\{\frac{3\sqrt{2}(2+\pi)n}{4\pi},1+\frac{na}{n-1}\right\}$$

$$R=r+\sqrt{2}$$

則完全標號三角形和原點軸{0}×[-1 , +∞ ）的距離都小於 R。

證明：

因為三角形稜長不超過 $\sqrt{2}$ ，所以有一頂點與原點距離超過 R 的三角形，三頂點與原點的距離都會超過 r，我們只須證明這樣的三角形不是完全標號三角形。

設頂點(z,k),k 為不小於-1 的整數

$(z_1,k_i),(z_2,k_i)$ 為上述三角形的兩個頂點，且 $k_i>-1$,兩點均由 w=f(z)計算標號。

$$\Rightarrow f(z)=z^n(1+\frac{a_{n-1}}{z}+\frac{a_{n-2}}{z^2}+\cdots+\frac{a_0}{z^n})$$

$$=z^n(1+g(z)) \quad 〔g(z)=(\frac{a_{n-1}}{z}+\frac{a_{n-2}}{z^1}+\cdots+\frac{a_0}{z^2})〕$$

因為 $|z_1|>r,|z_2|>r$

$$\Rightarrow |g(z)|\le\frac{|a_{n-1}|}{r}+\frac{|a_{n-2}|}{r^2}+\cdots+\frac{|a_0|}{r^n}$$

又 $a=\max\{|a_0|,|a_1|,\cdots,|a_{n-1}|\}$

$$\Rightarrow |g(z)|\le a(\frac{1}{r}+\frac{1}{r^2}+\cdots+\frac{1}{r^n})$$

$$\Rightarrow |g(z)|<a\cdot\frac{\frac{1}{r}(1-\frac{1}{r^n})}{1-\frac{1}{r}}=\frac{a}{r-1}$$

又 $r=\max\left\{\frac{3\sqrt{2}(2+\pi)n}{4\pi},1+\frac{na}{n-1}\right\}$

$$\Rightarrow |g(z)|<\frac{a}{(1+\frac{na}{n-1})-1}=\frac{n-1}{n}$$

$$|g(z_1)-g(z_2)|\le|a_{n-1}|\left|\frac{1}{z_1}-\frac{1}{z_2}\right|+\cdots+|a_0|\left|\frac{1}{z_1^n}-\frac{1}{z_2^n}\right| \quad \text{(根據三角不等式)}$$

其中 $\left|\frac{1}{z_1^k}-\frac{1}{z_2^k}\right|=\left|\frac{z_2^k-z_1^k}{z_1^k\cdot z_2^k}\right|$

$$=\left|\frac{(z_2-z_1)(z_2^{k-1}+z_2^{k-2}z_1+\cdots+z_1^{k-1})}{z_1^k z_2^k}\right|$$

又 $\left|\frac{z_2^{k-1}+z_2^{k-2}z_1+\cdots+z_1^{k-1}}{z_1^k z_2^k}\right|<\frac{k\cdot r^{k-1}}{r^k\cdot r^k}=\frac{k}{r^{k+1}}$

$$\Rightarrow |g(z_1)-g(z_2)|\le a|z_1-z_2|(\frac{1}{r^2}+\frac{2}{r^3}+\ldots+\frac{n}{r^{n+1}})$$

$$\le\frac{\sqrt{2}a}{(r-1)^2}$$

$$\le\frac{\sqrt{2}(n-1)}{n(r-1)}$$

又 $r=\max\left\{\frac{3\sqrt{2}(2+\pi)n}{4\pi},1+\frac{na}{n-1}\right\}$

$$\Rightarrow |g(z_1)-g(z_2)|<\frac{\sqrt{2}(n-1)\cdot 4\pi}{3n\sqrt{2}(2+\pi)(n-1)}=\frac{4\pi}{3n(2+\pi)}$$

$$\Rightarrow\left|\frac{g(z_1)-g(z_2)}{1+g(z_2)}\right|\le\frac{\frac{4\pi}{3n(2+\pi)}}{1-\frac{n-1}{n}}=\frac{4\pi}{3(2+\pi)}<1$$

若 $w_1=f(z_1),w_2=f(z_2)$,則由(命題 1)可得

$$\left|\mathrm{Arg}\frac{w_1}{w_2}\right|=\left|\mathrm{Arg}\frac{f(z_1)}{f(z_2)}\right|$$

$$=\left|\mathrm{Arg}\frac{z_1^n(1+g(z_1))}{z_2^n(1+g(z_2))}\right|$$

$$\le\left|\mathrm{Arg}\frac{z_1^n}{z_2^n}\right|+\left|\mathrm{Arg}\frac{1+g(z_1)}{1+g(z_2)}\right|$$

$$=\left|\mathrm{Arg}\frac{z_1^n}{z_2^n}\right|+\left|\mathrm{Arg}(1+\frac{g(z_1)-g(z_2)}{1+g(z_2)}\right|$$

$$\le n\cdot\left|\mathrm{Arg}\frac{z_1}{z_2}\right|+\frac{\pi}{2}\cdot\left|\frac{g(z_1)-g(z_2)}{1+g(z_2)}\right|$$

$$\le\frac{\sqrt{2}n}{\frac{3\sqrt{2}(2+\pi)n}{4\pi}}+\frac{\pi}{2}\cdot\frac{4\pi}{3(2+\pi)}=\frac{2\pi}{3}$$

因為三頂點在 w 平面的像，兩兩之間對原點的張角小於 $\frac{2\pi}{3}$，

所以此三角形不是完全標號三角形。

〔引理 2〕

$k \geq 0$ 時，C_k 平面上完全標號三角形任一頂點與多項式某個零點的距離不超過

$$\frac{\sqrt{2}(1+\frac{3n}{4})}{2^k}$$

證明：

因為 C_k 平面上三角形任兩頂點的距離不超過 $\delta = \frac{\sqrt{2}}{2^k}$，即不超過等腰直角形斜邊長。

所以只須證明完全標號三角形必有一頂點與多項式某零點的距離不超過 $\frac{3n\delta}{4}$。

若 $f(z) = (z-\xi_1)(z-\xi_2)\cdots(z-\xi_n)$

設 C_k 平面上三角形 $\{z_1, z_2, z_3\}$ 三頂點與多項式所有零點的距離都大於 $\frac{3n\delta}{4}$，

那麼對於 $j = 1,2,\cdots,n$，有

$$\left|\frac{z_2 - z_1}{z_1 - \xi_j}\right| < \frac{\delta}{\frac{3n\delta}{4}} = \frac{4}{3n} < 1$$

若 $w_i = f(z_i), i = 1,2,3$

$$\Rightarrow \left|\operatorname{Arg}\frac{w_2}{w_1}\right| = \left|\operatorname{Arg}\frac{f(z_2)}{f(z_1)}\right|$$

$$= \left|\operatorname{Arg}\frac{(z_2-\xi_1)(z_2-\xi_2)\cdots(z_2-\xi_n)}{(z_1-\xi_1)(z_1-\xi_2)\cdots(z_1-\xi_n)}\right|$$

$$\leq \left|\operatorname{Arg}\frac{z_2-\xi_1}{z_1-\xi_1}\right| + \cdots + \left|\operatorname{Arg}\frac{z_2-\xi_n}{z_1-\xi_n}\right|$$

$$= \left|\operatorname{Arg}(1+\frac{z_2-z_1}{z_1-\xi_1})\right| + \cdots + \left|\operatorname{Arg}(1+\frac{z_2-z_1}{z_1-\xi_n})\right|$$

$$< \frac{\pi}{2}\cdot\left|\frac{z_2-z_1}{z_1-\xi_1}\right| + \cdots + \frac{\pi}{2}\cdot\left|\frac{z_2-z_1}{z_1-\xi_n}\right|$$

$$< n \cdot \frac{\pi}{2} \cdot \frac{4}{3n} = \frac{2\pi}{3}$$

同理可得

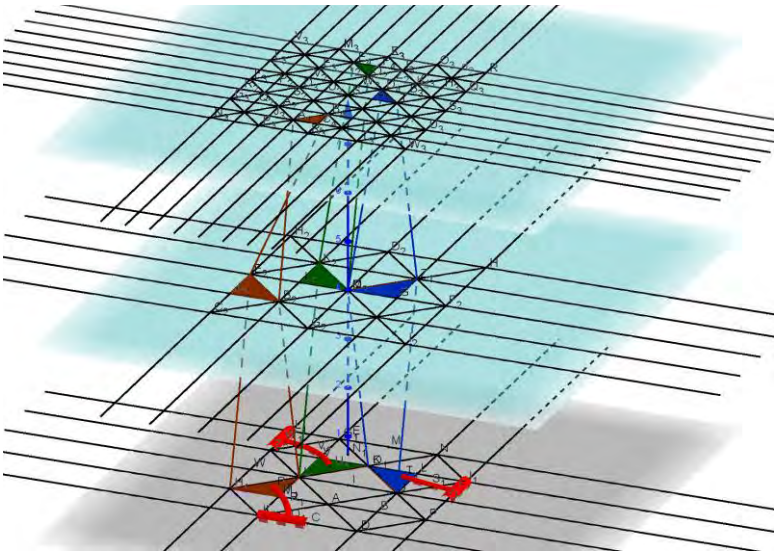
$$\left|\mathrm{Arg}\frac{f(z_3)}{f(z_2)}\right| < \frac{2\pi}{3} \text{，} \left|\mathrm{Arg}\frac{f(z_1)}{f(z_3)}\right| < \frac{2\pi}{3}$$

可知 $f(z_1), f(z_2), f(z_3)$ 在w平面上兩兩對原點的張角都小於 $\frac{2\pi}{3}$，

所以 $\{z_1, z_2, z_3\}$ 不可能是完全標號三角形。

依上述之證明，我們可以知道：C_k 平面上完全標號三角形任一頂點與多項式某個零點的距離不超過 $\frac{\sqrt{2}(1+\frac{3n}{4})}{2^k}$ 。

因為踏過的階梯均是經過計算出來的完全標號三角形，由上述二引理可知，可以步上階梯，走上尋根的路，如（圖十五）。



（圖十五）

二、各種求複係數方程式方法的比較

1.牛頓算法和階梯法

(1)牛頓算法的基本思想：曲線較複雜，計算起來會較麻煩，若切線是直線，計算起來就比較容易了。用切線取替曲線來計算，就是牛頓法的基本想法。

(2)沿曲線走應該可以走到代表函數的根的一點，但曲線是比較複雜的，沿曲線走看似容易，其實做起來不容易。因為切線是最貼近曲線的直線，所以試著沿切線走向函

數的根。於是從已知點出發沿切線走到 x 軸上，這就是牛頓方法（切線法）。

(3)牛頓算法的迭代公式： $z_k = z_{k-1} - \frac{f(z_{k-1})}{f'(z_{k-1})}, k = 1,2,3,\cdots$

(4)分析牛頓算法:

牛頓算法的優點在於敘述比較簡單，一個迭代公式就把算法說清楚了，而多項式的微分又十分容易學，所以牛頓算法是一種一學就會的算法。而數學家已經證明，如果把迭代出發點 x_0 選得離多項式或函數的根 ζ 很近，且這個根是單根，那麼牛頓迭代一定會收斂到 ζ，並且一定收斂得很迅速，在這種前提下，牛頓算法的收斂速度，比階梯算法快很多。

從另一方面來看，牛頓算法一次只能算一個根，雖然從理論而言，這或許稱不上是缺點，因為算出多項方程式 f(z)=0 的一個根 ζ 以後，用 $(x-\zeta)$ 除原來的多項式，得到一個階數降 1 的多項式，再用牛頓算法求這個新的多項式的根，……，這樣一次一次降階，便可以把原多項式的全部根算出來。但階梯算法則是可以同時求所有的根，在這方面沒那麼複雜。

再者，理論情形與實際計算之間難免會有些差距，依理論推導中的 ζ 是多項方程式的準確根，但是機器算出來的，只是準確根的近似值，所以在實際計算時，$(x-\zeta)$ 這個因式只是一個近似因式，多次使用近似因式除多項式之後，會造成誤差的積累，即誤差會越來越大。所以使用牛頓法求高階多項式全部根的計算有時候會變得不可靠。而階梯算法是用 n 群三角形階梯去踏向多項方程式的全部 n 個根，n 群三角形階梯可以同時進行，相互之間不相干擾，更不會造成誤差的積累。

還有，牛頓算法要計算函數值與函數微分的值，而階梯算法完全不必計算微分，這也是二者的不同的地方，所以，兩種算法是各有優劣。

牛頓算法最麻煩的地方，就是有時候計算時很不順利，雖然牛頓算法很容易學，但是計算是否能成功，卻無法保證。容易學但並不保證找得到根，這就令人失去信心，而階梯算法有時算得慢一些，卻能保證找得到所有根。

如果從 ζ 附近開始進行牛頓迭代，的確很快收斂到 ζ，但如果從離 ζ 較遠的地方開始進行牛頓迭代，而計算中若遇上微分值等於 0 就無法計算下去，也有可能因出發點的不當，造成過程的捉摸不定，也就是說無法收斂到一根。

對於牛頓算法，數學家已經證明了只要迭代出發點z_0充分靠近多項式的某個根，計算就會收斂到這個根，而且收斂得很迅速。由此可知：在多項式的每個單根附近，會有一個快速收斂區，如果牛頓迭代的出發點選在這個快速收斂區裏，計算就會很快收斂到這個單根。若是複變量，單根的快速收斂區則是包含這個單根的某片開區域。

(5)斯梅爾對牛頓算法的做法：

使用牛頓算法順利的找到根，最重要的是先找到一個足夠好的迭代出發點z_0，迭代出發點也叫做迭代的初值。什麼叫做足夠好的初值呢？斯梅爾提出了良好初值的概念；如果z_0使得牛頓迭代$z_k = z_{k-1} - \frac{f(z_{k-1})}{f'(z_{k-1})}$

對一切 k=1,2,3,⋯都有意義，迭代序列$z_0, z_1, z_2, \cdots$收斂到 f(z)的一個零點，並且對所有 k=1,2,3,⋯都滿足 $\left|\frac{f(z_{k+1})}{f(z_k)}\right| < \frac{1}{2}$，就稱$z_0$是 f(z)的一個良好初值。從這點開始的牛頓迭代可以順利做下去，直到收斂到函數的一個根，並且每次迭代都使函數值的絕對值衰減一半以上。我們知道，當用牛頓算法為函數求根時，只要找到一個良好初值，必能順利且極迅速的收斂到一個根。

有了明確的概念，斯梅爾便依此概念確立到達良好初值的條件，並設計尋找良好初值的方法：

使多項式 f(z)的微分函數$f'(z)$的值為零的點，稱作多項式的臨界點，臨界點的多項式值，稱作多項式的臨界值。 n 階多項式 f(z)的微分函數$f'(z)$是 n-1 階多項式，有 n-1 個根，由此可見， n 階多項式有 n-1 個臨界點（可以相重）。把 n 階多項式 f(z)的所有 n-1 個臨界值（亦可相重）的絕對值按大小排列起來，取最小的一個，記作$\rho = \rho(f)$，我們把它叫做多項式 f(z)的表徵值。

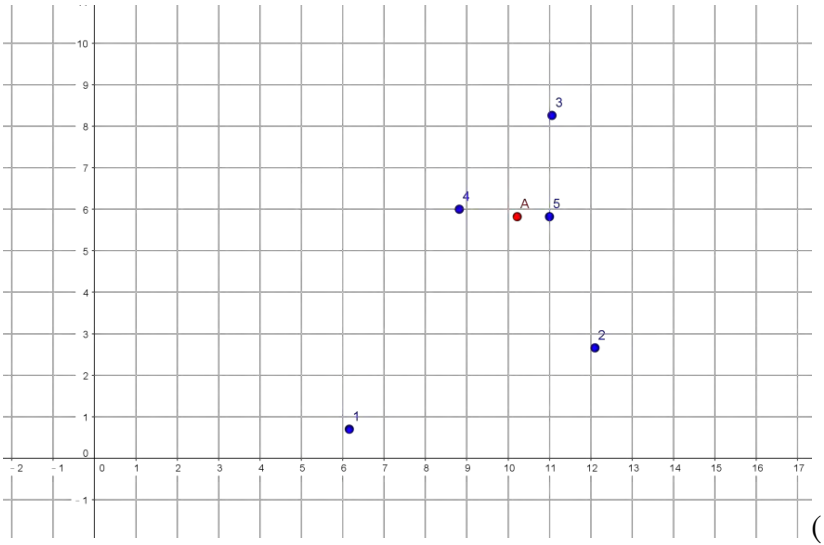
斯梅爾證明了下面的定理：

如果一點z_0的多項式值的絕對值小於多項式的表徵值的$\frac{1}{13}$，即如果$|f(z_0)| < \frac{\rho(f)}{13}$，$z_0$就是牛頓算法求多項方程式 f(z)=0 的根的一個良好初值。

初值若是良好的，可以較快進入快速收斂區，若選得不好，說不定一直不能進入快速收斂區。

(6)使用牛頓法求複數方程式之解：

我們一般都使用牛頓法處理實數係數方程式的計算，但牛頓法也可用於複係數方程式嗎？答案是肯定的，其求解圖形為在高絲平面上散亂的點，但向欲求方程式的根逼近。下圖為示意圖：(紅點為解的實際位置)(1 到 5 為逼近順序)



(圖十六)

(7)牛頓法與階梯法的優缺點比較：

(I) 牛頓法的計算速度較階梯法快許多，且階梯法每往上一層逼近速率會變為原來層數的四分之一倍。

(II) 牛頓算法一次只能算一個根，但階梯算法則可以同時求所有的根。

(III)牛頓法須用到微分，而階梯法完全不用。

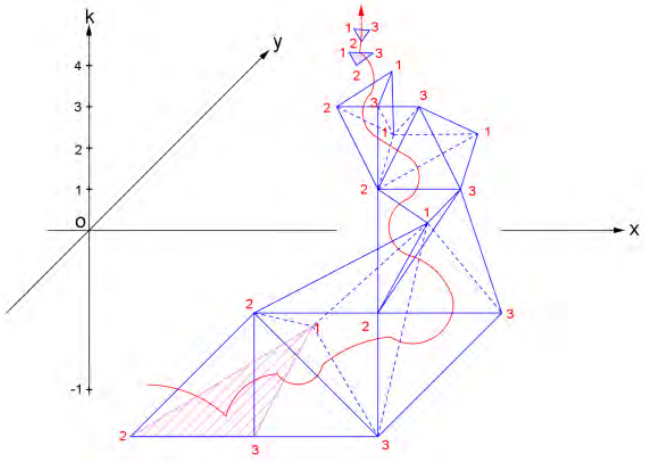
(IV)牛頓法會有誤差，且愈高階時誤差愈大；階梯法完全不會有誤差。

(V)牛頓法有尋找初始值的問題，若選得不好，有可能無法求解；階梯法不會有尋找初始值的困擾，在 C-1 平面時根即可明確的向上爬升。

然而，牛頓在 300 年前率先提出迭代法可謂一劃時代的想法，而更影響了許多後世的數學家，因此牛頓的貢獻仍是無法抹滅的重要。

2.庫恩法和階梯法

(1)庫恩法的步驟和階梯法的(1)~(3)相同，但 $C_k, k \geq 0$ 時，庫恩法須將兩層平面間分割成許多四面體，並在這些四面體中找到完全標號三角形後穿梭(如圖十七)。



(圖十七)

(2)庫恩法需要一直剖分四面體，分析各個四面體中標號三角形在空間中的相對位置，再沿著這些面向上生長，但途中有可能會先向下再向上長，過程較複雜。階梯法就簡單的多，不僅去除了向下的可能性，並只需向上尋找完全標號三角形即可。

3.標號法與階梯法

(1)由於在階梯法中，皆是在尋找逆時針的標號三角形，我們從數據和呈現出來的圖表發現，愈往上層，標號 I 、II、 III 會愈來愈集中，形成 I 、II、 III 區。且當 $C_k, k \geq 0$ ，k 愈來愈大時，三區交點會愈接近最後所求出的根。所以我們想，要是將每一層 Q_m 中所有的點都計算出來並標號，找出每一個逆時針完全標號三角形，再疊圖比較，是否就能直接找出根的位置呢？

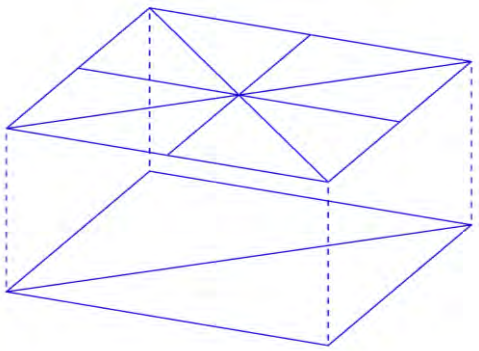
(2)標號法的優點在於可一目了然根的位置大概在何處，但每一層單位方格上每一點的數據都需經過計算，且不可縮小範圍，否則將有可能找到錯誤的根。

以標號法和階梯法為例：

※重複計算時：(在較低層算過的在較高層還要再算一次)

(I) C_{-1} 、 C_0 時狀況相同(皆要計算全部的標號)，故不討論

(II) $C_k, k \geq 1$ 時， C_{k-1} 的一個方格對到 C_k 的 4 個方格(如圖十八藍線實線部分)



(圖十八)

故階梯法只需計算 9 個標號；標號法仍須全部計算，每一層有$(2^{k+2}+1)^2$格，故要計算$(2^{k+2}+1)^2$次

※不重複計算時：(在較低層算過的數據不用再算一遍)

(I) C_{-1} 、C_0 和重複計算時狀況相同

(II) $C_k, k \geq 1$時，階梯法要算的格數仍是 9 格，而標號法要計算的格數可用C_k的格數減掉C_{k-1}的格數表示(為$(2^{k+2}+1)^2-(2^{(k-1)+2}+1)$格)，化簡後得$(3\times 2^k+1)(2^{k+2})$ 格。

不論是重複計算或不重複計算，當$k \geq 1$時，代入 k 可發現$(2^{k+2}+1)^2$或$(3\times 2^k+1)(2^{k+2})$皆遠大於 9。由此可知階梯法的效率比標號法還高，但如果將所有的計算過程交給電腦處理，這是能最簡單觀察根的位置的方法。

三、階梯算法與牛頓算法並用

由上述的討論我們可以知道，在階梯法、庫恩法、標號法中，階梯法是最有效率的，

因為牛頓算法的快速收斂區只能對沒有重根的多項式才能達到，且在無重根時牛頓算法需先找到良好初值，方能到達快速收斂區，而此良好初值的取得較為困難，所以我們綜合上述討論，採用階梯算法來解決進入牛頓算法所要求的快速收斂區的問題，之後再使用牛頓法來快速的找到所求之解。

以$f(z)=z^3+\left(\frac{2-i}{3}\right)z^2+\left(\frac{i}{9}\right)z+\frac{2-11i}{27}$為例：

$$z_k = z_{k-1} - \frac{f(z_{k-1})}{f'(z_{k-1})}, k=1,2,3,\cdots$$

$$f'(z)=3z^2+\left(\frac{4-2i}{3}\right)z+\left(\frac{i}{9}\right)$$

我們以完全標號的頂點為初值，進行牛頓迭代：

1. 若在階梯法中取 $z_{(2,j,1)}$ 為初始值,即
   取 $x_0 = 0.5 + 0.5i$
   $x_1 = 0.41941199 + 0.42521857i$
   $x_2 = 0.43860865 + 0.35924093i$
   $x_3 = 0.40501503 + 0.40901727i$
   $x_4 = 0.40910319 + 0.41143643i$
   $x_5 = 0.40908839 + 0.41145975i$
   $x_6 = 0.40904275 + 0.41142375i$

2. 若在階梯法中取 $z_{(4,j,1)}$ 為初始值,即
   取 $x_0 = 0.375 + 0.625i$
   $x_1 = 0.36625796 + 0.43745753i$
   $x_2 = 0.40841346 + 0.40815108i$
   $x_3 = 0.40907109 + 0.41143597i$
   $x_4 = 0.40906275 + 0.41142379i$
   $x_5 = 0.40906790 + 0.41142374i$
   $x_6 = 0.40906448 + 0.41141053i$

由上可知取較高層的完全標號的頂點為初值，較快速接近其根。

伍、研究結果與結論

一、以庫恩多項方程求根的方法為依據，改以尋求逆時針的完全標號三角形，減少求根的步驟，成為多項式零點階梯算法

二、牛頓方法的效率，用所需的牛頓迭代的次數來衡量，庫恩算法則可以穿越的四面體個數來衡量，而階梯法則可以階梯數來衡量；又牛頓算法一次只能算一個根，且牛頓算法有時候很不聽話，雖然很容易學，但是計算是否成功，卻是沒有保證的。但牛頓算法在找到一個足夠好的迭代出發點 z_0 時，會很迅速的逼近準確根，又因為階梯算法的優點是保證成功，所以先使用階梯算法把速迅收斂區域的位置確定，然後改用牛頓算法迅速向準確根靠近。

陸、參考文獻及其他

一、 江澤涵, 不動點類理論, 科學出版社, 1979。

二、 王則柯, 單純不動點算法基礎, 中山大學出版社,廣州,1986。

三、 王則柯, 計算的複雜性, 湖南教育出版社, 1993 。

四、 數學傳播,17 卷,3 期

五、 高中數學第一冊(龍騰文化,許志農、黃森山等編著) 。

【評語】050401

1. 更改庫恩植物栽培法的條件，使多項式根的近似值形如走階梯般一直往上攀升的被逼近。研究主題生動有趣。
2. 團體合作佳，現場報告表達清晰。
3. 僅用一例討論的方式，不足以證明階梯法較有效率。
4. 宜舉例比較不同計算方法之優劣。

作品海報

摘　要

一、階梯算法

更改庫恩植物栽培法的限制，使方程式的根形如走階梯般向上攀升，且不下降。

二、牛頓算法的限制

1. 牛頓算法一次只能算一個根，算出多項方程式 f(z)=0 的一個根 ζ 以後，用($x-\zeta$)除以原來的多項式，得到一個次數降 1 的多項式，再用牛頓算法求這個新的多項式的根，一次一次降階可以把全部的根算出來。

2. 牛頓算法有時很難控制，且計算是否成功是沒有保證的。要使牛頓算法成功，重要的是找到一個足夠好的迭代出發點。

三、比較各個求解方法，結合牛頓算法與階梯算法

階梯算法的優點是保證成功，牛頓算法的長處是進入快速收斂區後收斂極快；將兩種算法的長處結合起來：先使用階梯算法把迅速收斂區域的位置確定，再改用牛頓算法迅速向方程式的根靠近。

壹、研究動機

高中數學課程中，我們使用勘根定理、虛根成對等方法來找出實係數多項方程式的解。但當係數中包含虛數時，上述幾種方法就不適用了，所以我們著手尋找解複係數多項方程式的方法。我們在<數學傳播>月刊中看見了與所學完全不同的方法，這個方法的圖示如栽種植物般，將大籬笆放在培養皿上，然後栽種尋根的種子……，而 n 個新芽長出 n 條藤，並四處攀援，最後向上尋根。我們對此方法十分有興趣，於是將它做為我們專題研究的主題。

貳、研究目的

一、 以庫恩植物栽培法為依據，並限制尋根之規則，期能較迅速逼近複係數方程式的零點。

二、 證明尋根方法的理論依據。

三、 比較牛頓法與階梯法及其他方法的優缺點，並尋找更實在好用的方法。

參、研究設備器材

紙、筆、電腦、計算機、Geogebra。

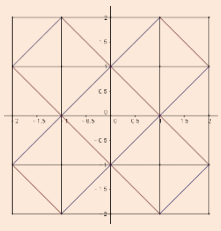
肆、研究過程與方法

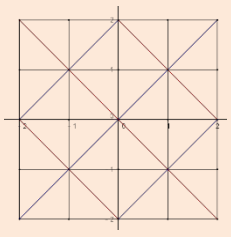
一、 多項式方程式求根的階梯算法(我們所使用的階梯法是由庫恩的論文改良而來，而庫恩法會在之後有詳細介紹)

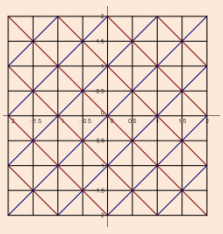
1. 階梯算法的步驟：

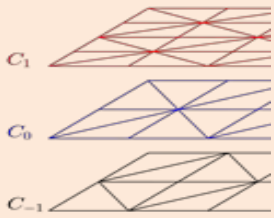
(1)首先我們把一系列複數平面 C_k, k=−1, 0, 1, 2…層層向上排好, 在上面劃線,層層均分割成三角形。除 C_{-1} 分割方式略有不同之外，其餘各層的分割規律完全相同。劃分規則如下：

對k=−1：$\{z\in C: x=p, p\in Z\}$、$\{z\in C: y=p, p\in Z\}$和 $\{z\in C: y=(2p+1)\pm x, p\in Z\}$ 分割 C_{-1} 平面(如圖一)；而對 k=0, 1, 2, 3, …，用四組平行直線：$\{z\in C: x=2^{-k}p, p\in Z\}$、$\{z\in C: y=2^{-k}p, p\in Z\}$和 $\{z\in C: y=2^{1-k}p\pm x, p\in Z\}$ 分割 C_k 平面。(如圖二及圖三)並且從 C_0 開始，每向上一層，則線的密度就增加一倍，三角形的邊就縮小二分之一（如圖四）。


（圖一）


（圖二）


（圖三）



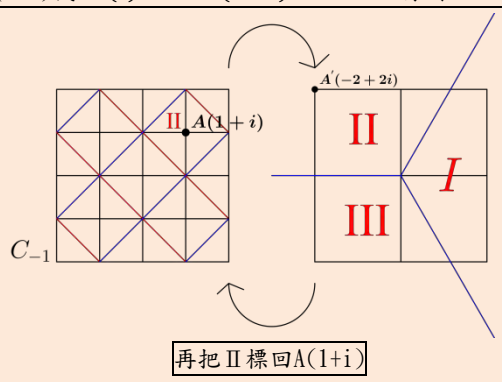
（圖四）

(2)接著我們在 C_{-1} 上取一個邊長2m格的方塊 Q_m（依庫恩算法，取 $m=\left\lceil\frac{3(1+\sqrt{2})n}{4\pi}\right\rceil$，n為方程式的次數)，再另取一個複數平面C'，用三條射線把複數平面C'分成相等的三個扇形部分I、II、 III區（如圖五）。對於 C_k 的每一 層面上的每個點z算出f(z)，

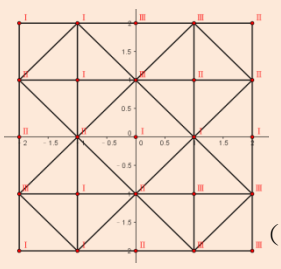
$$g(z)=\begin{cases} \text{I}, 若 f(z)=0 或 -\frac{\pi}{3}\le \operatorname{Arg} f(z)\le\frac{\pi}{3} \\ \text{II}, 若 \frac{\pi}{3}<\operatorname{Arg} f(z)\le\pi \\ \text{III}, 若 -\pi<\operatorname{Arg} f(z)<-\frac{\pi}{3} \end{cases}$$

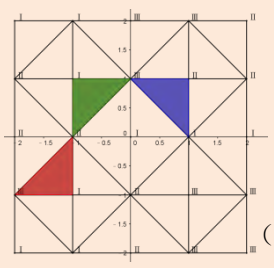
，若f(z)落在I區，z就標號I；若f(z)落在II區，z就標號II；若f(z)落在III區，z 就標號III。如此就可以把 C_k 上 Q_m 內的全部頂點依此方法完成標號。(如圖五)

將A(1+i)代入 $f(z)=z^3 \Rightarrow f(1+i)=-2+2i$ 得到A' ，再對應A' 將落在第II區

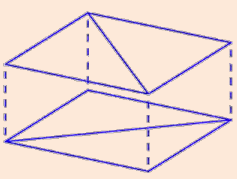


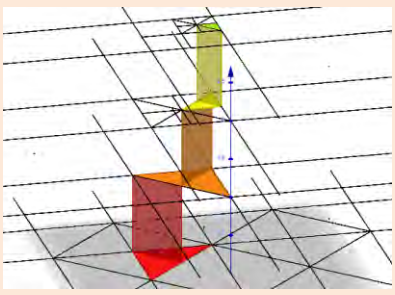
（圖五）


（圖六）


（圖七）

(3)由(圖五)我們可以知道，在 Q_m 的邊界上有三個I 、II相鄰的稜，我們從這些稜出發，在 C_{-1} 上尋找下一個I 、II相鄰的稜，直到找到的這個稜是完全標號三角形的其中一稜為止(如圖七)。在 C_{-1} 上找到完全標號三角形後， 再把 C_{-1} 與 C_0 之間的平面分成許多四面體，依庫恩的方法，在四面體的面上找出有完全標號三角形的那個面，持續此步驟直至到達 C_0 。(如圖八)


（圖八）


（圖九）

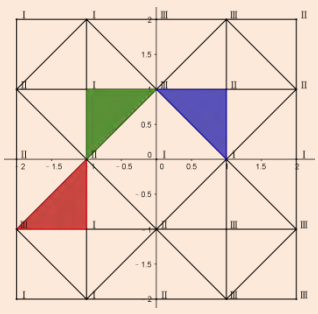
(4) $C_k, k\ge 0$ 的階梯法則：

我們將庫恩法改良為複數平面層 $C_k, k\ge 0$ 以後，遇到逆時針完全標號三角形，便建構一個三角形階梯，再往正上方穿過去，進入較高一

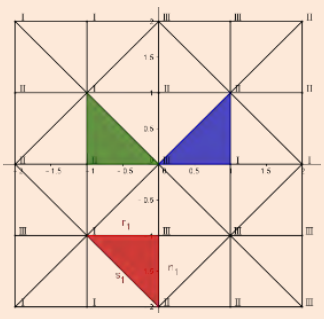
層平面的另一個縮小四分之一的完全標號三角形階梯。每踏上一個三角形階梯，就要計算等分點的f(z)，如此持續進行。因此當我們取極限時，踩上這n個階梯就能站在多項式的全部n個根上了（如圖九）。而我們會不會有極小的機率，在向上尋根的過程中就直接找到解了呢？此時平面上只會剩下兩個三角形，且其他兩個三角形不受干擾，仍會繼須往上逼近，直到找到方程式的根為止。

2. 實例操作（以$f(z)=z^3+\left(\frac{2-i}{3}\right)z^2+\left(\frac{i}{9}\right)z+\frac{2-11i}{27}$為例）

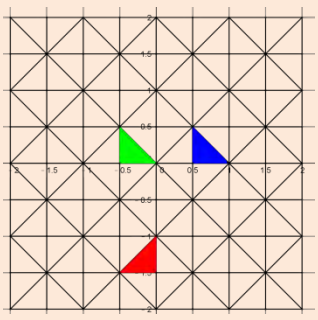
我們將冗長的計算過程結果製圖如下：（依庫恩算法，對於此三次方程式 n=3，因此我們取$m=\left\lceil\frac{3(1+\sqrt{2})\times 3}{4\pi}\right\rceil \fallingdotseq |1.729|=2$）



C_{-1}



C_0



C_1

3. 階梯法的逼近與收斂

在此仍以$f(z)=z^3+\left(\frac{2-i}{3}\right)z^2+\left(\frac{i}{9}\right)z+\frac{2-11i}{27}$為例。首先，我們先定義符號：

$z_{(k,j,i)}=C_k$ 複數平面上第 j 個逆時針完全標號三角形(我們由複數平面右方開始將三角形標號為 j_1、j_2、j_3，若有兩個逆時針完全標號三角形在同一行，則先將上方的三角形標號)標號為 i 的 z 值；

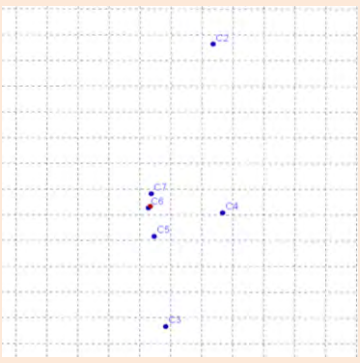
$f_k(j,i)=f(z_{(k,j,i)})$，即 C_k 複數平面上第 j 個逆時針完全標號三角形標號為 i 的 z 值代入函數所得 f(z)之值。

由表格可以發現，階梯法是可以逼近根的，當 $C_k, k\geq 0$，k 值愈大時，其值 f(z)會愈趨近於零。當我們取極限則可得到此方程式的根。

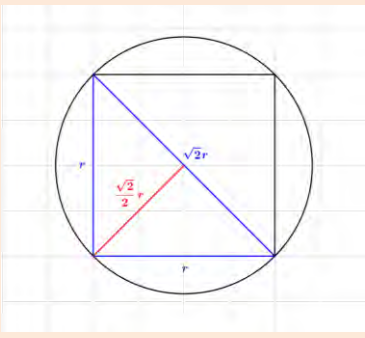
從另一方面來看，我們也能由預設根和實際根的位置或數值來觀察根的逼近。我們將預設根的位置定在三角形的內接圓圓心處(因其和三邊等距)，並比對和計算在每一層平面中，預設根和實際根的相對位置及數值上的差異。

以$f(z)=z^3+\left(\frac{2-i}{3}\right)z^2+\left(\frac{i}{9}\right)z+\frac{2-11i}{27}$的其中一根$(\frac{1}{3},\frac{2}{3})$為例，可得出如下表的結果：(表格詳情請見作品說明書)

我們將上述的點繪製於下圖中，可以發現當 k 值愈大時，其預設根的位置或數值有愈逼近實際根(紅點處)的現象(如圖十)。C_4和C_2、C_3的預設根相比，雖然數值相較之下偏離實際根，但位置上卻大大逼近了與實際根的距離，這和使用牛頓法逼近複數方程式的狀況非常雷同(稍後在牛頓法的部分會有詳細說明)；C_7和C_5、C_6的預設根相比，雖然位置稍稍偏離了實際根，但在數值上仍較接近實際根。由上述兩種不同面向的討論，我們可以得知：階梯法是可以逼近根的，並能準確的找到根的位置。



（圖十）



（圖十一）

4. 根的逼近與誤差值計算

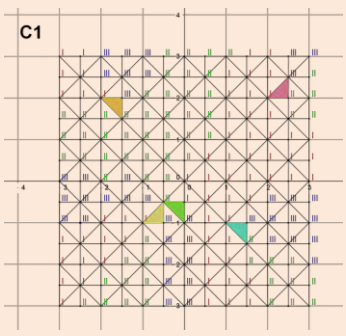
當 $C_k, k\geq 0$，我們取半徑為$\frac{\sqrt{2}}{2}\cdot 2^{-k}$的圓，也就是單位方格的外接圓，並假設欲求方程式的根皆在此圓中(如圖十一)（後稱此種圓為單位圓）。由此我們可知在C_k上的單位圓面積為$2^{-2k-1}\pi$，並推導出對於C_k和C_{k+1}單位圓的面積差為$3\pi\cdot 2^{-2k-3}$個單位長。

接著我們要定義所謂的逼近速率。我們知道當三角形階梯越向上範圍越小，即愈往上愈能逼近根，它所在的方格及單位圓也有同樣的性質。若以圓上每一個點所組成的圓，其圓面積代表在單位圓上所有根的可能性，則逼近速率即為C_k和C_{k+1}的圓面積差。

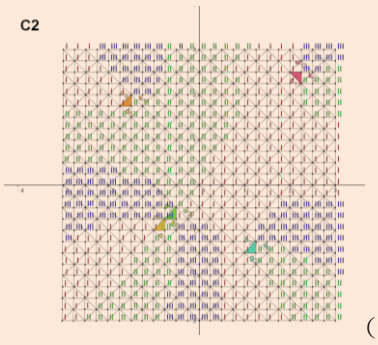
又C_k和C_{k+1}的圓面積差以及C_{k+1}和C_{k+2}圓面積差的比值：$\frac{3\pi\cdot 2^{-(2+1)-3}}{3\pi\cdot 2^{-2k-3}}=\frac{1}{4}$，則逼近速率每往上一層就減少$\frac{1}{4}$。

5. 方程式有重根的討論

首先，我們以已知解的五次方方程式$f(z)=\left[x-(-\frac{3}{5}-\frac{4}{5}i)\right]\left[x-(\frac{3}{5}-\frac{4}{5}i)\right]\left[x-(\frac{6}{5}-\frac{7}{5}i)\right]\left[x-(-\frac{8}{5}+\frac{9}{5}i)\right][x-(\frac{11}{5}+\frac{2}{5}i)]$ 進行實例操作，可得(圖十二、圖十三)：



（圖十二）



（圖十三）

如上述的舉例我們可以知道：根據庫恩法，無論是否有重根，在 C_{-1}層時皆會有 n 個起始點。而向 C_0以上的平面發展時，重根的兩個藤會分別進入極接近的兩個三角形。當我們取極限時，此兩個根會在較上方的平面重合，進而我們可以知道其為兩重根。

4. 階梯法之引理證明

我們利用下述引理，可知道其根必在完全標號三角形階梯內。(詳細證明過程如說明書)

(命題)：若複數 z 符合$|z|<1$，且幅角取值範圍為$(-\pi,\pi]$，則$\mathrm{Arg}(1+z)\leq\frac{\pi}{2}|z|$

〔引理 1〕

令$f(x)=x^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ 、 $a=\max\{|a_0|,|a_1|,\cdots,|a_{n-1}|\}$ 、 $r=\max\left\{\frac{3\sqrt{2}(2+\pi)n}{4\pi},1+\frac{na}{n-1}\right\}$ 、$R=r+\sqrt{2}$

則完全標號三角形和原點軸$\{0\}\times[-1,+\infty)$的距離都小於 R。

〔引理 2〕 $k\geq 0$ 時，C_k 平面上完全標號三角形任一頂點與多項式某個零點的距離不超過$\frac{\sqrt{2}(1+\frac{3n}{4})}{2^k}$

二、各種求複係數方程式方法的比較

1. 牛頓算法和階梯法的優缺點比較：

(I)牛頓算法一次只能算一個根，但階梯算法則可以同時求所有的根。

(II) 牛頓法須用到微分，而階梯法完全不用。

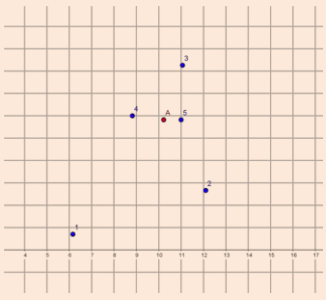
(III)牛頓法會有誤差，且愈高階時誤差愈大；階梯法完全不會有誤差。

(IV)牛頓法有尋找初始值的問題，若選得不好，有可能無法求解；階梯法不會有尋找初始值的困擾，在 C-1 平面時根即可明確的向上爬升，可知在開始尋根時，階梯法是較佳選擇。

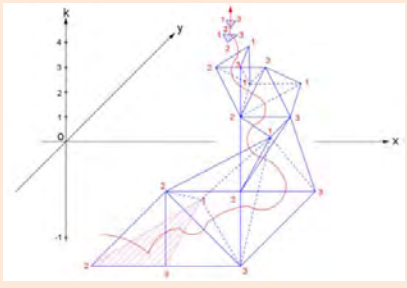
(V) 若使用牛頓算法的迭代公式：$z_k=z_{k-1}-\frac{f(z_{k-1})}{f'(z_{k-1})},k=1,2,3,\cdots$，牛頓法的計算速度較階梯法快許多。然而，牛頓在 300 年前率先提出迭代法可謂一劃時代的想法，而更影響了許多後世的數學家，因此牛頓的貢獻仍是無法抹滅的重要。

2. 使用牛頓法求複數方程式之解：

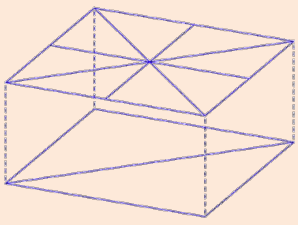
我們一般都使用牛頓法處理實數係數方程式的計算，但牛頓法也可用於複係數方程式嗎？答案是肯定的，其求解圖形為在高絲平面上散亂的點，但向欲求方程式的根逼近。下圖為示意圖：(紅點為解的實際位置)(1 到 5 為逼近順序)



(圖十四)



(圖十五)



(圖十六)

2. 庫恩法和階梯法

(1)庫恩法的步驟和階梯法的(1)~(3)相同，但$C_k,k\geq 0$時，庫恩法須將兩層平面間分割成許多四面體，並在這些四面體中找到完全標號三角形後穿梭(如圖十五)。

(2)庫恩法需要一直剖分四面體，分析各個四面體中標號三角形在空間中的相對位置，再沿著這些面向上生長，但途中有可能會先向下再向上長，過程較複雜。階梯法就簡單的多，不僅去除了向下的可能性，並只需向上尋找完全標號三角形即可。

3. 標號法與階梯法

(1)由於在階梯法中，皆是在尋找逆時針的標號三角形，我們從數據和呈現出來的圖表發現，愈往上層，標號Ⅰ、II、 III 會愈來愈集中，形成Ⅰ、II、 III 區。且當$C_k,k\geq 0$k 愈來愈大時，三區交點會愈接近最後所求出的根。所以我們想，要是將每一層Q_m中所有的點都計算出來並標號，找出每一個逆時針完全標號三角形，再疊圖比較，是否就能直接找出根的位置呢？答案是肯定的。

(2)標號法的優點在於可一目了然根的位置大概在何處，但每一層單位方格上每一點的數據都需經過計算，且不可縮小範圍，否則將有可能找到錯誤的根。

計算後發現標號法所要計算的標號數遠大於階梯法，由此可知階梯法的效率比標號法還高，但如果將所有的計算過程交給電腦處理，這是能最簡單觀察根的位置的方法。

三、階梯算法與牛頓算法並用

由上述的討論我們可以知道，在階梯法、庫恩法、標號法中，階梯法是最有效率的。因為牛頓算法的快速收斂區只能對沒有重根的多項式才能達到，且在無重根時牛頓算法需先找到良好初值，方能到達快速收斂區，而此良好初值的取得較為困難，所以我們綜合上述討論，採用階梯算法來解決進入牛頓算法所要求的快速收斂區的問題，之後再使用牛頓法來快速的找到所求之解。

以$f(z)=z^3+\left(\frac{2-i}{3}\right)z^2+\left(\frac{i}{9}\right)z+\frac{2-11i}{27}$為例：$z_k=z_{k-1}-\frac{f(z_{k-1})}{f'(z_{k-1})},k=1,2,3,\cdots$ 、 $f(z)=3z^2+\frac{4-2i}{3}z+(\frac{i}{9})$

我們以完全標號的頂點為初值，進行牛頓迭代：

1. 若在階梯法中取 $Z_{(2,j,1)}$ 為初始值, 即

取$x_0=0.5+0.5i$

$x_1=0.41941199+0.42521857i$

$x_2=0.43860865+0.35924093i$

$x_3=0.40501503+0.40901727i$

$x_4=0.40910319+0.41143643i$

$x_5=0.40908839+0.41145975i$

$x_6=0.40904275+0.41142375i$

2. 若在階梯法中取 $Z_{(4,j,1)}$ 為初始值, 即

取$x_0=0.375+0.625i$

$x_1=0.36625796+0.43745753i$

$x_2=0.40841346+0.40815108i$

$x_3=0.40907109+0.41143597i$

$x_4=0.40906275+0.41142379i$

$x_5=0.40906790+0.41142374i$

$x_6=0.40906448+0.41141053i$

由上可知取較高層的完全標號的頂點為初值，較快速接近其根。

伍、研究結果與結論

一、 以庫恩多項方程求根的方法為依據，改以尋求逆時針的完全標號三角形，減少求根的步驟，成為多項式零點階梯算法

二、 牛頓方法的效率，用所需的牛頓迭代的次數來衡量，庫恩算法則可以穿越的四面體個數來衡量，而階梯法則可以階梯數來衡量；又牛頓算法一次只能算一個根，且牛頓算法有時候很不聽話，雖然很容易學，但是計算是否成功，卻是沒有保證的。但牛頓算法在找到一個足夠好的迭代出發點時，會很迅速的逼近準確根，又因為階梯算法的優點是保證成功，所以先使用階梯算法把速迅收斂區域的位置確定，然後改用牛頓算法迅速向準確根靠近。