

國立交通大學

理學院科技與數位學習學程

碩士論文

科學計算與視覺化
Scientific Calculation and Visualization

研究生：李國寧

指導教授：陳永富 教授

中華民國 一百零一 年 六月

科學計算與視覺化
Scientific Calculation and Visualization

研究生：李國寧

Student : Kuo-Ning Li

指導教授：陳永富

Advisor : Yung-Fu Chen

國立交通大學
理學院科技與數位學習學程
碩士論文

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Degree Program of E-Learning

June 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年六月

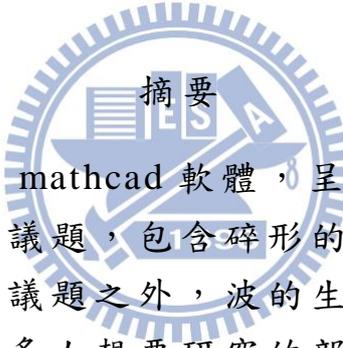
科學計算與視覺化

研究生：李國寧

指導教授：陳永富 教授

國立交通大學理學院碩士在職專班

科技與數位學習學程



摘要

在本篇論文，利用 mathcad 軟體，呈現出近代想要許多人想要研究的新興物理相關議題，包含碎形的圖形介紹、繪製與不同的生成方法，除了碎形議題之外，波的生成與波和波之間所產生的干涉圖案也是引發許多人想要研究的部分。另外，近年來彈簧擺的擺動軌跡圖案與擺動軌跡渾沌現象的討論也是相當重要的概念，經由基本的物理概念，給予不同的初始條件，表現出不同的擺動軌跡圖案，讓人可以不用經過實驗，即可以利用電腦畫出軌跡；此外繩結的紐結理論，也是一個有趣的議題，利用數學的方程式，模擬用繩結的交錯產生許多不同的圖案。這些議題的背後都有相關的數學方程式，我們可以透過數學軟體的運算，並呈現圖形之生成與變化，可以提供給未來相關的研究參考。

關鍵詞：Mathcad、碎形、準晶格、彈簧擺、紐結理論

Scientific Calculation and Visualization

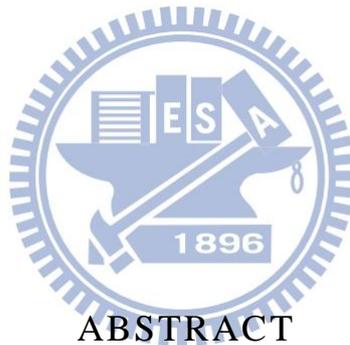
student: Kuo-Ning Li

Advisors: Dr. Yung-Fu Chen

Degree Program of E-Learning

College of Science

National Chiao Tung University



ABSTRACT

The purpose of this thesis is to study recently brand new physics topic with mathematical software. First of all, we introduce “fractal geometry” by representing the figures and discussing various generating methods. Also, we focus on the patterns of wave interference, in which we simulate the quasicrystal pattern. Secondly, we analyze spring pendulum without experiments and show the swing trajectories varied with initial conditions by the mathcad software only. Finally, we study the knot theory. We can generate many different and interesting knots with mathematical formulas. All the topics can be visualized with formulas and mathematical software.

誌謝

在交通大學求學的日子，轉眼飛逝，忙碌、充實卻不寂寞的學習經歷，成了我最難忘的回憶。

首先，非常感謝我的指導教授，陳永富教授，做學問上，老師總是耐心、用心地指導我；生活中為人處事的道理，老師也適時地建議與協助，老師給予我的不只是知識的豐收，對學問的熱情與對學生的關心，同時也給了我很多知識以外的想法，在此向老師致上最高的感謝。

梁興弛助教的耐心協助，令我受益良多，尤其在我論文上的提醒與指導，使我可以持續穩定的進步，也感謝一同走過這段美好日子的同學們，鳳蘭、佳琪、錦玫、瑋涵、長壽，他們給了我許多的信心與鼓勵。我很榮幸可以在這美麗的校園遇到一群美好的朋友，陪伴我度過研究所的生活。

感謝我的父母親，無怨無悔地不停為我付出，岳父岳母的體諒，還有我的太太佳勳，不論在工作或家庭上都給我全力的支持，讓我可以無後顧之憂，全力以赴，努力完成我的碩士學位，如果沒有他們，我沒有辦法完成我的碩士學位。

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
圖目錄	vi
表目錄	ix

目 錄

一、	緒論.....	1
	1.1 研究動機.....	1
	1.2 本文架構.....	2
二、	碎形圖案的視覺化.....	4
	2.1 樹木的分枝.....	6
	2.2 雪花碎形.....	11
	2.3 Julia sets & Manderbrot sets.....	16
	2.3.1 Julia sets.....	16
	2.3.2 Mandelbrot sets.....	24
	2.4 等值線繪圖.....	27
	2.5 經由 IFS 的方式產生碎形圖騰.....	29
	2.5.1 IFS 概念.....	29
	2.5.2 貼片.....	31
	2.5.3 IFS 等值線圖.....	38
附錄 2.1	分形樹 MathCad 程式碼.....	44
附錄 2.2	Koch 雪花曲線 MathCad 程式碼.....	48
附錄 2.3	Julia sets MathCad 程式碼.....	50
附錄 2.4	Mandelbrot sets MathCad 程式碼.....	51
附錄 2.5	Julia sets 等值線圖 MathCad 程式碼.....	52
附錄 2.6	IFS 雪花 MathCad 程式碼.....	54
附錄 2.7	山脈圖形等值線圖 MathCad 程式碼.....	55
附錄 2.8	山脈等值線圖清晰化 MathCad 程式碼.....	57

三、	波的干涉圖騰之視覺化.....	59
3.1	波.....	59
3.2	波的干涉.....	60
3.3	準晶格圖騰.....	62
3.4	波的變化.....	64
附錄 3.1	點波源干涉 MathCad 程式碼.....	66
附錄 3.2	波的干涉(時間變動) MathCad 程式碼.....	67
附錄 3.3	波的干涉(隨時間的變動變化) MathCad 程式碼.....	68
四、	彈簧擺的視覺化.....	69
4.1	二維空間彈簧擺.....	71
4.2	三維空間彈簧擺.....	75
4.3	傅立葉轉換.....	80
附錄 4.1	二維彈簧擺 MathCad 程式碼.....	84
附錄 4.2	彈簧擺三維擺動 MathCad 程式碼.....	86
五、	Knot & Torus 的視覺化.....	88
5.1	Torus Knot.....	89
5.2	Decorative Knot.....	93
5.3	Knotted Tori.....	95
附錄 5.1	Torus knot MathCad 程式碼.....	97
附錄 5.2	Decorative Knot MathCad 程式碼.....	98
附錄 5.3	Knotted tori MathCad 程式碼.....	99
Reference.....		100

圖目錄

圖 2-1	碎形圖形 acking.....	4
圖 2-2	函數圖形 $f(x)$, $a=0.5, b=5$	5
圖 2-3	Cantor 三分集.....	5
圖 2-4	分形樹.....	6
圖 2-5	直角坐標系.....	7
圖 2-6	複數平面.....	8
圖 2-7	分形樹第一層與分形樹示意圖.....	9
圖 2-8	分形樹第二層示意圖.....	10
圖 2-9	雪花碎形圖.....	11
圖 2-10	Koch 曲線示意圖.....	11
圖 2-11	Koch 曲線第一層示意圖.....	12
圖 2-12	Koch 曲線第二層示意圖.....	12
圖 2-13	搜尋區塊示意圖.....	17
圖 2-14	區塊分割示意圖.....	18
圖 2-15	區塊編碼示意圖.....	19
圖 2-16	二維轉一維示意圖.....	19
圖 2-17	$[-1,1]$ 平面切成 100×100 格圖.....	20
圖 2-18	$[-1,1]$ 平面切成 100×100 格隨機灑點圖.....	20
圖 2-19	不同 C 值 Julia sets 圖形.....	21
圖 2-20	Julia sets 疊代 50 次示意圖.....	22
圖 2-21	Julia sets 疊代 100 次示意圖.....	22
圖 2-22	Julia sets 疊代 500 次示意圖.....	22
圖 2-23	Julia sets 疊代 1000 次示意圖.....	23
圖 2-24	Mandelbrot sets 次方的變化圖.....	25
圖 2-25	(a)Mandelbrot sets (b) Julia sets 相似圖形.....	26
圖 2-26	Julia sets 等值線圖.....	27
圖 2-27	Mandelbrot sets 等值線圖.....	28
圖 2-28	Julia sets $c=-0.745429+0.113008i$ 不同範圍等值線圖..	28

圖 2-29 仿射變換相關變化圖.....	30
圖 2-30 Koch 雪花圖形貼片示意圖.....	32
圖 2-31 利用貼片灑 2000 點形成 Koch 曲線.....	34
圖 2-32 利用貼片灑 10000 點形成 Koch 曲線.....	34
圖 2-33 利用貼片灑 50000 點形成 Koch 曲線.....	34
圖 2-34 利用貼片灑 100000 點形成 Koch 曲線.....	35
圖 2-35 IFS 珊瑚圖形.....	35
圖 2-36 IFS 樹木圖形.....	36
圖 2-37 IFS 羊齒葉圖形.....	37
圖 2-38 IFS 羊齒葉擺動圖形.....	38
圖 2-39 80 × 80 格分割圖.....	39
圖 2-40 IFS 山脈圖.....	40
圖 2-41 IFS 山脈等值線圖.....	40
圖 2-42 IFS 山脈圖取 0.25 次方連續繪製 1 次圖.....	42
圖 2-43 IFS 山脈圖取 0.25 次方連續繪製 5 次圖.....	42
圖 2-44 IFS 山脈圖取 0.25 次方連續繪製 10 次圖.....	43
圖 3-1 $\sin 5x$ 函數圖形 $[-\pi, \pi]$	60
圖 3-2 波的干涉圖形.....	62
圖 3-3 不同數量點波圓干涉形成 Quasi-crystal 圖.....	63
圖 3-4 高斯分布圖形.....	64
圖 3-5 利用高斯分布控制波函數干涉圖形.....	65
圖 4-1 彈簧擺示意圖.....	69
圖 4-2 彈簧擺擺動示意圖.....	69
圖 4-3 單擺受力示意圖.....	70
圖 4-4 彈簧運動示意圖.....	70
圖 4-5 彈簧擺二維平面示意圖.....	71
圖 4-6 彈簧擺擺動軌跡 ($u=4, v_{x_0}=1$).....	74
圖 4-7 彈簧擺擺動軌跡 ($u=7, v_{x_0}=4.5432$).....	75
圖 4-8 彈簧擺擺動軌跡等值線圖 ($u=6, v_{x_0}=1.1001$).....	75
圖 4-9 彈簧擺三維空間示意圖.....	76
圖 4-10 不同初始條件下的彈簧擺軌跡.....	79

圖 4-11 非混沌彈簧擺軌跡.....	81
圖 4-12 快速傅立葉轉換圖 4-11 的 x, z 方向結果.....	82
圖 4-13 混沌彈簧擺軌跡.....	82
圖 4-14 快速傅立葉轉換圖 4-13 的 x, z 方向結果.....	83
圖 5-1 Torus 形成示意圖.....	88
圖 5-2 Torus 旋轉軸 $d = 0$ 示意圖.....	89
圖 5-3 Torus 示意圖.....	89
圖 5-4 $(1, q)$, q 變化 Torus Knot 圖.....	90
圖 5-5 $(2, q)$, q 變化 Torus Knot 圖.....	90
圖 5-6 $(3, q)$, q 變化 Torus Knot 圖.....	91
圖 5-7 $(p, q) = (1, 3)$ a, d 變化 Torus Knot 圖.....	91
圖 5-8 mathcad 生成圖(未旋轉).....	92
圖 5-9 mathcad 生成圖投影.....	92
圖 5-10 mathcad 生成圖(經旋轉).....	92
圖 5-11 Decorative Knot 相似於 Torus knot 的結果.....	93
圖 5-12 Decorative Knot $m > n, p - q = 5$ 圖.....	94
圖 5-13 Decorative Knot 加入不同參數示意圖.....	95
圖 5-14 knotted tori 不同粗細圖.....	96
圖 5-15 $(2, q)$ q 變化 knotted tori 示意圖.....	96

表目錄

表 2-1 Koch 曲線第一層到第三層角度表.....	13
表 2-2 Koch 曲線角度整理圖.....	14
表 2-3 IFS 仿射變化表.....	31
表 2-4 利用貼片參數形成 Koch 雪花曲線表.....	33
表 2-5 IFS 生成珊瑚參數表.....	35
表 2-6 IFS 生成樹木參數表.....	36
表 2-7 IFS 生成山脈參數表.....	39



第一章 緒論

1.1 研究動機

近代研究中，由於科技進步神速，使得電腦的運算可以一日千里，因此，有許多人可以利用電腦來進行設計與創造，還有許多研究人員會藉由電腦的運算，來呈現出實驗的數據結果，其中實驗數據的產生雖有決定性的影響，倘若我們可以利用電腦軟體的輔助，將這些實驗和數據視覺化，並透過視覺化的方式，來呈現出更多不同的實驗樣貌，甚至經由將實驗過程圖像的呈現，我們可以更直觀，而且更容易地獲得一些實驗的結果。

在呈現圖像的方式中，許多的呈現方法都需要透過程式設計來產生，然而程式的撰寫對於一般人來說並不容易入手，因此，我們尋求一種比較簡單的方式，透過較為直覺式介面結合與標準的數學標記法的 mathcad 數學軟體，經由軟體中直覺式的介面，我們可以將我們實驗所需的方程式直接鍵入軟體中，並透過軟體中標準的數學標記法，我們不需要額外的去學習程式的語言，利用軟體數學介面中，所見即所得的運算方式，來畫出我們所想要呈現出來的實驗方程式。

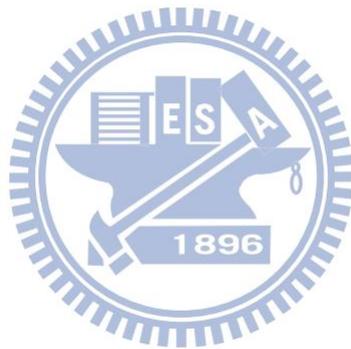
近年來，渾沌現象是許多人想要研究的議題，而碎形與渾沌是伴隨非線性系統研究而出現的，其中在動力行為渾沌的背後，常常會有靜態的碎形結構，經由了解碎形的許多概念，使我們擁有不同的角度可以去觀察我們原來的大自然[1]，同時也經由碎形幾何的研究，我們可以嘗試表現出大自然的樣貌，除了碎形之外，仍然有許多的研究議題都值得討論，其中包含利用波干涉所產生的圖形與在物理學上的彈簧擺，以及紐結理論，這些議題都具有豐富變化的圖騰與軌跡，因此，我們要利用這幾個議題為例子，透過 mathcad 軟體來繪製與呈現，

使得實驗和方程式不僅只是枯燥無味的單一結果，透過視覺化的表現，讓實驗的過程與結果可以直接地呈現在電腦畫面之中，而往後的研究者可以經由本文做為基石，輕易的上手來繪製各種圖案。

1.2 本文架構

在本論文將呈現出如何將依些新興的物理議題視覺化，並提供一些範例與相關的程式碼，讓讀者可以依循設計出自己想要實驗的目標，內容分為兩大部分，第一部分於第二章時，會介紹碎形的相關概念與多種圖案，像是樹枝、Koch 雪花、Julia sets、Mandelbrot sets 與 IFS 疊代等，並且利用 mathcad 軟體，透過運算相關的方程式，執行疊代，產生變化多端的碎形圖騰，並利用第二章繪製碎形圖案的過程中，藉由灑點、等高線圖等等的方式，來建立一些 mathcad 不同方式呈現圖形的方法；第二部份於第三章到第五章，利用第二章中所介紹到的 mathcad 軟體功能與相關的數學概念為基礎，表現出近代物理中，引人入勝的許多議題，包含第三章中波的干涉與生成，利用波的參數方程式，經由疊加運算，呈現出相關的波的圖案與波和波之間的干涉圖騰，在其後加入透過波的干涉所產生的準晶格圖騰的繪製與呈現，讓波的性質得以完整的呈現；第四章將討論基礎物理中的彈簧擺，藉由討論彈簧擺的擺動，並且藉由彈簧擺擺動的方程式，經由不同的初始條件，呈現出差異性極大的軌跡圖案，同時藉由程式的運算，我們也可以就像是直接在 mathcad 軟體中做實驗，畫出各種條件下的軌跡圖形，並且探討彈簧擺擺動過程中，配合快速傅立葉轉換，檢驗彈簧擺軌跡圖形是否呈現出渾沌的情況；在第五章中，討論的是近代常見的紐結理論，藉由圖形的呈現，我們可以更容易觀察紐結圖形的變化，同時也可以藉由繪圖，讓過去無法一一繪製呈現的數學方程式，可以

經由 mathcad 軟體，將這些原本生冷的方程式，可以活潑的呈現方程式之美。



第二章 碎形圖案的視覺化

17 世紀開始，眾所皆知的數學家與哲學家萊布尼茲(Leibniz)在相當於現今的拓樸學中提到 "the straight line is a curve, any part of which is similar to the whole"，將一條直線視為一條曲線，而且在其中取任一截斷線段，都會跟像直線一樣，相似於整體；此外他也提及，經由一個圓形中在最大半徑下填滿三個圓形，在旁邊空隙中也填入較小的圓形，經由不斷的自我重複程序，產生一個生生不息的圓形，如下圖 2-1，並稱之為"packing"，提出了自我相似性的看法。

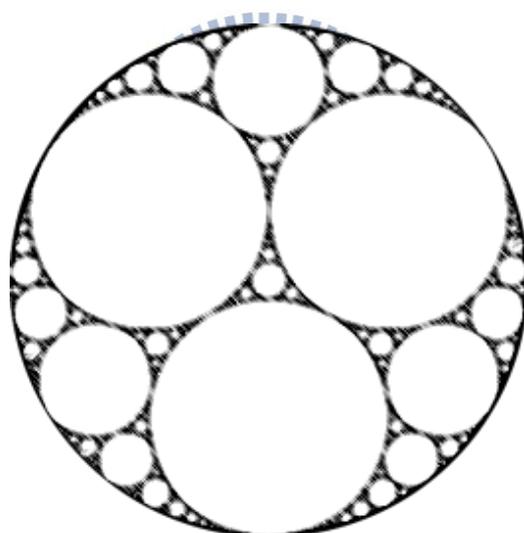


圖 2-1 碎形圖形 Packing

在 1872 年，在討論函數的極限與連續性著名的德國數學家維爾斯特拉斯(Karl Weierstrass)也提出一個方程式 Weierstrass function：

$$f(x) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi x) \quad (2.1)$$

其中 $0 < a < 1$ 且 b 為一個正奇數，這是一個處處連續，處處不可微分的函數，由函數圖形(圖 2-2)也可以發現到，他也具有很高的自我相似性，同時這個圖形也可以說是第一個被拿來做為碎形研究的圖形。

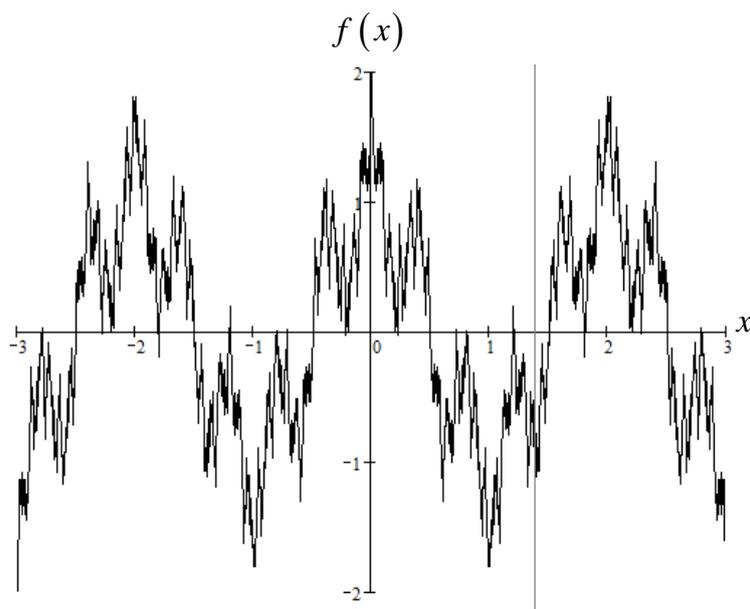


圖 2-2 函數圖形 $f(x)$, $a = 0.5, b = 5$

在 1883 年發展構成數學基本理論，集理論(set theory)的德國的數學家康托爾(Georg Cantor)，製造出一個 Cantor 三分集(Cantor ternary set)，在一段線段中刪除中間 $\frac{1}{3}$ 的部分，並且依序下去，將每一個新的線段的中間 $\frac{1}{3}$ 刪去，成為下一個新的圖形，如下圖 2-3 所示[2]。

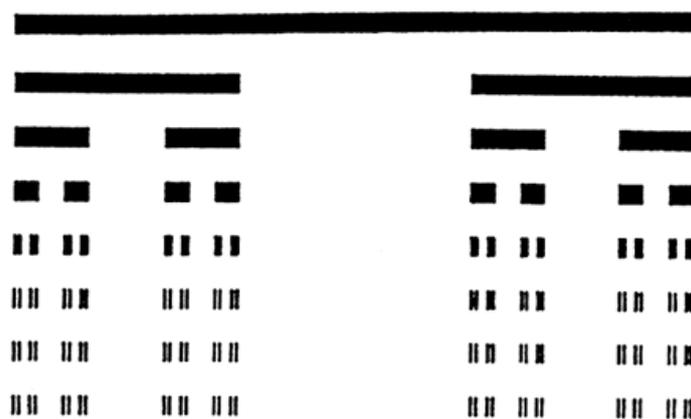


圖 2-3 Cantor 三分集

在 20 世紀之後，由於電腦科技的進步，人們有機會利用電腦來

呈現更多的碎形現象，並且經由曼德布拉特(Mandelbrot)發表”英國的海岸線有多長？”(“How Long Is the Coast of Britain?)-之後，人們對於碎形開始有了更深的一層定義與認識，也開啟了人們對於碎形的研究 [1]。

2.1 樹木的分枝

大自然中有許多植物為了生長，必須把養分輸送到每一個地方，而樹木之所以長出許多的分枝，目的就是為了輸送養分，這些自然界中的景象，巧妙地產生了許多 fractal 的結果[3]。經由電腦軟體的繪製，我們可以利用 mathcad 軟體，利用數學式子，以分層疊代的方式，將一條線段，經過平移，縮小、旋轉等方法繪製出像是樹枝的圖形。一開始我們可以畫出單獨樹枝的形體，圖 2-4(a)，接下來經由一些旋轉與縮小步驟，將原來的樹枝形狀作一些變化。在經過幾次數學運算或幾次的轉換偏移旋轉之後，即可產生下圖 2-4(b)的圖形[4]。

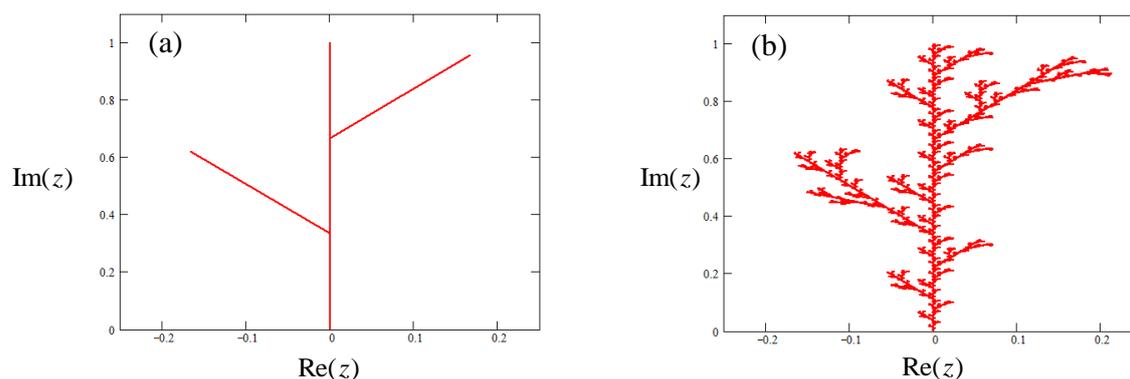


圖 2-4 分形樹

在 mathcad 軟體中，我們需要經由兩點的位置來繪製點與點連接的線段，在一般平面中，描述點的座標位置通常使用 (x, y) 來表示，

當描述兩線段的旋轉或偏移時，會產生許多相對應的係數，例如，將點 $A(x_1, y_1)$ 與點 $B(x_2, y_2)$ 的連線以點 A 為圓心旋轉 θ 角至點 $C(x_3, y_3)$ ，形成 AC 線段，如圖 2-5 所示：

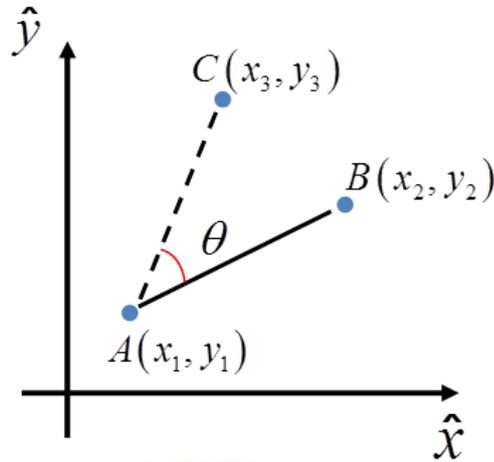


圖 2-5 直角坐標系

若用數學算式描述 AB 和 AC 線段的關係，

$$\begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} \quad (2.2)$$

則

$$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} \quad (2.3)$$

上式是新的點的位置，經由一個 2×2 的矩陣轉換，而得到的結果。

倘若利用複數平面的呈現，令 $z = x + iy$ ，經由複數的定義，可使歐式平面上的點轉換為：

$$z = x + iy = \sqrt{(x^2 + y^2)} \left(\frac{x}{\sqrt{(x^2 + y^2)}} + i \frac{y}{\sqrt{(x^2 + y^2)}} \right) = re^{i\phi} \quad (2.4)$$

而新的位置經由旋轉 θ 的 z' 點為

$$z' = x' + iy' = re^{i\theta'} = re^{i(\phi + \theta)} \quad (2.5)$$

由上述公式可以轉換，令 $z_1 = x_1 + iy_1$, $z_2 = x_2 + iy_2$, $z_3 = x_3 + iy_3$ ，

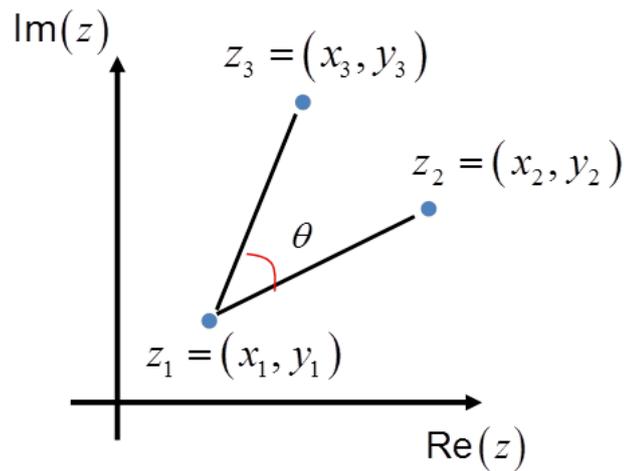


圖 2-6 複數平面

由圖 2-6 可知

$$z_3 - z_1 = (z_2 - z_1)e^{i\theta} \quad (2.6)$$

則

$$z_3 = z_1 + (z_2 - z_1)e^{i\theta} \quad (2.7)$$

由(2.5)式可知，將原來的 x, y 平面轉呈複數平面之後，在處理旋轉的角度時，只要乘上 $e^{i\theta}$ 即可，若是需要改變原來的長度，只要再乘上長度的係數即可。因此一開始我們要繪製出將一條由下而上的線段 $\frac{1}{3}$ 處向外左開 60 度角，並於線段的 $\frac{2}{3}$ 處向右開 60 度角，如圖 2-4(a)所示。因此我們需要利用點與點之間的連線繪製出 $\overline{ab}, \overline{bf}, \overline{bc}, \overline{ce}, \overline{cd}$ ，產生類似樹枝的圖形，示意圖如圖 2-7(a)所示。但是實際畫圖時，我們必須按照 $a, b, b, f, b, c, c, e, c, d$ 的順序才可以順利的繪製出圖形。因為在 mathcad 軟體中，點和點之間的連線是由該點的序數所決定，為了使圖形中各點的位置可以描述出分形樹的圖形。因此我們必須利用 10 個點來描述這個圖形[4]，第一個點到第九個點的相對應位置如下圖 2-7(b)：

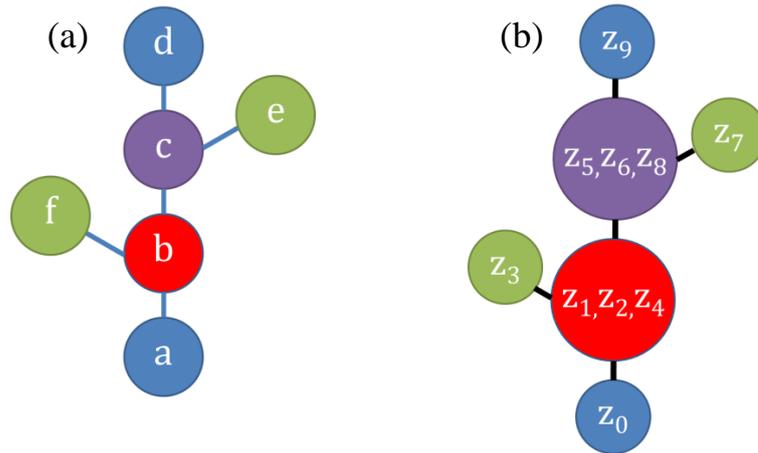


圖 2-7 分形樹第一層與分形樹示意圖

首先在複數平面上，先定義 z_0 和 z_9 起始兩個點，令 $z_0 = 0, z_9 = e^{i\frac{\pi}{2}}$ ，來決定分形樹主幹的首末位置，接下來利用 z_0 和 z_9 兩點，定義出中間的枝節部分，在 $\overline{z_0 z_9}$ 的 $\frac{1}{3}$ 處向外生長出的 $\frac{1}{3}\overline{z_0 z_9}$ ，並且將新長出的枝節旋轉 $\frac{\pi}{6}$ ，因此先定義 z_1 ，在原來 $\overline{z_0 z_9}$ 的 $\frac{1}{3}$ 處，如式子 2.8

$$z_1 = \frac{1}{3}(z_9 - z_0) + z_0 = \frac{2}{3}z_0 + \frac{1}{3}z_9 \quad (2.8)$$

接下來令 z_3 和 z_7 分別為向外分枝點的位置， z_3 由原本 z_1 出發的 $\overline{z_5 z_1}$ 向左旋轉 $\frac{\pi}{6}$ ， $z_3 = z_1 + (z_5 - z_1)e^{i\frac{\pi}{6}}$ ，同理， z_7 由原本 z_8 出發的 $\overline{z_8 z_9}$ 向左旋轉 $-\frac{\pi}{6}$ ， $z_7 = z_8 + (z_9 - z_8)e^{i\frac{-\pi}{6}}$ 依序發展出 z_1 到 z_8 ，剩下八個點的位置表示為：

$z_1 = \frac{2}{3}z_0 + \frac{1}{3}z_9, z_2 = z_1, z_4 = z_1, z_5 = \frac{1}{3}z_0 + \frac{2}{3}z_9, z_6 = z_5, z_8 = z_5,$
 $z_3 = z_1 + (z_5 - z_1)e^{i\frac{\pi}{6}}, z_7 = z_8 + (z_9 - z_8)e^{i\frac{-\pi}{6}}$ 經由定義 z_0 到 z_9 我們可以利用相同的繪製方式來增加點的位置，要讓每一個線段都可以發展出更小的線段，換句話說，就是在藉由疊代的方式，產生更多的點來繪製出更細微的樹枝。接下來定義點 $z_{10} = z_0, z_{19} = z_1$ ，並藉由定義 z_{10} 和 z_{19} 後可以決定 z_{11} 到 z_{18} 的點，接下來定義 $z_{20} = z_2, z_{29} = z_3$ ，

如法炮製後，經由不斷的定義新的點之後 $z_{30} = z_4, z_{39} = z_5, z_{40} = z_6, z_{49} = z_7, z_{50} = z_8, z_{59} = z_9$ ，並經由決定首尾兩點，發展出其他的枝節，產生第二層的疊代關係，示意圖如下圖 2-8。

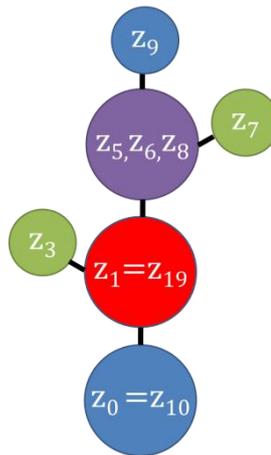


圖 2-8 分形樹第二層示意圖

因為實際上所運用的枝節線段為 5 段，所以第一層用了 10 個點之後，第二層必須用到 5×10 個點來畫出圖形，產生 5^2 小段，進而由此可以推估，在繪製第三層時，延續上述的方法，我們必須再將原來 5^2 小段中的每一段再分成 10 個點，因此必須用到序數為 $10 + 50 + 250$ ，經過不斷的決定疊代線段的首尾兩點，並且利用首尾兩點的相對位置決定該層疊代的其他枝節，我們可以利用這樣的方式，不斷地繪製下去第四、五、六……層，形成茂密的樹枝圖形，相關程式碼如附錄 2.1。

2.2 雪花碎形

在研究 fractal 相關圖形中還有一個著名的圖形，就是 Koch snowflake，如圖 2-9 所示：

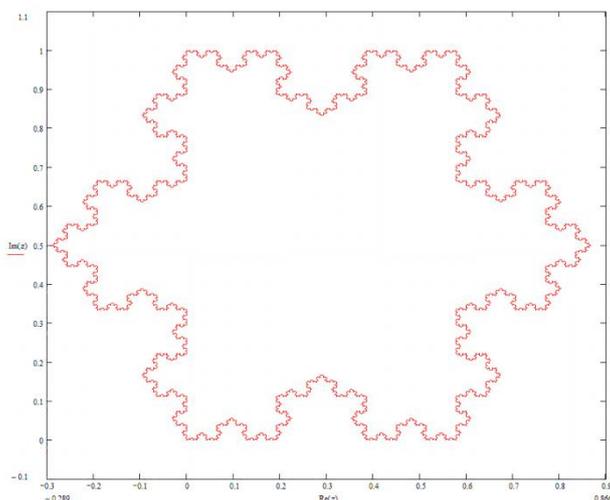


圖 2-9 雪花碎形圖

Koch 曲線具有碎形的自我相似性，經由不斷的延伸，產生無窮無盡的 Koch 曲線。雪花的原始概念為，將原有的直線的 $\frac{1}{3}$ 處截去，並將 $\frac{1}{3}$ 長度的直線，旋轉 $\frac{\pi}{3}$ 之後產生一新線段，接著將原來的 $\frac{1}{3}$ 長度旋轉 $-\frac{2\pi}{3}$ 皆在新線段的末端，最後再接回原來的直線，如圖 2-10 所示[5]：

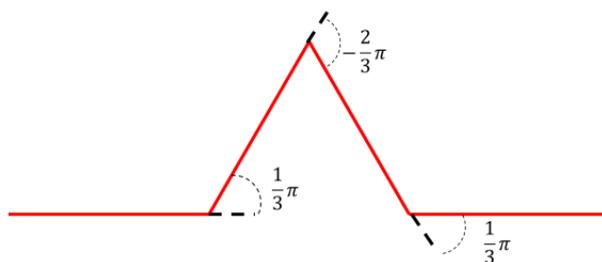


圖 2-10 Koch 曲線示意圖

因此由一正三角形出發，一開始先將第一層三角形先繪製出來，一樣

是藉由複數平面的觀念，由 $z_0 = 0$ 開始， $z_1 = 1i$ ， $z_2 = 1i \cdot e^{-\frac{2}{3}i\pi}$ ，就可以畫出正三角形，如圖 2-11

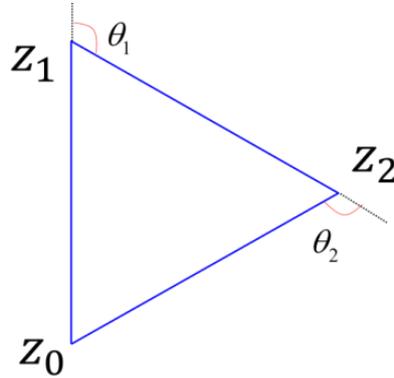


圖 2-11 Koch 曲線第一層示意圖

由圖 2-10，觀察可以得到 $\theta_1 = -\frac{2}{3}\pi$ ， $\theta_2 = -\frac{2}{3}\pi$ 。接下來我們觀察第二層的圖 2-12，

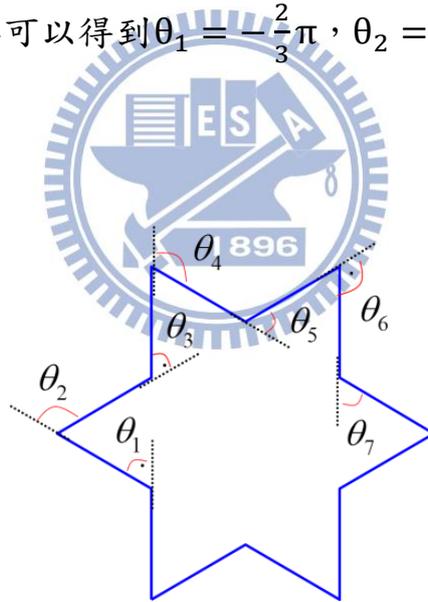


圖 2-12 Koch 曲線第二層示意圖

由 $z_0 = 0$ 開始， $z_1 = \frac{1}{3}i$ ， $z_2 = z_1 + \frac{1}{3}i \cdot e^{\frac{1}{3}i\pi}$ ， $z_3 = z_2 + (z_2 - z_1)e^{-\frac{2}{3}i\pi}$ ……，依此類推，得到每次的旋轉角度分別為分 $\theta_1 = \frac{1}{3}\pi$ ， $\theta_2 = -\frac{2}{3}\pi$ ， $\theta_3 = \frac{1}{3}\pi$ ， $\theta_4 = -\frac{2}{3}\pi$ ， $\theta_5 = \frac{1}{3}\pi$ ……，角度的旋轉，每四個一組，可由表 2-1 得知。

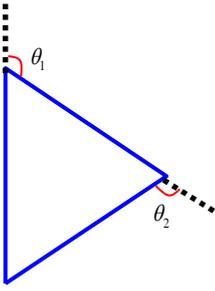
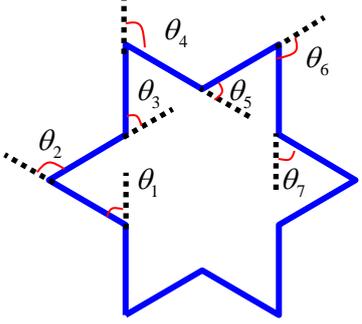
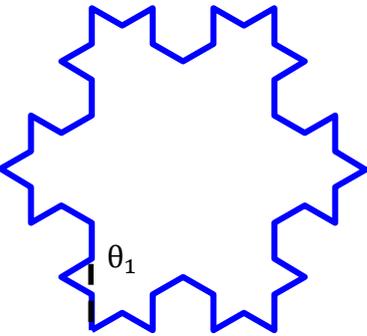
	$\theta_1 = -\frac{2}{3}\pi, \theta_2 = -\frac{2}{3}\pi$
	$\theta_1 = \frac{1}{3}\pi, \theta_2 = -\frac{2}{3}\pi, \theta_3 = \frac{1}{3}\pi, \theta_4 = -\frac{2}{3}\pi,$ $\theta_5 = \frac{1}{3}\pi, \theta_6 = -\frac{2}{3}\pi, \theta_7 = \frac{1}{3}\pi, \theta_8 = -\frac{2}{3}\pi,$ $\theta_9 = \frac{1}{3}\pi, \theta_{10} = -\frac{2}{3}\pi, \theta_{11} = \frac{1}{3}\pi, \theta_{12} = -\frac{2}{3}\pi$
	$\theta_1 = \frac{1}{3}\pi, \theta_2 = -\frac{2}{3}\pi, \theta_3 = \frac{1}{3}\pi, \theta_4 = \frac{1}{3}\pi,$ $\theta_5 = \frac{1}{3}\pi, \theta_6 = -\frac{2}{3}\pi, \theta_7 = \frac{1}{3}\pi, \theta_8 = -\frac{2}{3}\pi,$ $\theta_9 = \frac{1}{3}\pi, \theta_{10} = -\frac{2}{3}\pi, \theta_{11} = \frac{1}{3}\pi,$ $\theta_{12} = \frac{1}{3}\pi, \theta_{13} = \frac{1}{3}\pi, \theta_{14} = -\frac{2}{3}\pi,$ $\theta_{15} = \frac{1}{3}\pi, \theta_{16} = -\frac{2}{3}\pi, \theta_{17} = \frac{1}{3}\pi,$ $\theta_{18} = -\frac{2}{3}\pi$

表 2-1 Koch 曲線第一層到第三層角度表

經由表 2-1 格觀察可以得到，第二層的 θ_4 等於第一層 θ_1 ，同理第二層的 θ_8 等於第一層 θ_2 ，經由這樣的順序，我們可以知道第三層 θ_4 等於第二層的 θ_1 ，綜合上述的邏輯我們可以整理出一個表 2-2。

第一層							θ_1							θ_2									
第二層		θ_1		θ_2		θ_3		θ_4		θ_5		θ_6		θ_7		θ_8		θ_9		θ_{10}		θ_{11}	
第三層	[A]	θ_4	[A]	θ_8	[A]	θ_{12}	[A]	θ_{16}	[A]	θ_{20}	[A]	θ_{24}	[A]	θ_{28}	[A]	θ_{32}	[A]	θ_{36}	[A]	θ_{40}	[A]	θ_{44}	[A]

表 2-2 Koch 曲線角度整理圖

其中 $[A] = [\frac{1}{3}\pi, -\frac{2}{3}\pi, \frac{1}{3}\pi]$ ，經由表 2-2 的規律我們可以判斷出，除了每四個一個循環之外，下一層的 4 的倍數為上一層的角度，因此當我們需要作出更高層次的圖形時，需要作一些 4 的次方數的檢驗來作一些轉換，以上表第三層的 θ_{16} 來說，我們可以利用 mod 檢驗餘數的方式來檢驗規律，例如， $\text{mod}(16,4) = 0$ ，代表 16 除以 4 所得到的餘數為 0， $\text{mod}(\frac{16}{4}, 4) = 0$ ，代表序數 16 先除以 4 之後，再取餘數，目的是要檢驗是否在轉角處符合上一層的規律，經檢驗後發現還是跟第一次一樣，於是我們再取 $\text{mod}(\frac{16}{4^2}, 4) = 1$ ，代表序數 16 先除以 4^2 之後再除以 4，此時我們發現取出來的餘數變成 1，表示 16 所代表的角色相當於第一層的 θ_1 ，如此不停的旋轉與檢驗，經由 $z_{k+1} = z_k + (z_k - z_{k-1}) \cdot e^{i\theta}$ 不斷疊代，可以產生雪花圖形。

以雪花曲線延展五層為例，在程式的撰寫上，第一部為先定義雪花曲線的發展為五層 $m := 5$ ，接下來定義第一個點 $z_0 := 0$ ，由 z_0 開始出發，接下來因為 z_1 的長度會因為層數越多，而使改變一開始的長度越短，每多一層則長度會變成原來的 $\frac{1}{3}$ ，若是 m 層，則初始 z_1 長度為

$\left(\frac{1}{3}\right)^m$ ，因此定義 z_1 的為 $\left(\frac{1}{3}\right)^m \cdot e^{i \cdot \frac{\pi}{2}}$ ，剩下的部分可以用類似複製的方式，將序數延伸，即可以產生其他段，因此定義另外兩段的旋轉處之角度為 $\theta_{4^m} = -\frac{2}{3}\pi$ ， $\theta_{2 \cdot 4^m} = -\frac{2}{3}\pi$ 。

定義 $k := 1..4^m - 1$ ，讓 k 可以依序數的不同，調整旋轉的次數，接下來利用 mod 內定的參數，取出每一次的餘數，例如： $\text{mod}(a, b)$ 代表 a 除以 b 所得到的餘數，由此可以決定當餘數是1時旋轉 $\frac{1}{3}\pi$ ，餘數是2時旋轉 $-\frac{2}{3}\pi$ ，餘數是3時旋轉 $\frac{1}{3}\pi$ ，因此定義

$$\theta_k = (\text{mod}(k, 4) - 1) \cdot (-\pi) + \frac{1}{3}\pi \quad (2.9)$$

又當餘數等於0時，則用 4^m 來做旋轉度的判斷， $m = 1, 2, 3, 4, 5$ ，來決定旋轉的角度。此外在程式必須做出邏輯的判斷，所以必須要到內建於 Programming 中的邏輯參數 `if`，並利用 `if` 參數做出相關的判斷，例如：

$$\theta_k := \text{if} \left[\text{mod}(k, 4) = 0, (\text{mod}(k, 4) - 1) \cdot (-\pi) + \frac{1}{3}\pi, \theta_k \right] \quad (2.10)$$

表示 θ_k 在 `if` 參數的判斷下，若滿足 $\text{mod}(k, 4) = 0$ ，就輸出 $(\text{mod}(k, 4) - 1) \cdot (-\pi) + \frac{1}{3}\pi$ ，否則就輸出原來的 θ_k ；因為這樣的判斷模式下，可以決定每四個序數一次的額外旋轉，當在高層次的圖形時，隨著 m 的增加，必須在 4^m 的序數上做出旋轉的相關判斷；又 `mathcad` 在判斷程式時會由上而下辨識，並下方的程式碼會覆蓋上面的程式定義，所以在定義完一般化的 θ_k 之後，可以再利用 `if` 參數，依序判斷其他狀況，就可以判斷何時需要旋轉，相關程式碼如附錄 2.2。

2.3 Julia sets & Manderbrot sets

在製作 fractal 圖形中，除了上述利用分層方法進行疊代產生碎形圖形之外，還有兩種特殊的集合，利用集合中的數的位置，來產生美麗的碎形圖案。在複數系統中，複數 $z = x + iy$ ，其中 x, y 為實數，經由不同的方程式的疊代，可以產生一些特殊的圖形，Julia sets 與 Mandelbrot sets 為當中利用複數疊代方式產生的特殊圖形，也是較為人所稱道圖形，這兩個圖形都是經由複數不斷的疊代，產生許多自我相似的圖形。

2.3.1 Julia sets

在複雜的動態背景下，Julia 集為一個經由反覆疊代同樣屬行的函數，所產生的一個集合，在微小的變動之下，可以呈現出許多巨大的變化，因此吸引許多人想要了解裡面的結構。

Julia sets是由一個二次映射 $z_{n+1} = z_n^2 + C$ ，其中 C 為一個固定值，而幾乎所有的 C 都會得到一個映射的結果，並且將所有代入映射條件中，經反覆疊代之後，所有不發散的點之邊界，稱之為Julia sets，他的圖形具有碎形的相關性質[6]。

對於 $z = x + iy$ 來說， x 部分為 z 的實部， y 部分為 z 的虛部，若是 $C = a + bi$ ，我們用數對 (a, b) 表示，則對於 $z \rightarrow z^2 + C$ 來說，

$$x \rightarrow x^2 - y^2 + a \quad (2.11)$$

$$y \rightarrow 2 \cdot x \cdot y + b \quad (2.12)$$

因此複數 z 會因為對應到一個固定的 C 之下 $z^2 + C$ ，經由不停的疊代會產生許多不同的結果，而Julia sets就是找出所有的 z 在一定的疊代次數下，所有可以收斂的 z 的點。

其中我們在執行疊代的時候，首先我們要先定義 z 在一個區間中，

經由搜尋所有區間中的 z 值並帶入執行疊代，最後蒐集所有經由 $z \rightarrow z^2 + C$ 疊代一定次數後，還不會發散的點，而這些點，所形成的集合，就是Julia sets。所以在執行Julia sets的程式疊代之前，我們要先決定三個問題，第一我們所要執行的區塊在哪裡？，第二我們所要執行疊代到底運算到多大時算是發散？以及第三疊代的次數與運算時間是多少？其中，尤其在疊代次數的多寡尤為重要，除了會影響計算時間的長短之外，同時也會影像出現的圖形。

接下來我們要先學會找尋區塊內的點的方法，在定義區塊時，我們會先將一個區塊切割成許多小的區塊，並將我們想要搜尋的 z 定義在區塊的中間，如圖2-13所示

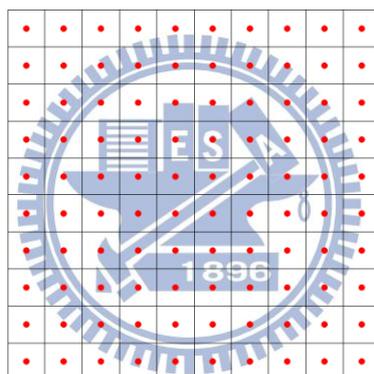


圖2-13 搜尋區塊示意圖

並將每一格紅點 z ，代入 $z \rightarrow z^2 + C$ ，其中切割的格子數的多寡，也會影響最後圖形出現的解析度，將區塊內的格子切割的數量越多，代表圖形可以呈現的解析度越高，反之，則解析度越低，若解析度太低，在微觀之中，看不到特別的變化，但是有時追求太高的解析度，容易使程式執行運算的時間過長。

其次，在執行運算的過程中，通常只要 $|z_n| > 1$ ，就會發現當這些 z_n 值代入 $z \rightarrow z^2 + C$ 中，就會以很快的速度發散，因此我們會定義一個臨界發散值，一旦超過這個值，就不再執行疊代，但是有時候會

因為 C 值的正負或是疊代的次數，影響到一些臨界的地方的邊界點，倘若定義發散值太大，其結果也會隨著最後疊代的次數越大而趨向發散，定義的太小會使運算時間拉長，因此在這個程式中，我們將會定義當 $|z_n| > 2$ 時，由mathcad軟體執行判定為發散不再執行疊代。

第一點的解析度與第二點的發散大小決定好了之後，最後我們要確定執行的時間，因為mathcad為一種計算軟體，透過數值的運算，決定留下最後可以不被發散的點，當我們要在短時間之內得到一個Julia sets的圖形，我們要將必須將疊代次數減少，或是解析度要變少，否則隨著疊代的次數越多，解析度越大，將會使圖形產生的過程會越來越久。

首先定義我們要將圖形切成 $m \times m$ 格，切割完成之後我們再取每一小格內部的點，在選取的過程中，由於計算時間上的考量，在計算二維空間較為費時，於是我們將原有的二維平面上的一個一個區塊，轉換成一為空間上的區塊，以將原有平面切割成 10×10 ，100份的二維平面轉換為一維空間，如圖2-14所示，

(1,10)	(2,10)	(3,10)	(4,10)	(5,10)	(6,10)	(7,10)	(8,10)	(9,10)	(10,10)
(1,9)	(2,9)	(3,9)	(4,9)	(5,9)	(6,9)	(7,9)	(8,9)	(9,9)	(10,9)
(1,8)	(2,8)	(3,8)	(4,8)	(5,8)	(6,8)	(7,8)	(8,8)	(9,8)	(10,8)
(1,7)	(2,7)	(3,7)	(4,7)	(5,7)	(6,7)	(7,7)	(8,7)	(9,7)	(10,7)
(1,6)	(2,6)	(3,6)	(4,6)	(5,6)	(6,6)	(7,6)	(8,6)	(9,6)	(10,6)
(1,5)	(2,5)	(3,5)	(4,5)	(5,5)	(6,5)	(7,5)	(8,5)	(9,5)	(10,5)
(1,4)	(2,4)	(3,4)	(4,4)	(5,4)	(6,4)	(7,4)	(8,4)	(9,4)	(10,4)
(1,3)	(2,3)	(3,3)	(4,3)	(5,3)	(6,3)	(7,3)	(8,3)	(9,3)	(10,3)
(1,2)	(2,2)	(3,2)	(4,2)	(5,2)	(6,2)	(7,2)	(8,2)	(9,2)	(10,2)
(1,1)	(2,1)	(3,1)	(4,1)	(5,1)	(6,1)	(7,1)	(8,1)	(9,1)	(10,1)

圖2-14 區塊分割示意圖

將區塊分別編碼，如圖2-15

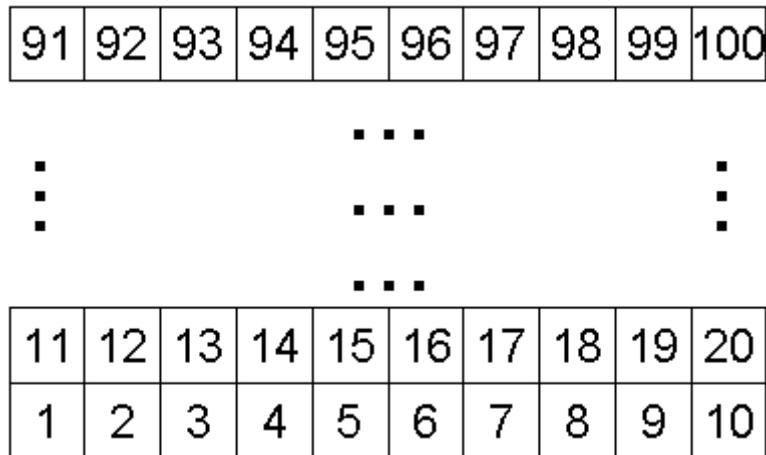


圖2-15 區塊編碼示意圖

轉成圖2-16



圖2-16 二維轉一維示意圖

在轉換的過程中，利用mathcad軟體中的參數floor與mod，其中floor參數的意義相當於取高斯數，舉例來說， $\text{floor}(3.6) = 3$ ， $\text{floor}(0.6) = 0$ ，而mod則是取餘數的方式，藉由floor來決定y方向的值，由mod決定x方向的值，以上例來說，若是編號23號，則 $\text{mod}(23,10) = 3$ ，則知道相當於 $x = 3$ 的位置，又 $\text{floor}\left(\frac{23}{10}\right) + 1 = 3$ ，得到相當於 $y = 3$ ，如此在運算上可以將原來的二維平面上的位置轉成一維的直線來運算，在時間花費上可以節省許多。

在分割完平面之後，將平面上的每一個區塊選擇某一點代入方程式中，若是都取該區間的中間位置，會使圖形感覺較為制式化，如圖2-17所示，

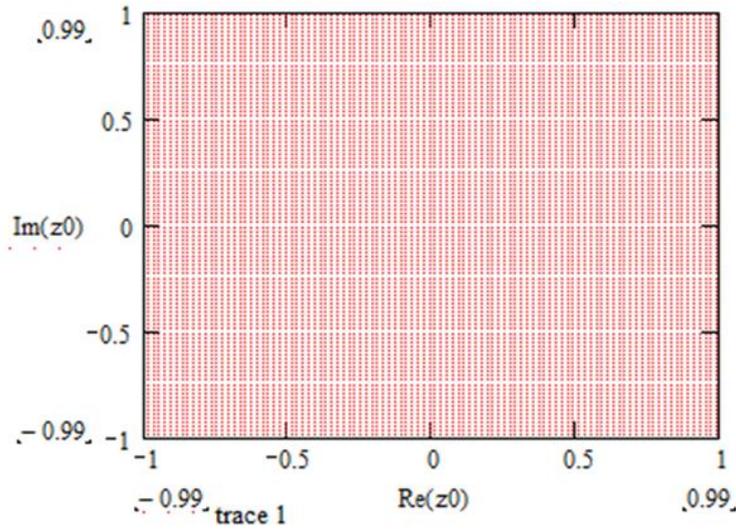


圖2-17 [-1,1]平面切成 100×100 格圖

於是我們可以設定一些隨機性，使得圖形的變化更為豐富，以圖2-18為例，加入一些隨機性

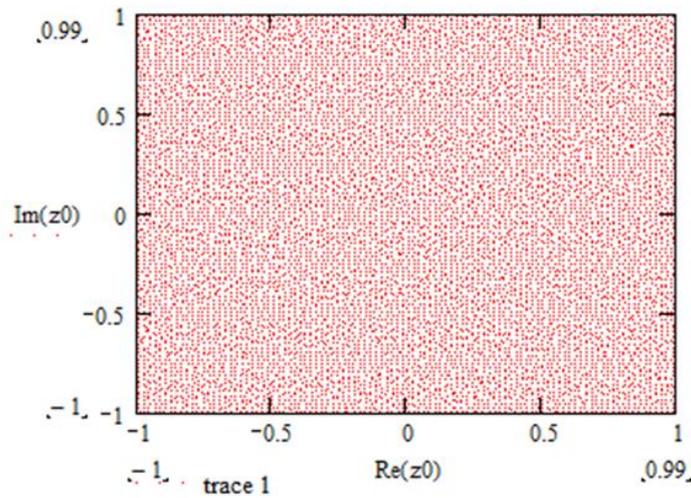


圖2-18 [-1,1]平面切成 100×100 格隨機灑點圖

在定義完區間中取格子的方式之後，我們給予格子內的點 $z0_i$ 實部 x ，虛部 y 區間 $[-a, a]$ 中，將其平面各切 M 個點，並利用隨機性，將點 $z0_i$ 不要都只有產生在該區塊的中央位置，讓 i 的個數一共有 M^2 個， $z0_i$ 的方程式如下所示，

$$z0_i = \left(\text{floor} \left(\frac{i}{M} \right) + \frac{\text{rnd}(1)}{2} \right) \cdot \frac{2a}{M} - a + i \cdot \left[\frac{\left[\text{mod}(i, M) + \frac{\text{rnd}(1)}{2} \right]}{M} \cdot 2a - a \right] \quad (2.13)$$

定義完 $z0_i$ 之後，我們可以在給定 C 值與區間範圍之內，先將該區間範圍切成 $M = 600$ 份，產生 600×600 格，成為我們所要產生圖形的解析度，接著因為程式上需要設定疊代的次數和並利用迴圈方式去計算，因此先利用「add program line」產生我所需要運算的迴圈，在此我們設定疊代次數為200次，並且定義發散值 $P = 2$ ，根據方程式即可畫出Julia sets圖形，相關程式碼如附錄2.3。

經由不同的 C 值帶入，也會得到不同的結果，分別以 C 等於 $-0.777 + 0.11i$ 如圖2-19(a)， $-0.11 + 0.777i$ 如圖2-19(b)， $-0.4 + 0.3i$ 如圖2-19(c)， $0.3 - 0.4i$ 如圖2-19(d)。

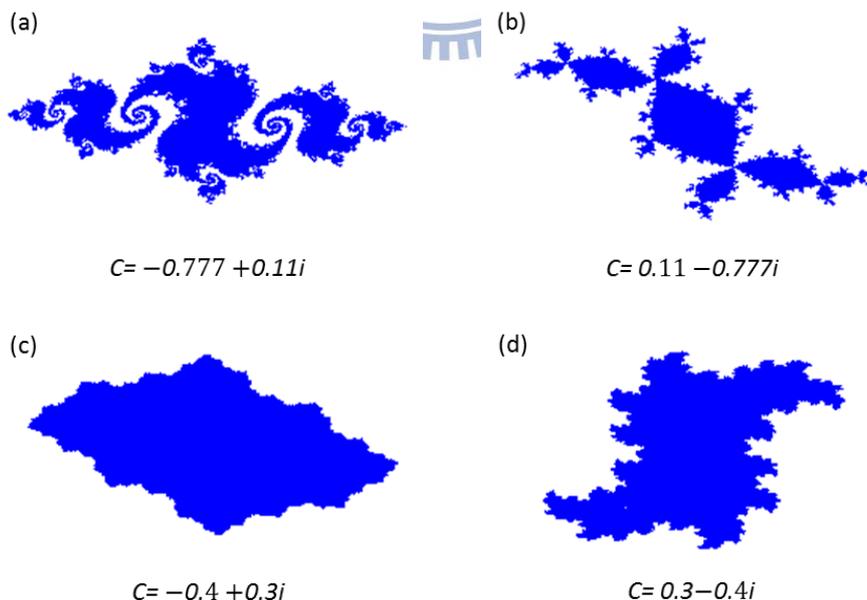


圖2-19 不同 C 值 Julia sets圖形

在上例中，Julia sets 所留下的點都是可以順利通過200次的疊代，

但是在試驗的過程中會發現到，我們所設立的疊代次數是指經過多少次數的疊代之後，所留下來不會發散的點，因此隨著疊代次數的調整，圖形也會產生許多不同的變化，以下我們分別由不同的疊代次數的圖形做一些比較：

疊代50次：

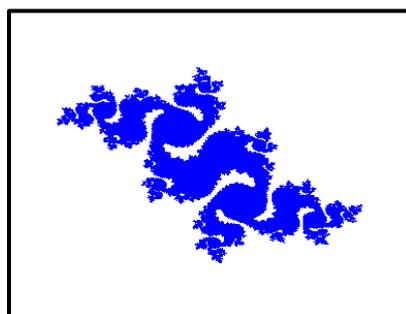


圖2-20 Julia sets 疊代50次示意圖

疊代100次：

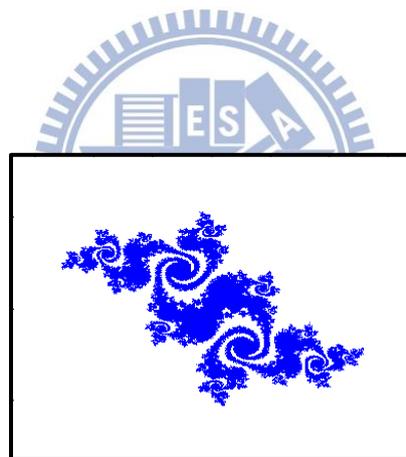


圖2-21 Julia sets 疊代100次示意圖

疊代500次：

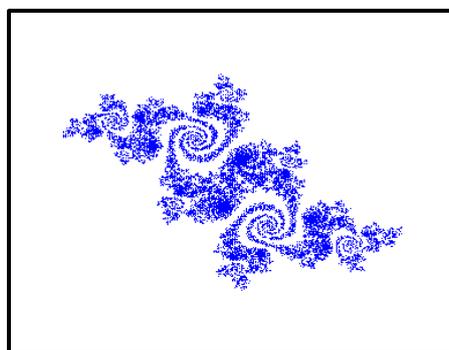


圖2-22 Julia sets 疊代500次示意圖

疊代1000次：

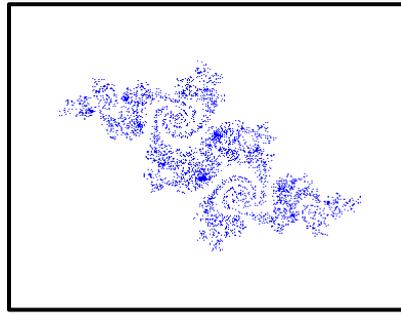


圖2-23 Julia sets 疊代500次示意圖

由圖2-20到圖2-23可知，有時候疊代的次數太多，反而會造成點數太少不易辨識圖形的反效果，因此要選擇各個圖形合適的疊代次數是很重要的，但是到底要如何拿捏疊代的次數，也是一個需要去掌握的問題，在後面的小節會介紹出不管疊代多少次數，都可以一樣清晰的繪圖方法。



2.3.2 Mandelbrot sets

除了著名的 fractal 圖形中，除了 Julia sets 之外還有一個就是 Mandelbrot sets，其中 Mandelbrot sets 的產生過程和 Julia sets 極為相似，但兩者取決疊代的方式卻不同。在產生 Mandelbrot sets 圖形時，疊代的方程式和 Julia sets 相同都是 $z \rightarrow z^2 + C$ 的二次映射，和 Julia sets 不同的是，Julia sets 是固定 C 的情況下去尋找不會發散的 z ，而 Mandelbrot sets 則是固定 $z = 0$ 的情況下，將所有該區間範圍中的值一一代入 $z \rightarrow z^2 + C$ 的 C 中，經過反覆疊代之後，所有不會發散的點集合，就是 Mandelbrot sets[6]。

經由之前所學的 Julia sets 的方法，我們可以先將想要繪製的區間加以切割，其中在實數部分範圍為 $[-a, a]$ ，虛數部分範圍為 $[-b, b]$ ，將這兩個範圍切割成 M 份之後，定義出 C_i

$$C_i = \left(\text{floor} \left(\frac{i}{M} \right) + \frac{\text{rnd}(1)}{2} \right) \cdot \frac{2a}{M} - a + i \cdot \left[\frac{\left[\text{mod}(i, M) + \frac{\text{rnd}(1)}{2} \right]}{M} \cdot 2b - b \right] \quad (2.14)$$

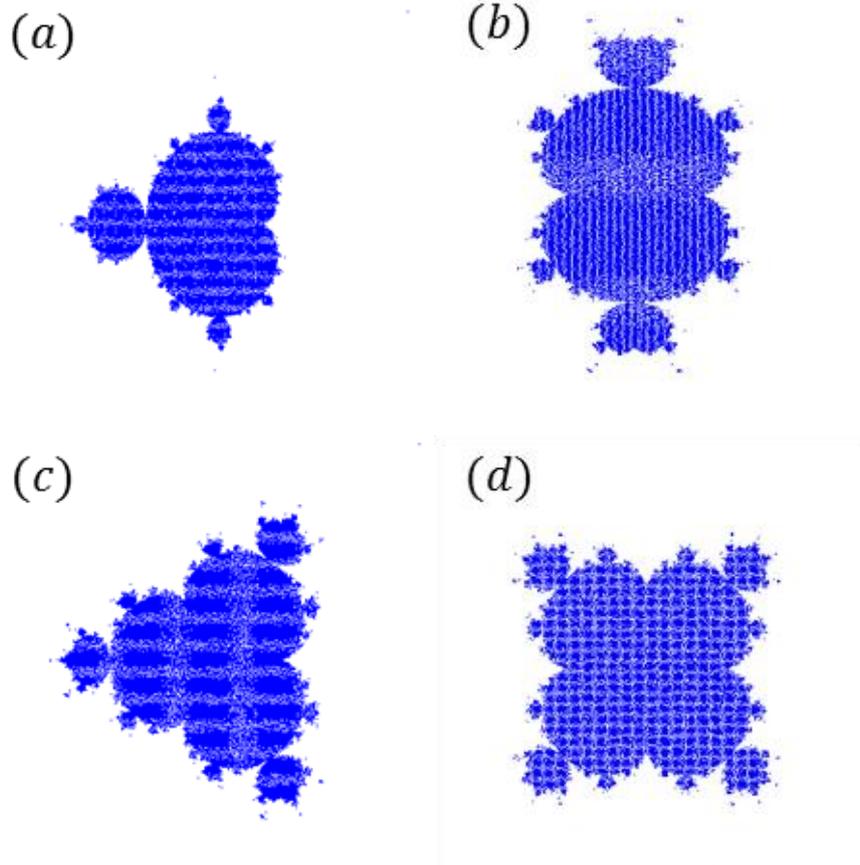
經由決定疊代次數 N 次與發散值 P 之後，即可繪製出 Mandelbrot sets 的圖形，但在繪製的過程中會發現到 Mandelbrot sets 和 Julia sets 雖有相同的疊代過程，但是圖形的結果卻是大相逕庭，因此在繪製的過程中，需多注意顯示圖形的邊界，所以我們可以利用加入 Δa 與 Δb 的方式，並且讓繪圖的邊界改為 $[-a + \Delta a, a + \Delta a]$ 與 $[-b + \Delta b, b + \Delta b]$ ，讓圖形有置中的效果，

$$C_i = \left(\text{floor} \left(\frac{i}{M} \right) + \frac{\text{rnd}(1)}{2} \right) \cdot \frac{2a}{M} - a + \Delta a + i$$

$$\cdot \left[\frac{\left[\text{mod}(i, M) + \frac{\text{rnd}(1)}{2} \right]}{M} \cdot 2b - b + \Delta b \right] \quad (2.15)$$

讓圖形可以變得更集中於我們所要觀察的地方，相關程式碼如附錄 2.4。

也因為Mandelbrot sets和Julia sets有相似的疊代過程，我們也可以利用修改方程式的次方的方式，如圖2-24中，繪製了2次方到7次方的圖形。



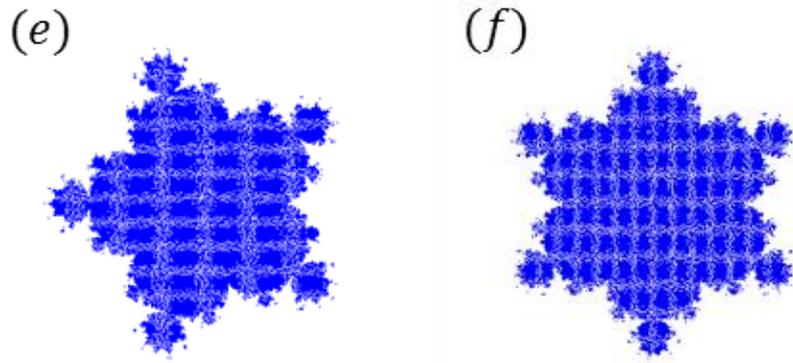
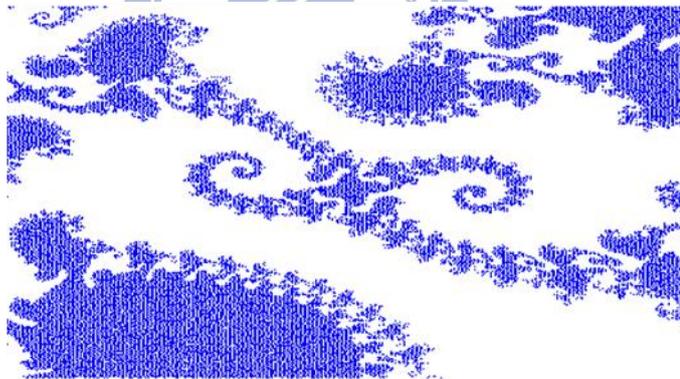


圖2-24 Mandelbrot sets 次方的變化圖

除了圖2-24的變化之外，我們也可以深入的觀察Mandelbrot sets 在某些點的結構與Julia sets相似[5]，分別以 $C = -0.745429 + 0.113008i$ 為中心，範圍0.00007的Mandelbrot sets 放大圖，如圖2-25(a) 與 $z = -0.745429 + 0.113008i$ 為中心，範圍為0.00049的Julia sets放大圖，如圖2-25(b)。

(a)



(b)



圖2-25 (a)Mandelbrot sets (b) Julia sets相似圖形

2.4 等值線繪圖

在繪製上述的兩種集合時，常常會發現因為在 mathcad 軟體的限制下，在一維空間中沒有辦法繪出一些顏色豐富的圖形，因此我們可以給予圖形一些高度的變化，並且利用等高線圖形或是曲面圖形繪圖的方式，讓圖形有更多不同的顏色變化，同時也可以增加許多邊界的可視性。

因為疊代的次數會影響點是否屬於集合之中，所以疊代次數越高而會使得圖形上的點數越來越少，難免會產生遺珠之憾，為了避免這些情形產生，我們可以利用在每一個點的位置上加上在第幾次才發散的狀況，來做為該點的高度。

舉例來說，我們可以利用該點在疊代時是第幾次被否決的次數來決定該點的高度，假設我們將每個點都疊代 700 次，若可以疊代 700 次依然在集合的點會獲得 700 的高度，若是疊代第 100 次就發散的點，將會得到高度 100，如此我們就可以經由高度的變化給予顏色或是亮暗的差異，讓圖形產生更多改變 如圖 2-26，2-27 所示，相關程式碼如附錄 2.5。

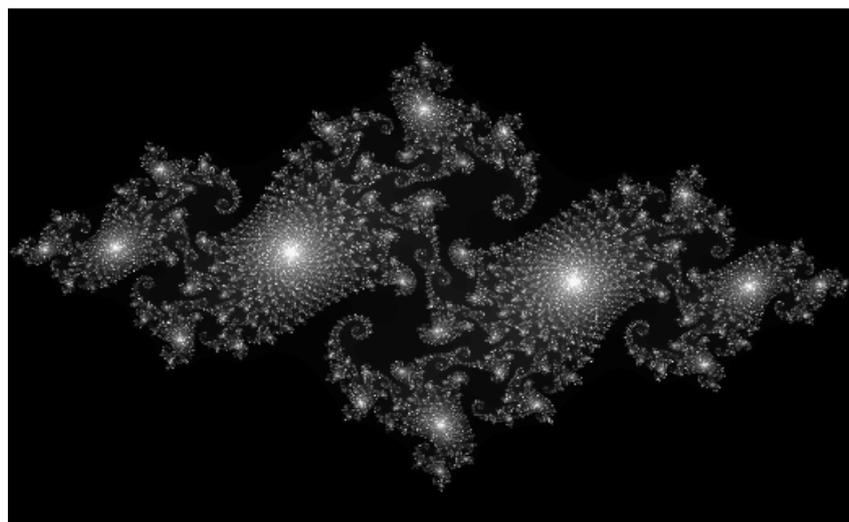


圖 2-26 Julia set 等值線圖

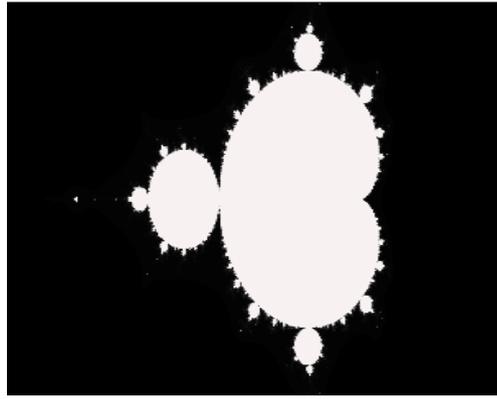


圖 2-27 Mandelbrot set 等值線圖

此外，經由清晰化的步驟，我們可以經由不斷放大局部圖形來觀察細部的圖形變化，如下圖，以(0,0)為圖形的中心點，以 $C = -0.745429 + 0.113008i$ 依不同的放大比例的範圍繪製 Julia sets 圖形，分別以範圍 0.5 如圖 2-28(a)，範圍 0.05 如圖 2-28(b)，範圍 0.005 如圖 2-28(c)，範圍 0.0005 如圖 2-28(d)，

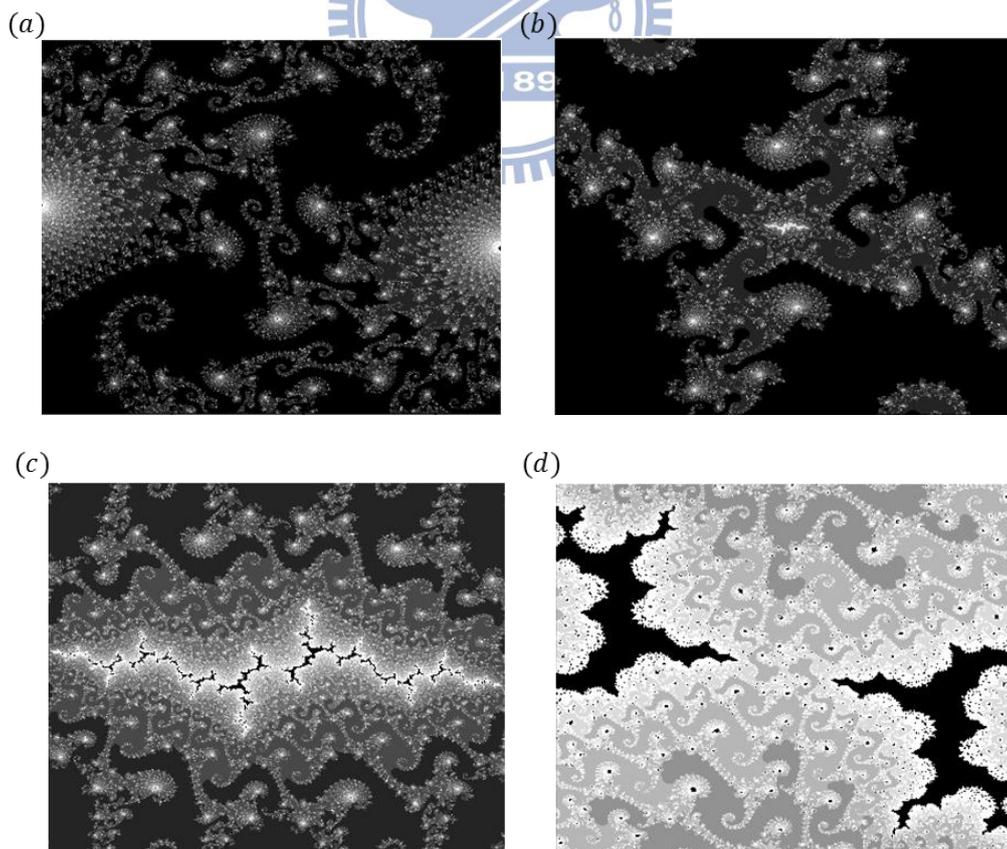


圖 2-28 Julia sets $c = -0.745429 + 0.113008i$ 不同範圍等值線圖

2.5 經由 IFS 的方式產生碎形圖騰

在繪製碎形圖案中，還有另外一種方法，可以產生碎形圖案。在數學上，疊代函數系統(iterated function systems 簡稱 IFS)，製作出來的圖形通常具有碎形圖形的自我相似性，因此通常用來繪製 fractal 圖形[5]。而這些製作出來的圖形在二維的計算與繪製中，可以是任何尺寸的。其中最典型的例子就是史賓斯基三角形(Sierpinski gasket)，經由不斷的收縮，緊密的結合之所產生；而 IFS 碎形形狀是由幾個自身的形狀，在自身的形狀內也有相同的形體，反覆循環，產生自我相似性的碎形圖案。

2.5.1 IFS 概念

IFS 起先由 1981 年的 John E. Hutchinson 所提出，並在 Michael Barnsley 的<Fractals Everywhere>[7]書中被大量的推廣運用；在自然界中，許多的樹葉與蕨類植物的形狀，所呈現出來的結構往往都具有自我相似性[8]，但也因為隨機性的變化之下，往往找不到兩個蕨類植物的在葉狀體的分支與規模較小的葉片之中形體是完全一樣的，但是經由 IFS 的不斷疊代下，可以助於幾何建模[9]。

在數學的定義上則為 $\{f_i: X \rightarrow X | i = 1, 2, 3, \dots, N, N \in \mathbb{N}\}$ ，是一個在 complete metric space 上的 contraction mappings[10]。

在幾何學上，利用仿射變換(affine transformation)，改變原有二維空間上的角度與長度，利用 IFS 疊代，產生許多自我相似圖形。

利用仿射變換將原來的座標軸做映射進而產生新的座標，將原有的位置做一個長度、角度和位移的變化，

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = W \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (2.16)$$

其中， $a = r\cos(\theta)$ ， $b = r\sin(\theta)$ ， $c = -s\sin(\phi)$ ， $d = s\cos(\phi)$ ，係數 r 代表 x 方向的縮放程度， s 代表 y 方向的縮放程度， θ 和 ϕ 則分別代表 x 和 y 方向的旋轉角度， e 和 f 則為 x 和 y 方向的位移值。

相關的變化如下圖 2-29 所示[9]：

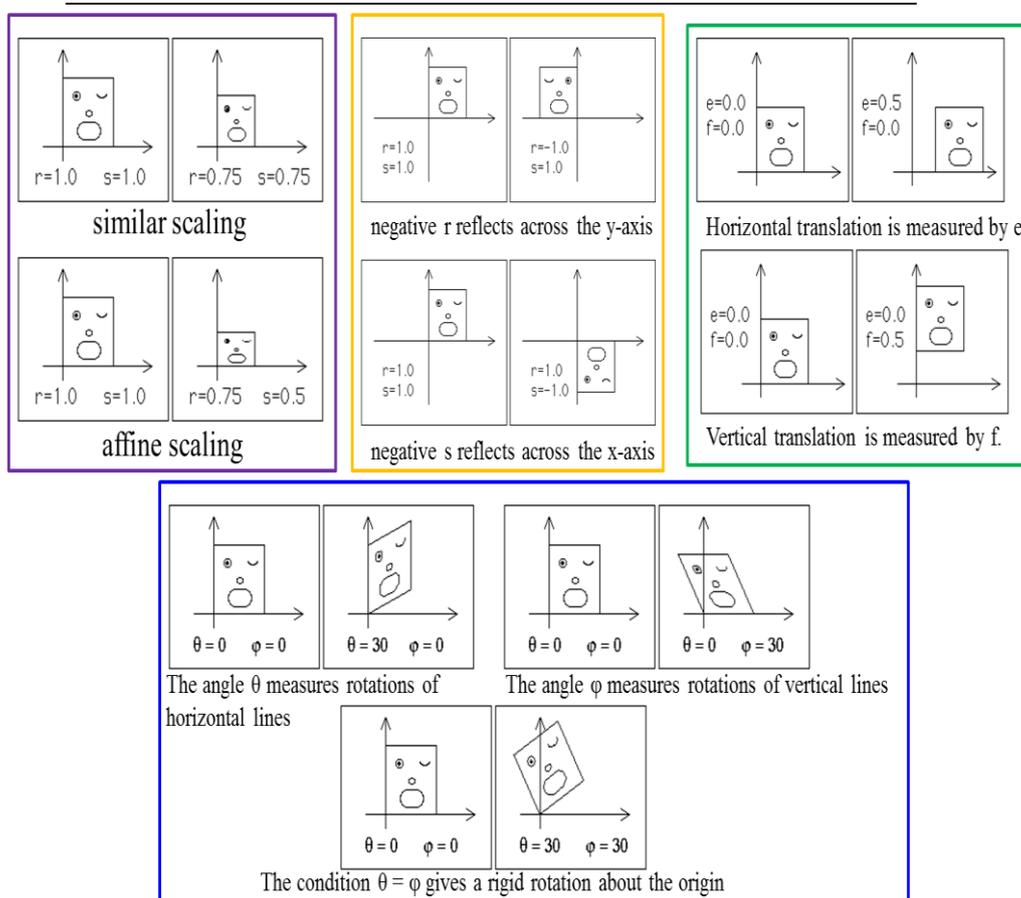


圖 2-29 仿射變換相關變化圖

為了防止不斷疊代的過程中，產生發散的圖形現象，因此在仿射變換中，有下列限制，

$$w: (x, y) \rightarrow (x', y')$$

$$\begin{cases} x' = ax + by + e \\ y' = cx + dy + f \end{cases}$$

$$s = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$s' = \sqrt{(x'_1 - x'_2)^2 + (y'_1 - y'_2)^2}$$

$$= \sqrt{[a(x_1 - x_2)]^2 - [b(y_1 - y_2)]^2 + [c(x_1 - x_2) + d(y_1 - y_2)]^2}$$

$$s > s'$$

$$\Rightarrow (x_1 - x_2)^2 - (y_1 - y_2)^2$$

$$> (a^2 + c^2)(x_1 - x_2)^2 + (b^2 + d^2)(y_1 - y_2)^2 + 2(ab + cd)(x_1 - x_2)(y_1 - y_2)$$

若是此仿射變換是可推定的則基本範圍為

$$\begin{cases} (a^2 + c^2) < 1 \\ (b^2 + d^2) < 1 \end{cases}$$

於是，我們利用一些平面的幾何轉換，如：縮放、旋轉、位移來改變原來的平面位置，如此若運用於點與點之間的關係，藉由疊代的方式，即可產生相關的碎形圖形。在 IFS 圖形產生的順序過程中，我們會以先縮放、在旋轉、後偏移的方式產生下一個疊代的位置，如此，我們可以利用一個矩陣關係，來建立點和點的關係，下面以

r, s, θ , ϕ , e, f 產生一個表格，如表 2-3

r	s	θ	ϕ	e	f

表 2-3 IFS 仿射變化表

在之後的製作碎形過程中，我們將利用上面的表格來繪製相關的碎形圖形。

2.5.2 貼片

碎形圖形是一種較為特殊的幾何圖形，他通常都會依變異程度的大小具有不同的自我相似性質，而經由不斷的反覆疊代過程中，碎形圖形的某些部份都會產生相似的特徵圖形，或相似結構的圖案。以數學的角度來看，若經過不斷的反覆疊代產生的一個收縮性的函數極限

點集，我們會稱之為吸子(attractor)[5]，例如：以 $f(x) = -x$ 來說， $x_{n+1} = -x$ ， $x_0 = 1$ 經由不斷的反覆疊代，最後會發現這個圖形只有在兩點跳動，產生兩個吸子；又 $f(x) = \frac{x}{2}$ 為例， $x_{n+1} = \frac{x}{2}$ ， $x_0 = 1$ ，經過不斷的反覆疊代，會發現到 $f(x)$ 的值將會不斷地朝 0 趨近，而 0 就是 $f(x) = \frac{x}{2}$ 的吸子。其中，吸子可以為點、有限集合、曲線、或是一種複印本。

因為碎形圖形的特殊性，在圖形中可以觀察到許多的規律性，但是這樣的吸子結構較為複雜，我們稱呼在碎形中的吸子為奇異吸子(strange attractor)，因為經由這樣的吸子，可以產生相同的結構的圖形，無窮無盡的延伸，不斷的產生相同特徵的圖案，而貼片理論的產生，目的就是藉由貼片希望產生一些奇異吸子的方向，進而產生不斷延伸的碎形圖形。

在數學中，貼片理論是將我們之前提到的 IFS 疊代函數系統中的吸子接近給定的集合中，經由貼片的方法所產生的碎形圖形，將會形似於初始生成子圖形。

以 Koch 雪花曲線為例，若是用貼片表示，如圖 2-30[5]，

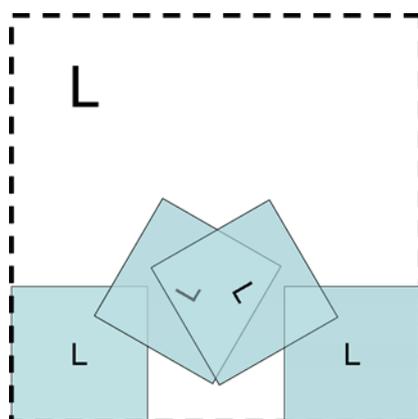


圖 2-30 Koch 雪花圖形貼片示意圖

並利用 mathacad 軟體中的內建參數 rnd，建立隨機性，讓碎形的撒點狀況可以因為隨機性使得圖形變得更均勻的分布於畫面中，其中 $rnd(1)$ 代表由 $[0,1]$ 之間隨機選出數字，加上參數 Φ ，其中 $\Phi(x - a)$ 代表若是 $x > a$ 則值為 1，反之則為 0，所以當用兩組參數 $\Phi(x - a_1)\Phi(a_2 - x)$ ，則代表當 $a_1 < x < a_2$ 時， $\Phi(x - a_1)\Phi(a_2 - x) = 1$ ，反之則為 0，綜合上述兩點可知道，在四個貼片中，若定義 $d_n = rnd(1)$ 安排由 $[0,1]$ 中取出隨機數字，並利用參數 $\Phi\left(d_n - \frac{0}{4}\right)\Phi\left(\frac{1}{4} - d_n\right)$ 、 $\Phi\left(d_n - \frac{1}{4}\right)\Phi\left(\frac{2}{4} - d_n\right)$ 、 $\Phi\left(d_n - \frac{2}{4}\right)\Phi\left(\frac{3}{4} - d_n\right)$ 、 $\Phi\left(d_n - \frac{3}{4}\right)$ ，當乘積的結果為 1 時才會出現該點，使點分別安排於四個貼片的參數之中，如此，圖形便可以變得均勻分布於四個貼片之中。

由貼片的形狀可以得知，在單位為 1 的原始圖形中，由左至右，其中第一個貼片縮小 $\frac{1}{3}$ 後置於原點，第二個貼片縮小 $\frac{1}{3}$ 後，並旋轉 $\frac{\pi}{3}$ ，再將圖形平移至 $\left(\frac{1}{3}, 0\right)$ 的位置，第三個貼片縮小 $\frac{1}{3}$ 後，旋轉 $-\frac{\pi}{3}$ ，並平移至 $\left(\frac{1}{2}, \frac{\sqrt{3}}{6}\right)$ 的位置，第四個貼片則縮小 $\frac{1}{3}$ 後，並平移至 $\left(\frac{2}{3}, 0\right)$ 的位置，如表 2-4

r	S	θ	Φ	e	F
$\frac{1}{3}$	$\frac{1}{3}$	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{\pi}{3}$	$\frac{\pi}{3}$	$\frac{1}{3}$	0
$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{\pi}{3}$	$-\frac{\pi}{3}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{6}$
$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{2}{3}$	0

表 2-4 利用貼片參數形成 Koch 雪花曲線表

將上述的值輸入程式之後，決定灑入的點數，即可以得到相關的結果，相關程式碼如附錄 2.6。

灑 2000 點時：

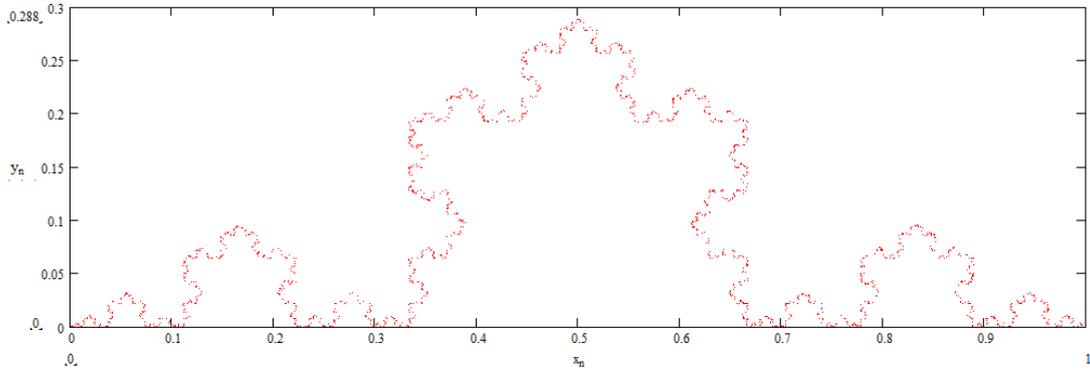


圖 2-31 利用貼片灑 2000 點形成 Koch 曲線

灑 10000 點時：

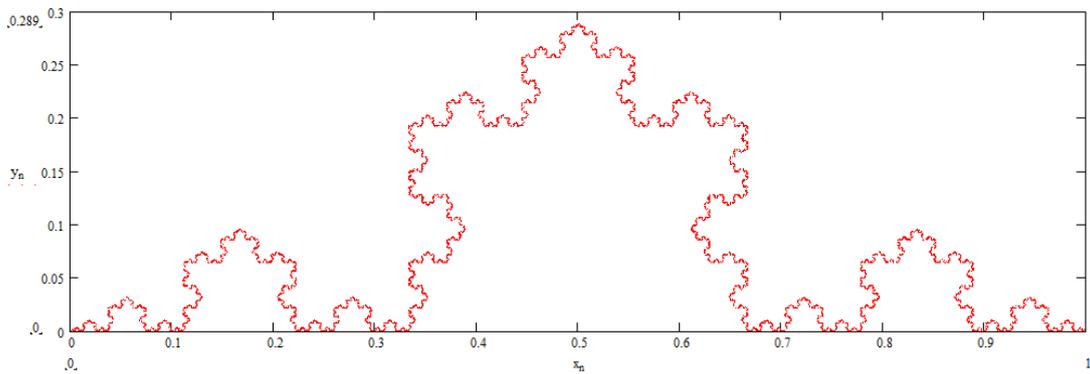


圖 2-32 利用貼片灑 10000 點形成 Koch 曲線

灑 50000 點時：

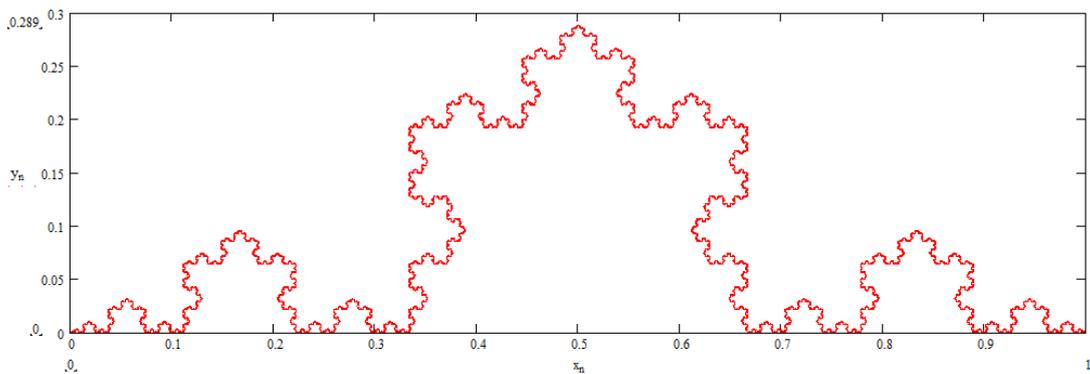


圖 2-33 利用貼片灑 50000 點形成 Koch 曲線

灑 100000 點時：

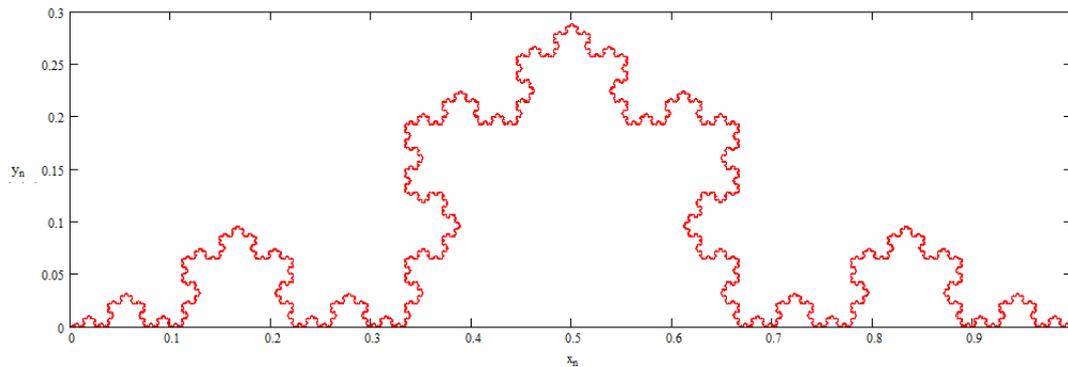


圖 2-34 利用貼片灑 100000 點形成 Koch 曲線

由上述的圖形可以發現到，當灑的點數太少時，影像不夠清晰，但當灑的點數夠多時，即可產生足夠清晰的 Koch 雪花曲線

在此提供額外的一些圖形範例，以珊瑚圖形來說，並建立 r , s , θ , ϕ , e , f 表，如表 2-5，最後產生 IFS 圖形，則結果為圖 2-35。

r	s	θ	Φ	e	f
0.6	0.6	0	0	0.12	0
0.5	0.5	$\frac{\pi}{4}$	$\frac{\pi}{4}$	0	0.25
0.5	0.5	$-\frac{\pi}{4}$	$-\frac{\pi}{4}$	0.5	0.6

表 2-5 IFS 生成珊瑚參數表

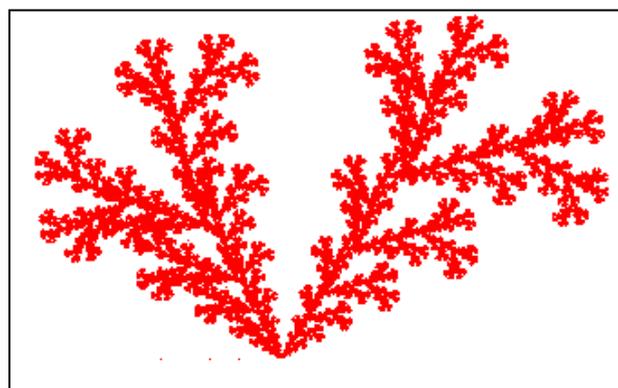


圖 2-35 IFS 珊瑚圖形

另外，以樹的圖形來說，建立 r, s, θ, ϕ, e, f 表，如表 2-6，則結果為圖 2-36

r	s	θ	ϕ	e	f
0.05	0.6	0	0	0	0
0.05	-0.5	0	0	0	1
0.6	0.5	$\frac{40\pi}{180}$	$\frac{40\pi}{180}$	0	0.6
0.5	0.45	$\frac{20\pi}{180}$	$\frac{20\pi}{180}$	0	1.1
0.5	0.55	$-\frac{30\pi}{180}$	$-\frac{30\pi}{180}$	0	1
0.55	0.4	$-\frac{40\pi}{180}$	$-\frac{40\pi}{180}$	0	0.7

表 2-7 IFS 生成樹木參數表

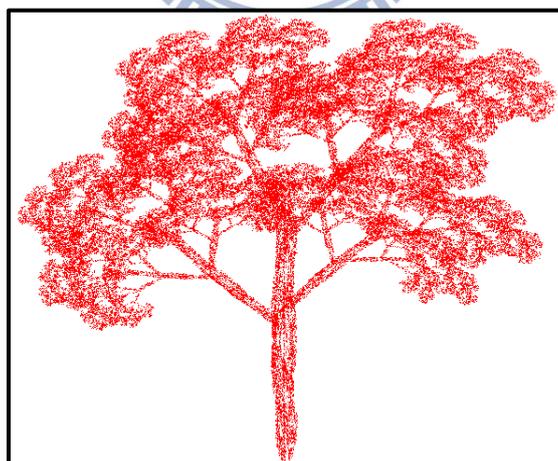


圖 2-36 IFS 樹木圖形

我們也可以利用 IFS 的方式繪製出羊齒葉圖形，首先先建立 r, s, θ, ϕ, e, f 表，如表 2-6，則結果為圖 2-37

r	s	θ	Φ	e	f
0	0.16	0	0	0	0
0.85	0.85	$\frac{-2.5\pi}{180}$	$\frac{-2.5\pi}{180}$	0	1.6
0.3	0.34	$\frac{49\pi}{180}$	$\frac{49\pi}{180}$	0	1.6
-0.3	0.37	$-\frac{50\pi}{180}$	$-\frac{50\pi}{180}$	0	0.44

表 2-6 IFS 生成樹木參數表

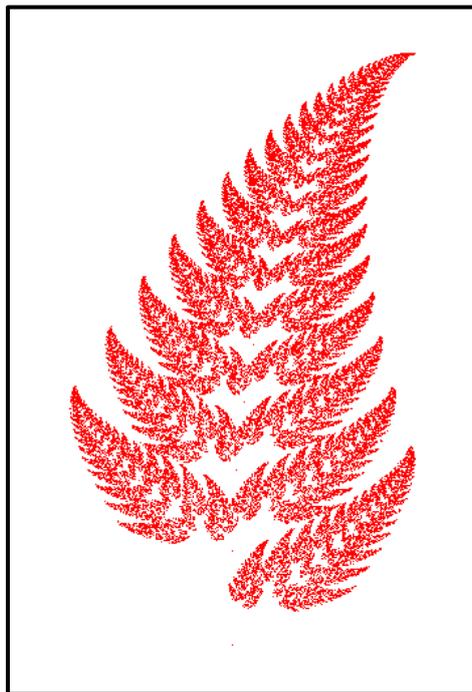


圖 2-37 IFS 羊齒葉圖形

甚至我們可以藉由一些貼片參數中， θ 和 Φ 參數很小很小的變動，讓羊齒葉擺動起來，例如圖 2-38，每隔一張圖便讓 θ 和 Φ 不為 0 的參數，同時增加 0.5 度，如此可以產生許多不同的變化，呈現出類似擺動的效果。

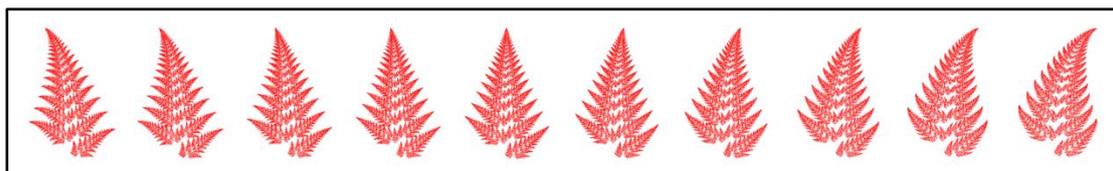


圖 2-38 IFS 羊齒葉擺動圖形

2.5.3 IFS 等值線圖

經由上一節介紹貼片理論以及 IFS 疊代函數產生圖形的產生方法之後，相信大家已經可以利用數學方程式，或是貼片的方法，來繪製一些碎形相關的圖形，其中在繪製貼片圖形時，是利用撒點的方式，讓點根據貼片的旋轉、縮放與偏移，產生碎形，因此在圖形的邊緣，在視覺上總是讓人感覺有點模模糊糊，只有點的變化，加上 mathcad 的顏色上的限制，變化上較少，所以我們可以利用一些小技巧，增加一些能見度與顏色上的變化。

在 2-4 章中，我們的方法就是利用給予圖形上的每一個點不同的高度，並藉由每一個底的高度不同的方式呈現，讓圖形具有一些變化，進而藉由高度繪製出等高線圖來產生高度變化的顏色。首先我們要先確定圖形的邊界位置，利用 min 參數找到 x 與 y 的最小值，max 參數找到 x 與 y 的最大值，並在 x,y 的邊界再多加上一些距離，經由掌握最大與最小的 x,y 值，使得整張圖形可以完整地呈現，接下來我們將圖形切割成 n 份，將整張圖切割 $n \times n$ 格，假設 n 為 80，圖形將切割成 6400 份，如圖 2-39

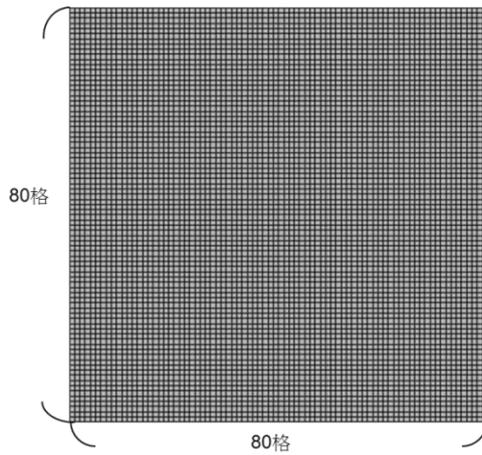


圖 2-39 80 × 80 格分割圖

其中 x 方向的每一個單位為 $\frac{\max(x)-\min(x)}{n}$ ，同理 y 方向的每一個單位為 $\frac{\max(y)-\min(y)}{n}$ ，接下來我們可以利用邏輯函數來判斷，若是在 IFS 疊代的過程中，x,y 的值有出現在某一個區間中，我們就會給該區間 1 的值，若沒有出現在該區間中，我們就給他 0 的值，並將我們給予這些區間的 0,1 值輸入於一個矩陣之中，若是 x,y 的值不斷的出現於該區間之中，則相對應的矩陣值會越高，如此藉由 x 區間的矩陣與 y 區間的矩陣相乘，可以得到一個矩陣結果，並由結果顯示出切割成 6400 塊的區間中，若出現 mathcad 中的 surface plot 圖形中，可以看見各區間不同的高度。先建立 r, s, θ , ϕ , e, f 表，表 2-7

r	s	θ	ϕ	e	f
0.5	0.49	0	0	0.2	0
0.5	0.49	0	0	0.2	0.51
0.74	0.5	$\frac{\pi}{4}$	0	0	0
0.74	0.5	$-\frac{\pi}{4}$	0	0.48	0.48

表 2-7 IFS 生成山脈參數表

所形成的圖形，如圖 2-40 所示

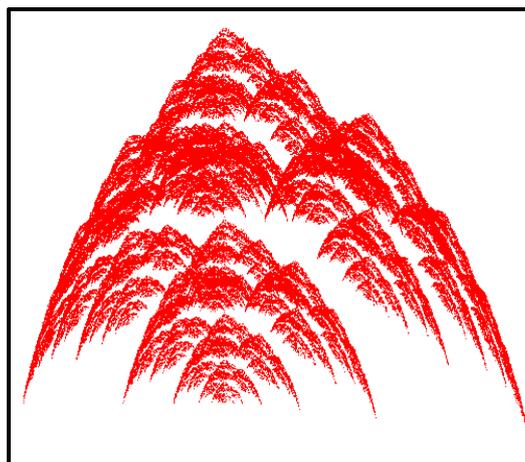


圖 2-40 IFS 山脈圖

繪製完基本的 IFS 圖形後，我們可以利用點的高度，產生類似等高線的高度的圖形，其中左邊呈現的為 surface plot 圖 2-41(a)，右邊呈現的為 contour plot 圖 2-41(b)

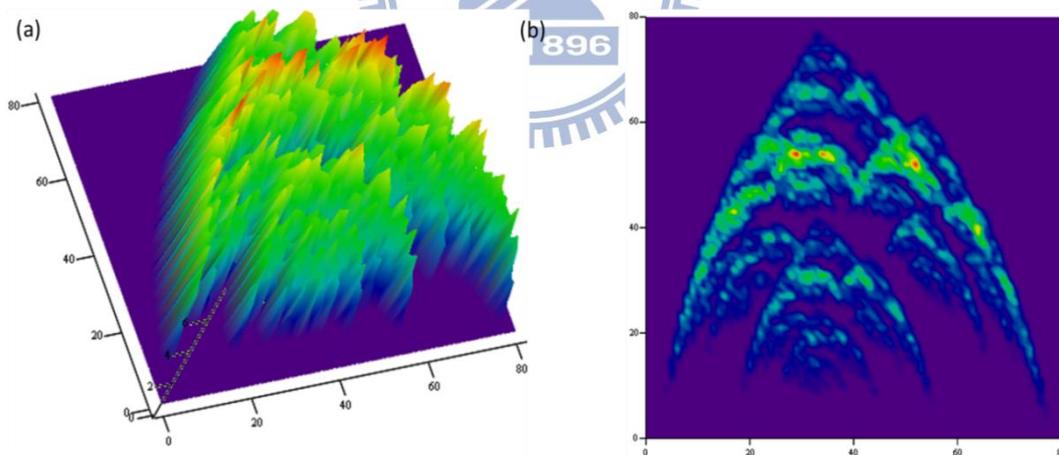


圖 2-41 IFS 山脈等值線圖

如此可以增加一些圖形集中程度與邊界的可視性，同時也可以為原來單調的顏色產生更多不同的變化，相關程式碼如附錄 2.7。

經由上面的程式來協助計算，我們可以經由給予點高度的方式，讓圖形增添許多高度的顏色，並呈現出三維空間的效果，同時也可以

讓圖形有較多的變化，但是在繪製的過程中，實際上會發現到 mathcad 的計算時間和過程十分冗長，解析度方面也會因為計算時間太長，導致解析度不能增加得太多，過程中也容易產生程式不穩定的情形，使得實際操作時，常常出現等待的狀況，也很困難的去轉換相關的結果。而造成程式計算緩慢原因就在於將剛始疊代的區塊切成二維空間之後，再將二維空間轉成矩陣才計算繪製圖形，這樣會導致程式計算程式時間拉得很長。

在繪製山脈圖形得等值線圖程式的前半部分都和繪製平面部分的方程式相同，在平面圖形程式碼後面部分我們會接續使用之前繪製 julia sets 的方法，先將圖形的平面範圍先分割成 800 份，使得整張平面圖分成 800×800 個區塊，接下來利用 z_j 值的上界與下界再各加減 0.05 的距離成為整張圖的邊界，如此可以使畫面可以有置中的效果，再來利用碎形的具有特性，將初始條件稍微修改，即可產生一些不同的變化，於是我們定義出，當初始值 R 小於重複繪製次數 $N = 3$ 時，則 $z_0 = rnd(1) + 1i \cdot rnd(1)$ 就會重複產生一個新的初始值，並經由初始質的改變，在原來的圖形上重新繪製一張新的圖形，在原來圖形上的每一個點的位置附近產生新的點，再給予每一個點重複累加的值，使圖形產生高度，並且在最後繪製圖形的程式碼中，利用次方數的關係，讓圖形的點的高度比例更集中，使圖形更清晰，相關程式碼如附錄 2.8。

而在前面繪製 Julia sets 和 Manderbrot sets 時，我們將二維空間的區塊的計算過程利用 floor 參數取商數和利用 mod 參數取餘數的方式，將原來的二維空間的計算過程轉換成一維空間來處理，同理，在此我們也可以將原來的二維空間區塊先轉換成一維空間之後，等待程式計算完成，在繪製圖形時，再將原來一維的結果轉回二維空間，將點的

位置描繪在圖形上，再者我們可以利用多次繪製的方式，讓圖形經由起始點些微的變化，經由連續的疊代過程，產生幾張極為接近的圖形的點，散佈在同一張圖面上，經由連續的繪製，如此可以增加圖形的清晰程度，讓圖形的線條呈現更清楚。除此之外，我們可以經由之前敘述過的方法，先將 x,y 方向的極大值與極小值得點取出後，在邊界再加上一點點空間，使得圖形可以完整地顯示，在之後的程式碼呈現，我們會接續之前用複數的實部、虛部分別呈現的做法將 x,y 平面轉換為 z_j 的方式，並用方程式的方式繪製出有關圖形。

繪製 1 次，取 0.25 次方

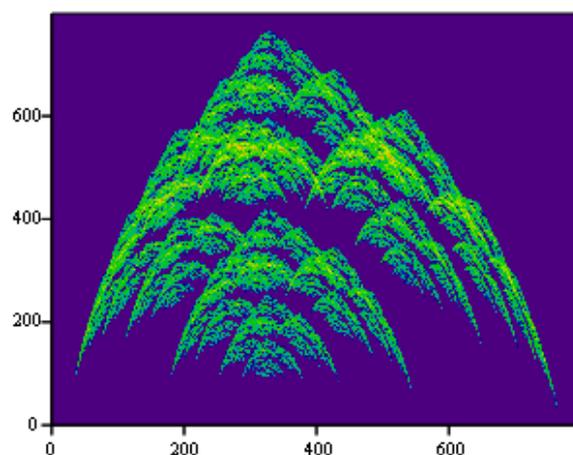


圖 2-42 IFS 山脈圖取 0.25 次方連續繪製 1 次圖

繪製 5 次，取 0.25 次方

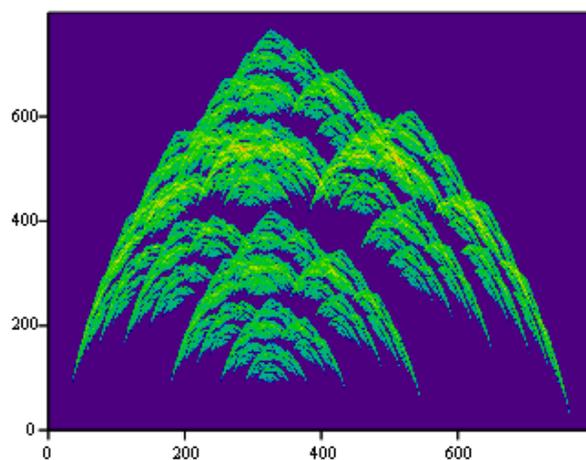


圖 2-43 IFS 山脈圖取 0.25 次方連續繪製 5 次圖

繪製 10 次，取 0.25 次方，

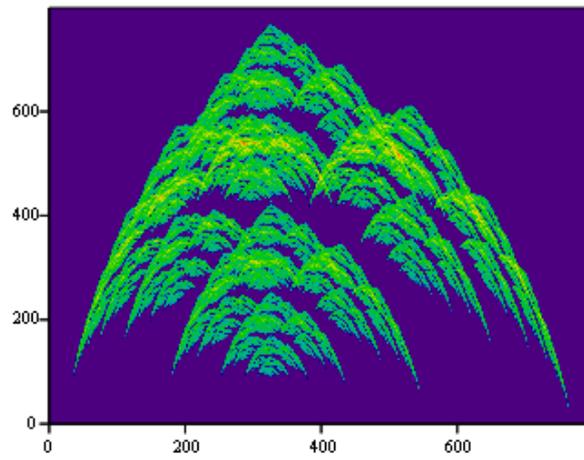
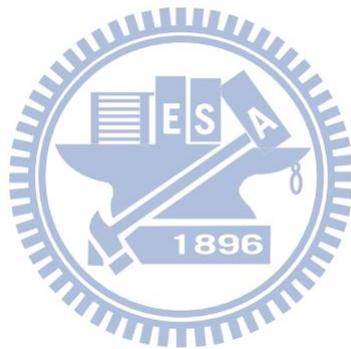


圖 2-44 IFS 山脈圖取 0.25 次方連續繪製 10 次圖



附錄 2.1

分形樹 MathCad 程式碼

第一層分形樹程式碼：

$$z_0 = 0$$

$$z_9 = e^{i\frac{\pi}{2}},$$

$$z_1 = \frac{2}{3}z_0 + \frac{1}{3}z_9$$

$$z_2 = z_1$$

$$z_4 = z_1$$

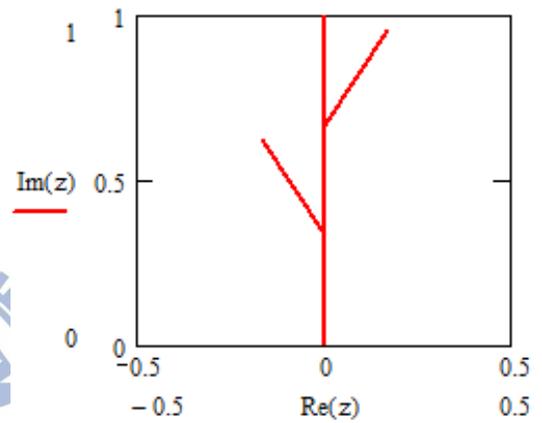
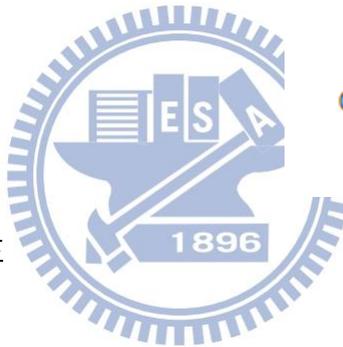
$$z_5 = \frac{1}{3}z_0 + \frac{2}{3}z_9$$

$$z_6 = z_5$$

$$z_8 = z_5$$

$$z_3 = z_1 + (z_5 - z_1)e^{i\frac{\pi}{6}}$$

$$z_7 = z_8 + (z_9 - z_8)e^{i\frac{-\pi}{6}}$$



第二層分形樹程式碼：

k:= 0..4

$$Z_{0+(k+1)*10} := Z_{2k} \quad Z_{9+(k+1)*10} := Z_{2k+1}$$

$$Z_{1+(k+1)*10} := \frac{2}{3}Z_{0+(k+1)*10} + \frac{1}{3}Z_{9+(k+1)*10}$$

$$Z_{2+(k+1)*10} := Z_{1+(k+1)*10}$$

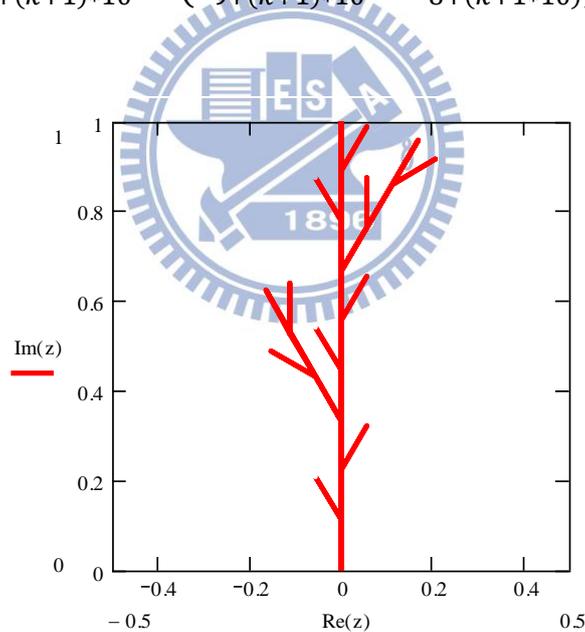
$$Z_{4+(k+1)*10} := Z_{1+(k+1)*10}Z_{5+(k+1)*10} := \frac{1}{3}Z_{0+(k+1)*10} + \frac{2}{3}Z_{9+(k+1)*10}$$

$$Z_{6+(k+1)*10} = Z_{5+(k+1)*10}$$

$$Z_{8+(k+1)*10} = Z_{5+(k+1)*10}$$

$$Z_{3+(k+1)*10} = Z_{1+(k+1)*10} + (Z_{5+(k+1)*10} - Z_{4+(k+1)*10})e^{i\frac{\pi}{6}}$$

$$Z_{7+(k+1)*10} = Z_{8+(k+1)*10} + (Z_{9+(k+1)*10} - Z_{8+(k+1)*10})e^{i\frac{-\pi}{6}}$$



第三層分形樹程式碼：

$k := 0..24$

$$Z_{0+(k+1)*10+50} := Z_{2k+10} \quad Z_{9+(k+1)*10+50} := Z_{2k+1+10}$$

$$Z_{1+(k+1)*10+50} := \frac{2}{3}Z_{0+(k+1)*10+50} + \frac{1}{3}Z_{9+(k+1)*10+50}$$

$$Z_{2+(k+1)*10+50} := Z_{1+(k+1)*10+50}$$

$$Z_{4+(k+1)*10+50} := Z_{1+(k+1)*10+50}$$

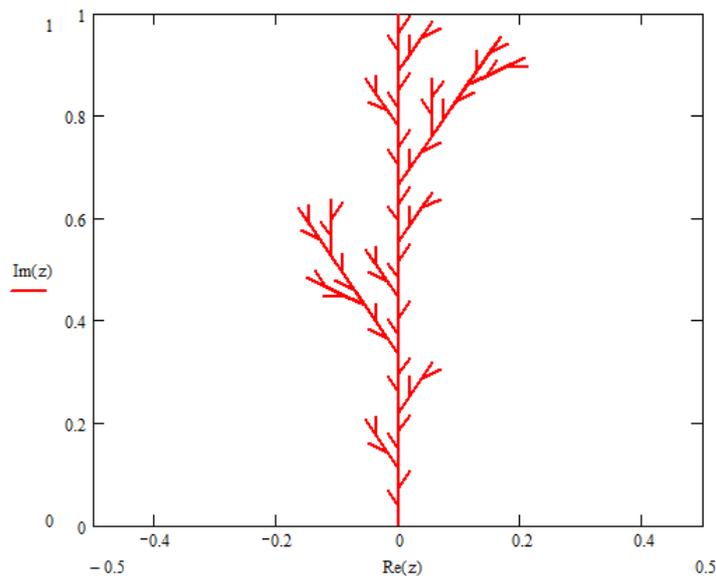
$$Z_{5+(k+1)*10+50} := \frac{1}{3}Z_{0+(k+1)*10+50} + \frac{2}{3}Z_{9+(k+1)*10+50}$$

$$Z_{6+(k+1)*10+50} = Z_{5+(k+1)*10+50}$$

$$Z_{8+(k+1)*10+50} = Z_{5+(k+1)*10+50}$$

$$Z_{3+(k+1)*10+50} = Z_{1+(k+1)*10+50} + (Z_{5+(k+1)*10+50} - Z_{4+(k+1)*10+50})e^{i\frac{\pi}{6}}$$

$$Z_{7+(k+1)*10+50} = Z_{8+(k+1)*10+50} + (Z_{9+(k+1)*10+50} - Z_{8+(k+1)*10+50})e^{i\frac{-\pi}{6}}$$



第四層分形樹程式碼：

$k := 0.124$

$$Z_{0+(k+1)*10+50+250} := Z_{2k+10+50} \quad Z_{9+(k+1)*10+50+250} := Z_{2k+1+10+50}$$

$$Z_{1+(k+1)*10+50+250} := \frac{2}{3} Z_{0+(k+1)*10+50+250} + \frac{1}{3} Z_{9+(k+1)*10+50+250}$$

$$Z_{2+(k+1)*10+50+250} := Z_{1+(k+1)*10+50+250}$$

$$Z_{4+(k+1)*10+50+250} := Z_{1+(k+1)*10+50+250}$$

$$Z_{5+(k+1)*10+50+250} := \frac{1}{3} Z_{0+(k+1)*10+50+250} + \frac{2}{3} Z_{9+(k+1)*10+50+250}$$

$$Z_{6+(k+1)*10+50+250} := Z_{5+(k+1)*10+50+250}$$

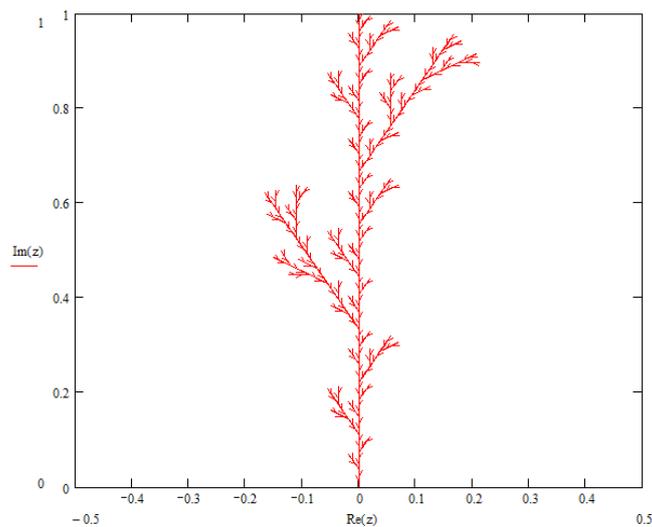
$$Z_{8+(k+1)*10+50+250} := Z_{5+(k+1)*10+50+250}$$

$$Z_{3+(k+1)*10+50+250} :=$$

$$Z_{1+(k+1)*10+50+250} + (Z_{5+(k+1)*10+50+250} - Z_{4+(k+1)*10+50+250}) e^{i\frac{\pi}{6}}$$

$$Z_{7+(k+1)*10+50+250} :=$$

$$Z_{8+(k+1)*10+50+250} + (Z_{9+(k+1)*10+50+250} - Z_{8+(k+1)*10+50+250}) e^{i\frac{-\pi}{6}}$$



附錄 2.2

Koch 雪花曲線 MathCad 程式碼

$$m := 5 \quad z_0 := 0 \quad z_1 := \left(\frac{1}{3}\right)^m \cdot e^{i\frac{\pi}{2}} \quad \theta_{4^m} := -\frac{2\pi}{3} \quad \theta_{2 \cdot 4^m} := -\frac{2\pi}{3}$$

$$k := 1..4^m - 1$$

$$\theta_k := (\text{mod}(k, 4) - 1) \cdot (-\pi) + \frac{1}{3}\pi$$

$$\theta_k := \text{if} \left[\text{mod}(k, 4) = 0, \left(\text{mod} \left(\frac{k}{4}, 4 \right) - 1 \right) \cdot (-\pi) + \frac{1}{3}\pi, \theta_k \right]$$

$$\theta_k := \text{if} \left[\text{mod}(k, 4) = 0, \left(\text{mod} \left(\frac{k}{4^2}, 4 \right) - 1 \right) \cdot (-\pi) + \frac{1}{3}\pi, \theta_k \right]$$

$$\theta_k := \text{if} \left[\text{mod}(k, 4) = 0, \left(\text{mod} \left(\frac{k}{4^3}, 4 \right) - 1 \right) \cdot (-\pi) + \frac{1}{3}\pi, \theta_k \right]$$

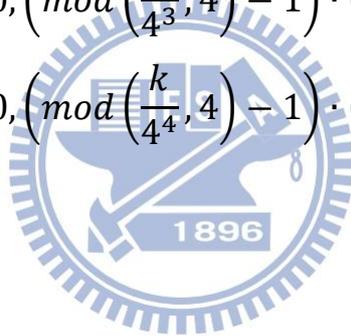
$$\theta_k := \text{if} \left[\text{mod}(k, 4) = 0, \left(\text{mod} \left(\frac{k}{4^4}, 4 \right) - 1 \right) \cdot (-\pi) + \frac{1}{3}\pi, \theta_k \right]$$

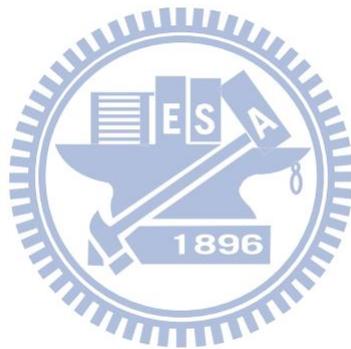
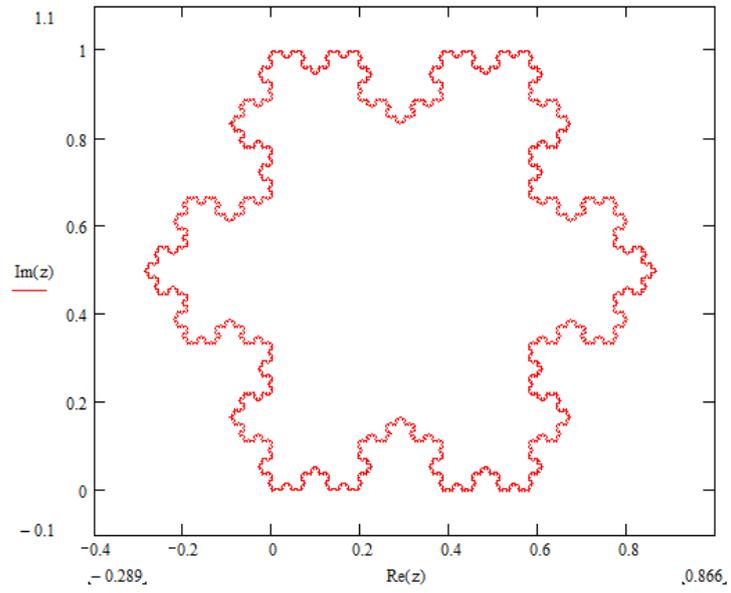
$$\theta_{k+4^m} := \theta_k$$

$$\theta_{k+2 \cdot 4^m} := \theta_k$$

$$k := 1..3 \cdot 4^m - 1$$

$$z_{k+1} = z_k + (z_k - z_{k-1}) \cdot e^{i\theta_k}$$





附錄 2.3

Julia sets MathCad 程式碼

$$M := 600 \quad a := 1.75 \quad N := M^2$$

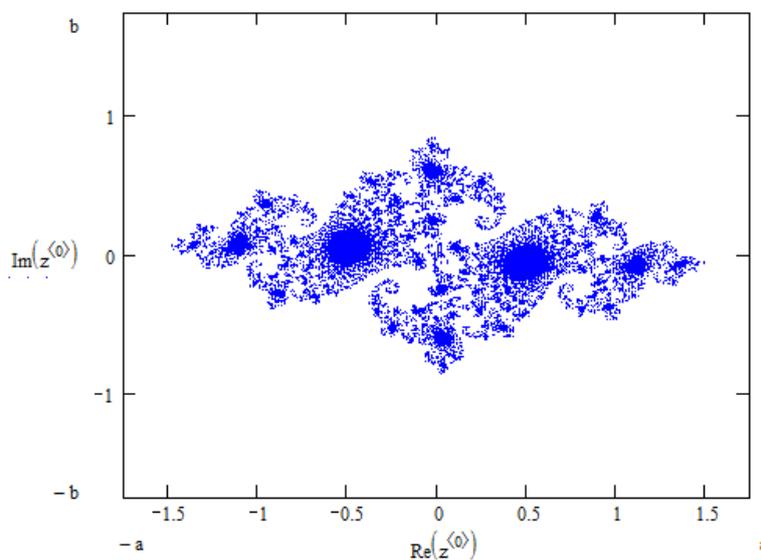
$$C := -0.745429 + 0.113008 * 1i \quad P := 2 \quad i := 0..N - 1$$

$$z0_i = \left(\text{floor} \left(\frac{i}{M} \right) + \frac{\text{rnd}(1)}{2} \right) \cdot \frac{2a}{M} - a + i$$

$$\cdot \left[\frac{\left[\text{mod}(i, M) + \frac{\text{rnd}(1)}{2} \right]}{M} \cdot 2a - a \right]$$

$$\text{Julia}(z0, C, P) = \left[\begin{array}{l} i \leftarrow 0 \\ j \leftarrow 0 \\ \text{while } i < \text{last}(z0) \\ \quad Z_0 \leftarrow z0_i \\ \quad \text{for } j \in 0..200 \\ \quad \quad Z_{j+1} \leftarrow (Z_j)^2 + C \\ \quad \quad n \leftarrow j + 1 \\ \quad \quad \text{break if } |Z_{j+1}| > P \\ \quad z0_i \leftarrow 0 \text{ if } |Z_n| > P \\ \quad i \leftarrow i + 1 \\ \quad z0 \end{array} \right]$$

$$z := \text{Julia}(z0, C, P)$$



附錄 2.4

Mandelbrot sets MathCad 程式碼

M := 300 N := 200 N1 := M² a := 1.5 b := 1

Δa := -0.25 Δb := 0 z0 := 0 N1 := M² i := 0..N1 - 1

$$C_i = \left(\text{floor} \left(\frac{i}{M} \right) + \frac{\text{rnd}(1)}{2} \right) \cdot \frac{2a}{M} - a + \Delta a + i$$

$$\cdot \left[\frac{\left[\text{mod}(i, M) + \frac{\text{rnd}(1)}{2} \right]}{M} \cdot 2b - b + \Delta b \right]$$

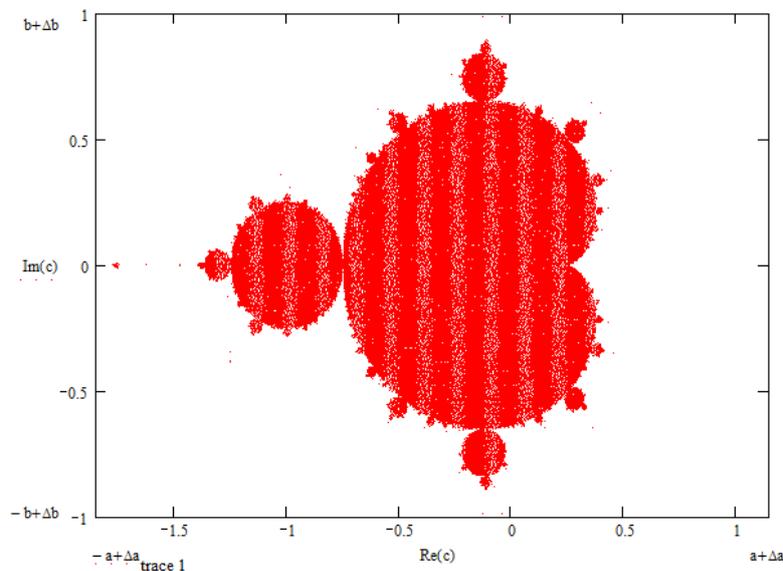
Mand(z0, C, N, P) =

```

    i ← 0
    j ← 0
    while i < last(C)
        Z0 ← z0
        for j ∈ 0..N - 1
            Zj+1 ← (Zj)2 + Cj
            n ← j + 1
        break if |Zj+1| > P
        Ci ← 0 if |Zn| > P
        i ← i + 1
    C

```

c := Mand(z0, C, N, P)



附錄 2.5

Julia sets 等值線圖 MathCad 程式碼

M:= 600 N:= 700 a:= 1.5 b:= 1 N1:= M²

C := -0.745429 + 0.113008 * 1i

P:= 2

i:= 0..N - 1

m_i:= 0

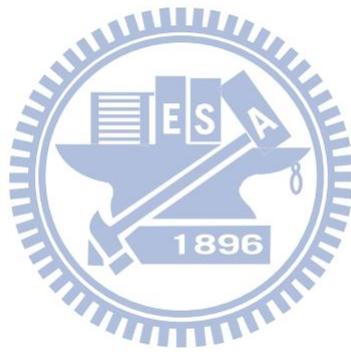
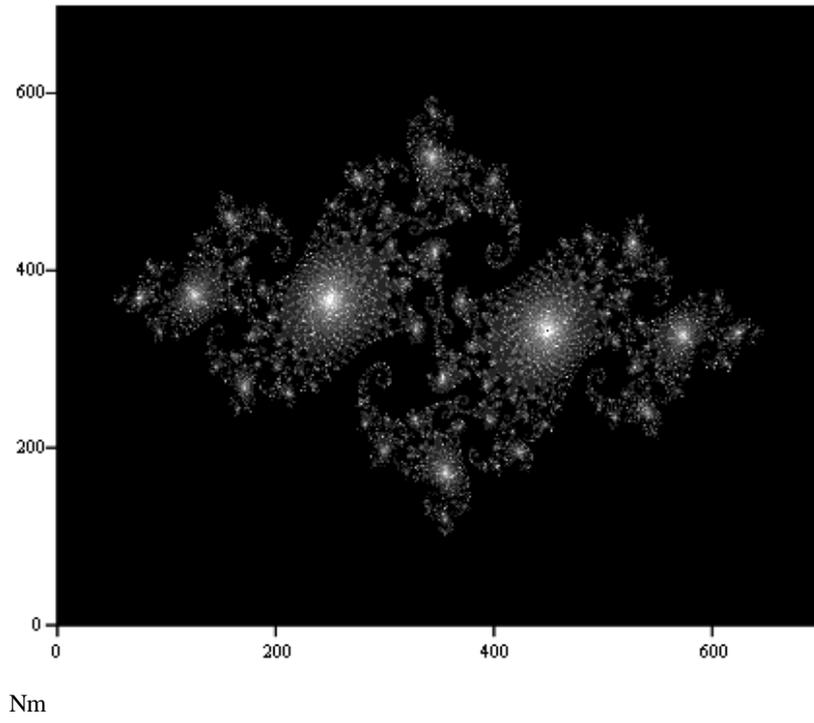
$z0_i = \left(\text{floor}\left(\frac{i}{M}\right) + \frac{\text{rnd}(1)}{2} \right) \cdot \frac{2a}{M} - a + i$

$\cdot \left[\frac{\left[\text{mod}(i, M) + \frac{\text{rnd}(1)}{2} \right]}{M} \cdot 2b - b \right]$

Julia(z0, m, C, N, P) = $\left| \begin{array}{l} i \leftarrow 0 \\ j \leftarrow 0 \\ \text{while } i < \text{last}(z0) \\ \quad Z_0 \leftarrow z0_i \\ \quad \text{for } j \in 0..200 \\ \quad \quad Z_{j+1} \leftarrow (Z_j)^2 + C \\ \quad \quad n \leftarrow j + 1 \\ \quad \quad \text{break if } |Z_{j+1}| > P \\ \quad \quad m_i \leftarrow n \\ \quad z0_i \leftarrow 0 \text{ if } |Z_n| > P \\ \quad i \leftarrow i + 1 \\ \quad \text{augment}(z0, m) \end{array} \right|$

z:= Julia(z0, m, C, N, P)

$Nm_{\text{floor}\left(\frac{i}{M}\right), \text{mod}(i, M)} := (z_{i,1})^1$



附錄 2.6

IFS 雪花 MathCad 程式碼

$N := 50000$

$n := 0..N$

$x_n := 0$

$y_n := 0$

$d_n := rnd(1)$

$i := 1..4$

$$r_1 := \frac{1}{3} \quad s_1 := \frac{1}{3} \quad \theta_1 := 0 \quad \phi_1 := 0 \quad e_1 := 0 \quad f_1 := 0$$

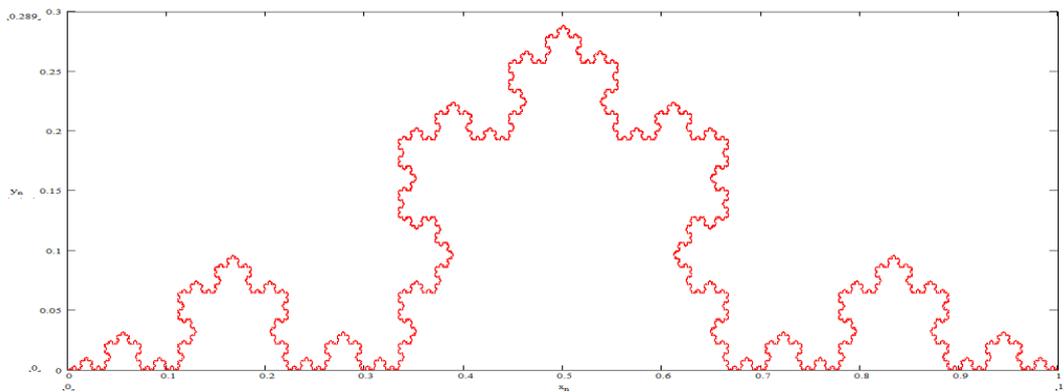
$$r_2 := \frac{1}{3} \quad s_2 := \frac{1}{3} \quad \theta_2 := \frac{\pi}{3} \quad \phi_2 := \frac{\pi}{3} \quad e_2 := \frac{1}{3} \quad f_2 := 0$$

$$r_3 := \frac{1}{3} \quad s_3 := \frac{1}{3} \quad \theta_3 := -\frac{\pi}{3} \quad \phi_3 := -\frac{\pi}{3} \quad e_3 := \frac{1}{2} \quad f_3 := \frac{\sqrt{3}}{6}$$

$$r_4 := \frac{1}{3} \quad s_4 := \frac{1}{3} \quad \theta_4 := 0 \quad \phi_4 := 0 \quad e_4 := \frac{2}{3} \quad f_4 := 0$$

$$P_i := \frac{i}{4}$$

$$\begin{aligned} \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \sum_{i=1}^4 \left[\begin{pmatrix} r_i \cdot \cos(\theta_i) & -s_i \cdot \sin(\phi_i) \\ r_i \cdot \sin(\theta_i) & s_i \cdot \cos(\phi_i) \end{pmatrix} \cdot \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} \right] \cdot \Phi(d_n \\ - P_{i-1}) \Phi(P_i - d_n) \end{aligned}$$



附錄 2.7

山脈圖形等值線圖 MathCad 程式碼

$N := 50000 \quad j := 0..N \quad x_j := 0 \quad y_j := 0 \quad d_j := rnd(1)$

$i := 1..4$

$r_1 := 0.5 \quad s_1 := 0.49 \quad \theta_1 := 0 \quad \phi_1 := 0 \quad e_1 := 0.2 \quad f_1 := 0$

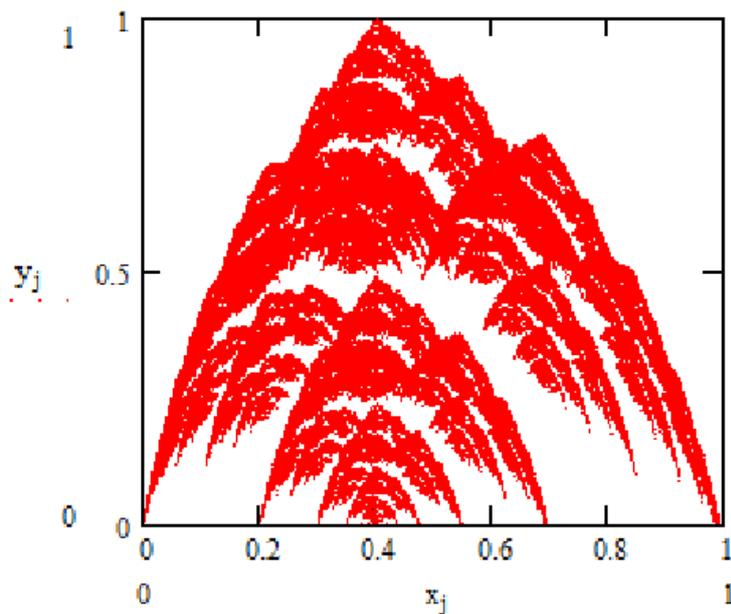
$r_2 := 0.5 \quad s_2 := 0.49 \quad \theta_2 := 0 \quad \phi_2 := 0 \quad e_2 := 0.2 \quad f_2 := 0.51$

$r_3 := 0.74 \quad s_3 := 0.5 \quad \theta_3 := \frac{\pi}{4} \quad \phi_3 := 0 \quad e_3 := 0 \quad f_3 := 0$

$r_4 := 0.74 \quad s_4 := 0.5 \quad \theta_4 := \frac{-\pi}{4} \quad \phi_4 := 0 \quad e_4 := 0.48 \quad f_4 := 0.48$

$P_i := \frac{i}{4}$

$$\begin{pmatrix} x_{j+1} \\ y_{j+1} \end{pmatrix} = \sum_{i=1}^4 \left[\begin{pmatrix} r_i \cdot \cos(\theta_i) & -s_i \cdot \sin(\phi_i) \\ r_i \cdot \sin(\theta_i) & s_i \cdot \cos(\phi_i) \end{pmatrix} \cdot \begin{pmatrix} x_j \\ y_j \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} \right] \cdot \Phi(d_j - P_{i-1}) \Phi(P_i - d_j)$$



$X_{mi} := \min(x) - 0.05$

$X_{ma} := \max(x) + 0.05$

$Y_{mi} := \min(y) - 0.05$

$Y_{ma} := \max(y) + 0.05$

$n := 80 \quad t := 0..n \quad m := 0..n$

$$X_t := \frac{t}{n}(X_{ma} - X_{mi}) + X_{mi}$$

$$Y_m := \frac{m}{n}(Y_{ma} - Y_{mi}) + Y_{mi}$$

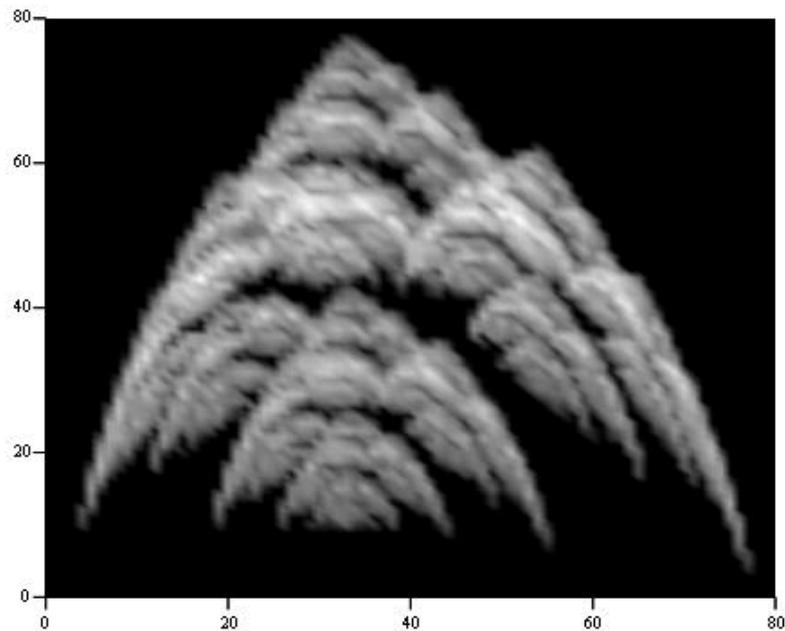
$$t := 1..n \quad m := 1..n$$

$$A1_{t,j} := \text{if}[(X_t \geq x_j \geq X_{t-1}), 1, 0] \quad A2_{j,m} := \text{if}[(Y_m \geq y_j \geq Y_{m-1}), 1, 0]$$

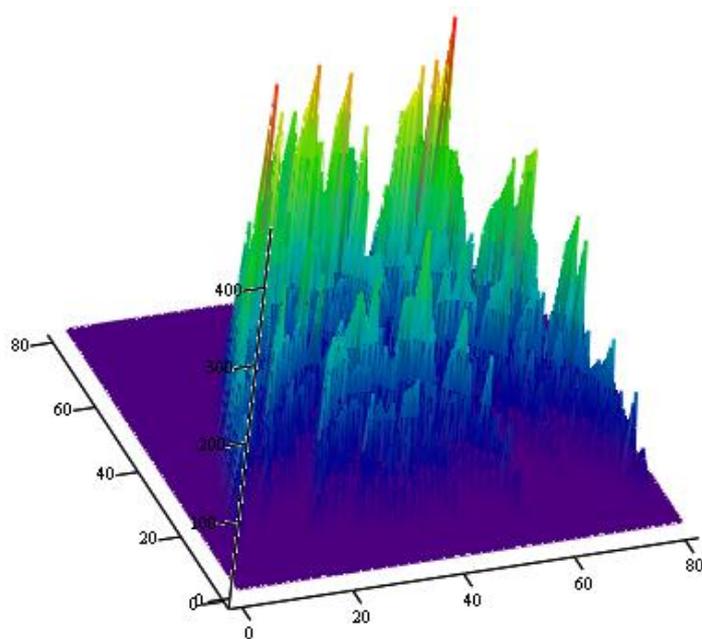
$$A := A1 \cdot A2$$

$$B_{n,m} := (A_{n,m})^{\frac{1}{3}}$$

coutour plot



surface plot



附錄 2.8

山脈等值線圖清晰化 MathCad 程式碼

$N := 50000 \quad j := 0..N \quad x_j := 0 \quad y_j := 0 \quad d_j := rnd(1)$

$i := 1..4$

$r_1 := 0.5 \quad s_1 := 0.49 \quad \theta_1 := 0 \quad \phi_1 := 0 \quad e_1 := 0.2 \quad f_1 := 0$

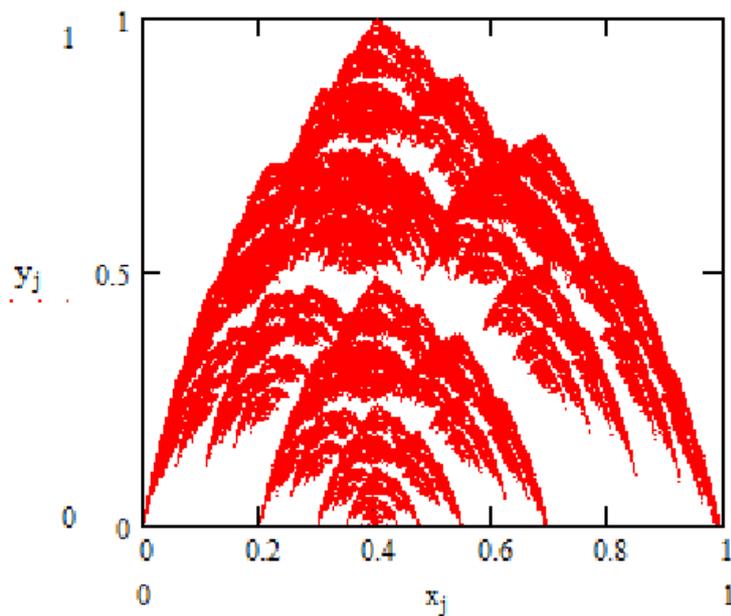
$r_2 := 0.5 \quad s_2 := 0.49 \quad \theta_2 := 0 \quad \phi_2 := 0 \quad e_2 := 0.2 \quad f_2 := 0.51$

$r_3 := 0.74 \quad s_3 := 0.5 \quad \theta_3 := \frac{\pi}{4} \quad \phi_3 := 0 \quad e_3 := 0 \quad f_3 := 0$

$r_4 := 0.74 \quad s_4 := 0.5 \quad \theta_4 := \frac{-\pi}{4} \quad \phi_4 := 0 \quad e_4 := 0.48 \quad f_4 := 0.48$

$P_i := \frac{i}{4}$

$$\begin{pmatrix} x_{j+1} \\ y_{j+1} \end{pmatrix} = \sum_{i=1}^4 \left[\begin{pmatrix} r_i \cdot \cos(\theta_i) & -s_i \cdot \sin(\phi_i) \\ r_i \cdot \sin(\theta_i) & s_i \cdot \cos(\phi_i) \end{pmatrix} \cdot \begin{pmatrix} x_j \\ y_j \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} \right] \cdot \Phi(d_j - P_{i-1}) \Phi(P_i - d_j)$$



$z_j := x_j + 1i * y_j$

$Xmi := \min(Re(z)) - 0.05$

$Xma := \max(Re(z)) + 0.05$

$Ymi := \min(Im(z)) - 0.05$

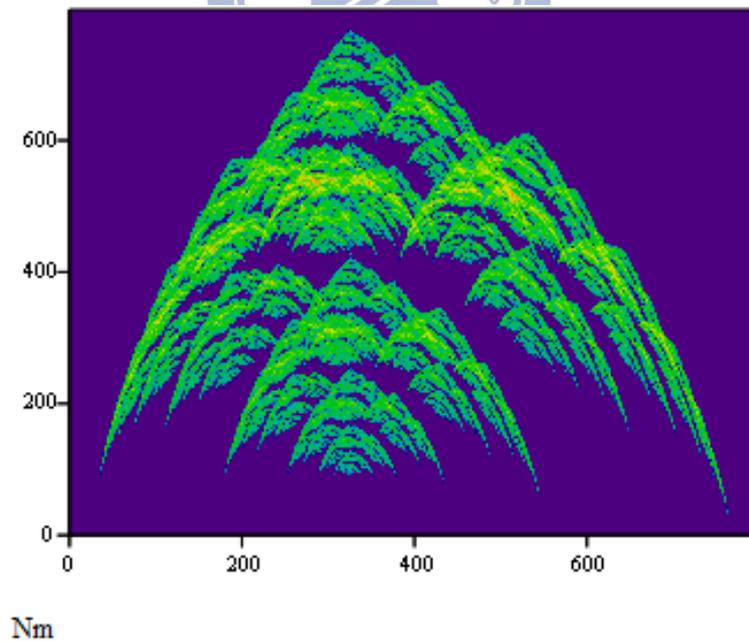
$Yma := \max(Im(z)) + 0.05$

ILS($r, s, \theta, \phi, e, f, P, N, M, Xmi, Xma, Ymi, Yma, n$)

$$\begin{aligned}
 & R \leftarrow 0 \\
 & \text{while } R < N \\
 & \quad z_0 \leftarrow \text{rnd}(a) + 1i * \text{rnd}(a) \\
 & \quad \text{for } j \in 0..M^2 - 1 \\
 & \quad \quad q_j \leftarrow \text{rnd}(1) \\
 & \quad \quad z_{j+1} \leftarrow \sum_{k=1}^4 \left[\begin{aligned} & \left[\left(r_k \cdot \cos(\theta_k) \cdot \text{Re}(z_j) - s_k \cdot \sin(\phi_k) \cdot \text{Im}(z_j) \right) + e_k \right] \\ & + \\ & \left(\left(r_k \cdot \sin(\theta_k) \cdot \text{Re}(z_j) + s_k \cdot \cos(\phi_k) \cdot \text{Im}(z_j) \right) + f_k \right) \cdot 1i \end{aligned} \right] \\
 & \quad \quad \quad \cdot \Phi(d_j - P_{i-1}) \Phi(P_i - d_j) \\
 & \quad \quad k \leftarrow \text{floor} \left(\frac{\text{Re}(z_{j+1}) - Xmi}{Xma - Xmi} \cdot M \right) \cdot M + \text{floor} \left(\frac{\text{Im}(z_{j+1}) - Ymi}{Yma - Ymi} \cdot M \right) \\
 & \quad \quad \quad n_{k+1} \leftarrow n_k \\
 & \quad \quad R \leftarrow R + 1 \\
 & \quad n
 \end{aligned}$$

I := ILS($r, s, \theta, \phi, e, f, P, 5, M, Xmi, Xma, Ymi, Yma, n$)

$Nm_{\text{floor}(\frac{i}{M}), \text{mod}(i, M)} := (I_i)^{0.25}$



第三章 波的干涉圖騰之視覺化

在我們的生活周遭常常可以看見一些有關波的現象，每一個人的生活經驗中，一定會曾經見過水的波紋過在水的表面漸漸散開的樣子，同心圓的紋路，不斷的由波的中心漸漸散開，而在許多物理性質中，波的性質與探討具有很重大的意義[1]，因此我們希望可以藉由數學方程式，利用 mathcad 軟體描繪出波紋的圖案。在波的干涉圖案中，比較為人熟知的就是單狹縫與雙狹縫的干涉圖案，除此之外，最近受到很多注目的準晶格圖案，亦可由波的干涉產生，所以本章節，主要探討波干涉產生準晶格圖案的視覺化。

3.1 波

波或波動在物理中是空間上傳播的一種物理現象，而電磁波能夠在真空中傳播，其餘大部分波如水波、繩波等機械波只能在某些介質中傳播。依波在介質中的震動傳遞的方向，可以分為縱波和橫波，其中縱波的特點是介質的振動方向與傳播方向相同，比如空氣中的聲波、地震波中的P波。橫波的特點是介質的振動方向與傳播方向垂直。如：電磁波、地震波中的S波。

最常見的波就是行進波，例如 sin 函數的波形。我們可以給予 sin 函數的參數式如 3.1 式，來描述 sin 函數的行進與變化，

$$y(x, t) = A(x, t)\sin(kx - \omega t + \phi) \quad (3.1)$$

其中， $A(x, t)$ 代表波的振幅， x 是位置， t 是時間， k 是波數， ϕ 是初始相位。常見的 sin 圖形常常會簡化相位，令 $t = 0$ ， $\phi = 0$ ，將 $A(x, t) = 1$ ， $k = 5$ ，如圖 3-1。

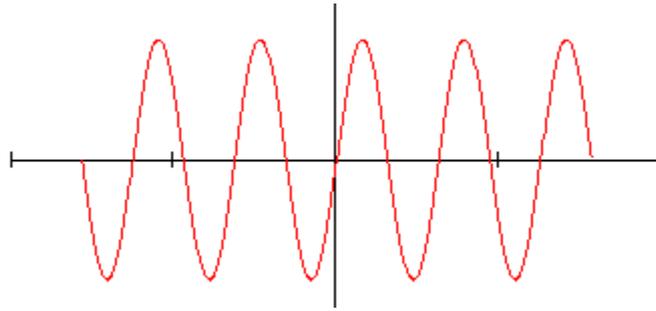


圖 3-1 $\sin 5x$ 函數圖形 $[-\pi, \pi]$

其中波的相位與振幅會決定同一個波的形狀差異，經由改變振幅，會使波的形狀放大或縮小，而相位的改變，會使波位移產生變化。

而駐波，是指波維持在固定的位置，而駐波通過時，每一個質點皆作簡諧運動。各質點振盪的幅度不相等，振幅為零的點稱為節點或波節 (Node)，振幅最大的點位於兩節點之間，稱為波腹 (Antinode)。撥動兩端固定張緊的弦，使波經兩固定端反射可干涉產生駐波。弦的兩固定端必為節點。



3.2 波的干涉

想像一下，在平靜的湖面，同時丟進兩個落入不同位置的石頭，那泛起陣陣的漣漪波紋，逐漸在湖面交錯，而兩個彼此交錯的波紋，所產生的紋路就是所謂的干涉現象。在物理學中，干涉是兩列或兩列以上的波在空間中重疊時發生疊加從而形成新波形的現象。因為兩種波同時存在，因此彼此會發生互相影響的現象，使得光的波紋產生許多變化，其中在近代物理中，藉由波的干涉現象，來證明光具有波動性，為了製造出足夠強的光源，雷射的技術得到廣泛的應用，在各種精密測量或是實驗中都能見到雷射干涉儀的身影。現在人們知道，兩束電磁波的干涉是彼此振動的電場強度向量疊加的結果，而由於光的波粒二象性，光的干涉也是光子自身的機率幅疊加的結果。

兩列波在同一介質中傳播發生重疊時，重疊範圍內介質的質點同時受到兩個波的作用。若波的振幅不大，此時重疊範圍內介質質點的振動位移等於各別波動所造成位移的向量和，這稱為波的疊加原理。若兩波的波峰（或波谷）同時抵達同一地點，稱兩波在該點同相，干涉波會產生最大的振幅，稱為相長干涉（建設性干涉）；若兩波之一的波峰與另一波的波谷同時抵達同一地點，稱兩波在該點反相，干涉波會產生最小的振幅，稱為相消干涉（摧毀性干涉）[12]。

在波的干涉實驗中，楊氏雙狹縫實驗是最早被提出的光的干涉演示實驗（托馬斯·楊，1801年），這一實驗的重要意義在於它是對光的波動說的有力支持，由於實驗觀測到的干涉條紋是牛頓所代表的光的微粒說無法解釋的現象，雙縫實驗使大多數的物理學家從此逐漸接受了光的波動理論。

在 3.1 式中，我們可以知道，距離會決定波的相位，換句話說，設定不同的距離觀察，我們所看到的波的干涉圖騰會不同，因此我們要先決定距離，在初始的平面上，令兩點波源的位置分別為 (x_1, y_1) 與 (x_2, y_2) ，在距離 L 的位置的平面 Z 上觀察波的疊加，在平面 Z 上的各點位置令為 X_m 和 Y_n ，在平面 Z 上我們所要畫的範圍為 $-d$ 到 d ，同時我們也要決定所要繪製的範圍的精細程度 $N = 400$ ，則 $X_m = -d + \frac{2d}{N} \cdot m$ ， $Y_n = -d + \frac{2d}{N} \cdot n$ ，則波源和平面 Z 上的各點距離為

$$\begin{aligned} d_1 &= \sqrt{(X_m - x_1)^2 + (Y_n - y_1)^2 + L^2} \\ d_2 &= \sqrt{(X_m - x_2)^2 + (Y_n - y_2)^2 + L^2} \end{aligned} \quad (3.2)$$

根據不同的位置定義出該點的相位為

$$\frac{2\pi}{\lambda} \sqrt{(X_m - x_2)^2 + (Y_n - y_2)^2} \quad (3.3)$$

決定好上述的條件，我們就可以畫出波源的干涉圖案，如圖 3-2。

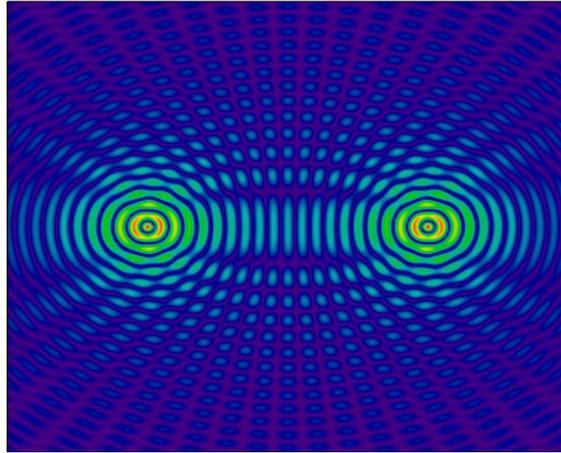


圖 3-2 波的干涉圖形

相關的程式碼如附錄 3.1。

3.3 準晶格圖騰

準晶體，亦稱為「準晶」或「擬晶」，是一種介於晶體和非晶體之間的固體。準晶體具有與晶體相似的長程有序的原子排列；但是準晶體不具備晶體的平移對稱性。普通晶體只能具有二次、三次、四次或六次旋轉對稱性，但是準晶格的具有其他的對稱性，例如五次對稱性或者更高的如六次以上的對稱性。

數學家在 20 世紀 60 年代就發現了這種非周期平鋪 (aperiodic tilings) 圖形。但是直到快 20 年後這種理論上的結構才和準晶的研究聯繫起來。20 世紀 80 年代在開始受到重視。在 1976 年被 Roger Penrose 解決 Wang tiles 的非週期性規則後發現 Quasi-crystal 的圖樣，Quasi-crystal 圖樣的無平移對稱而有獨特的旋轉對稱的性質吸引許多人的研究，其中有許多伊斯蘭教的幾何藝術也都呈現出 Quasi-crystal 的圖樣[13]。

在自然界中有許多礦物他們的晶格排列可區分為無結晶形、準結晶形及結晶形。在兩個世紀以前，科學家只發現於 2、3、4 與 6 軸對

稱的晶格，其他如5 與7 軸對稱的晶格，甚至於更高階軸的對稱晶格未
 在礦物中發現 [14]。在波的干涉中，我們可以利用將波源平均分布在
 一個圓上，並且經由波源與波源之間的干涉情況，產生準晶格的圖
 案，因此，我們可以定義在x-y平面中，以原點為中心畫圓，將波
 源的位置 $x_j = R \cdot \cos(\frac{2\pi}{N} \cdot j)$ 和 $y_j = R \cdot \sin(\frac{2\pi}{N} \cdot j)$ ，其中R為所要畫圓的
 大小，j則為相對應的布點位置，如圖3-3將點波源的分布由1點逐次
 增加到12點的圖形。

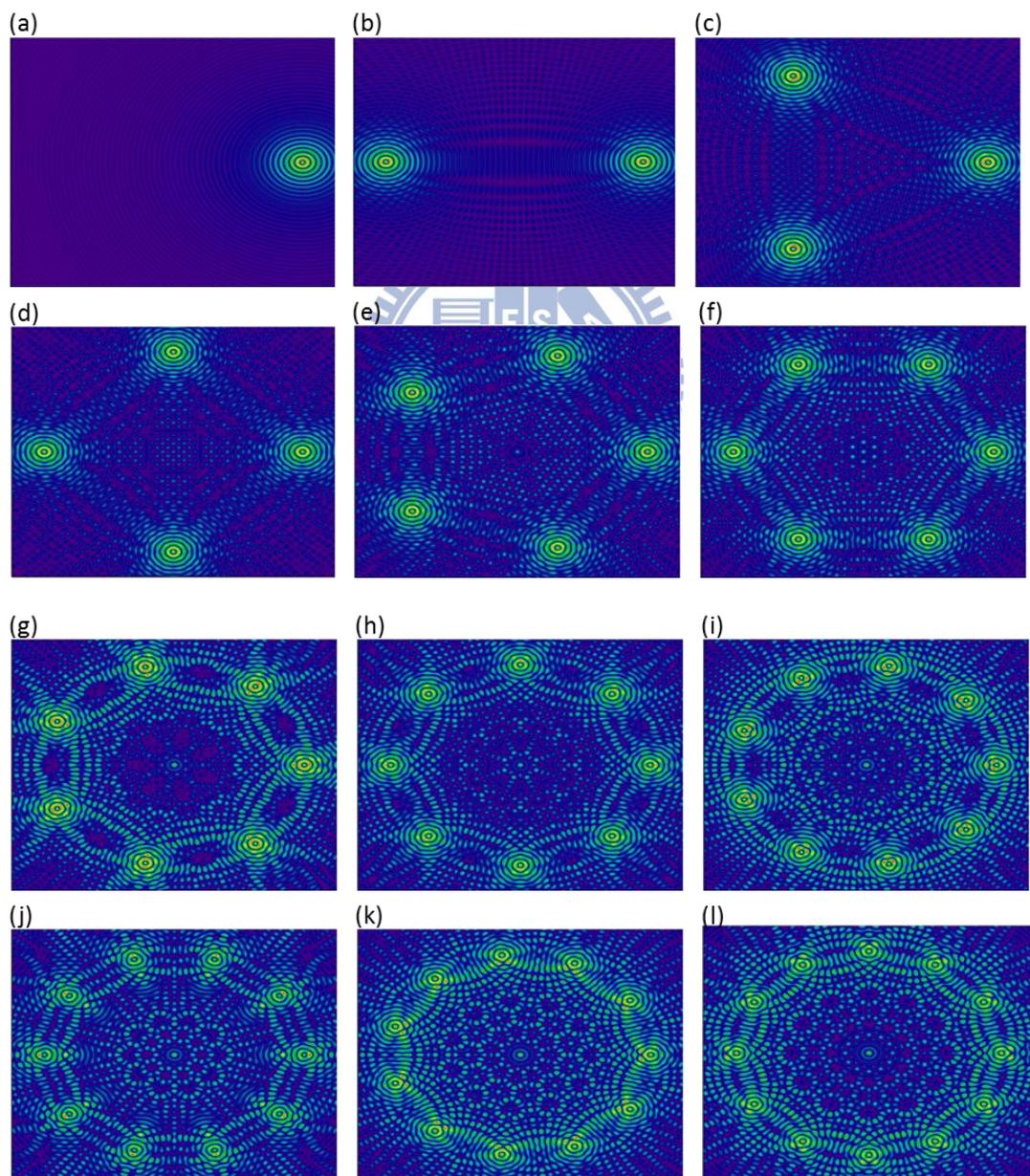


圖 3-3 不同數量點波圓干涉形成 Quasi-crystal 圖

3.4 波的變化

在前面小節中，所畫出的圖案都是波已經經過一段時間之後，彼此互相交疊所形成的圖騰，而無法看到時間變化使波產生的變化，因此我們為了可以依時間更細微的觀察波的干涉狀況，在製作波的圖案時，我們可以利用高斯分布如 3.4 式

$$y = e^{-\frac{(x-x_0)^2}{\Delta x}} \quad (3.4)$$

其中 $\Delta x > 0$ ，若是將 3.4 式中，令 $x_0 = 0$ ， $\Delta x = 5$ ，並繪製 x - y 平面高斯曲線圖，如圖 3-4

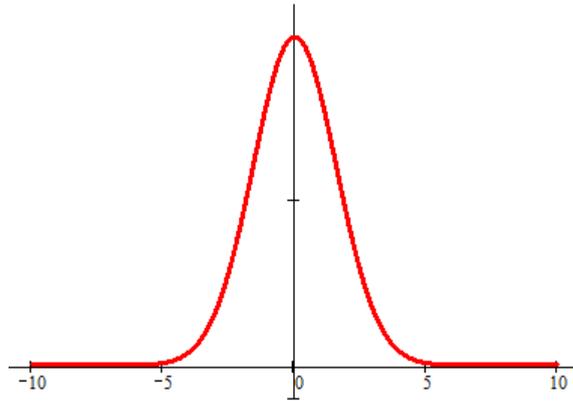


圖 3-4 高斯分布圖形

經由高斯分布曲線的圖形，我們可以發現，當 $x = x_0 = 0$ 時，高斯分布曲線有最大值為 1，並且在 $\pm\Delta x$ 之內，整個曲線的值衰退幅度非常快，而在 $\pm\Delta x$ 之外所有的值都趨近於 0；因此我們可以利用高斯分布曲線具有在 $x = x_0$ 有極值為 1 的集中性，與在 Δx 值內快速衰退以及在 $\pm\Delta x$ 之外的值都趨近於 0 的這些性質，控制我們原來所要呈現的波函數所產生的值，並經由逐漸擴大 Δx 之值，決定我們所要繪製波的範圍，將原來波的波紋範圍越擴越大，就如同剛丟進水池所產生的波紋，隨著時間越來越大一般，而波與波之間，也會因為時間的關係，

彼此逐漸產生干涉情況，如圖 3-5，令 $\Delta x = 2L^2$ ， $L = \{1,3,5,7\}$ 。

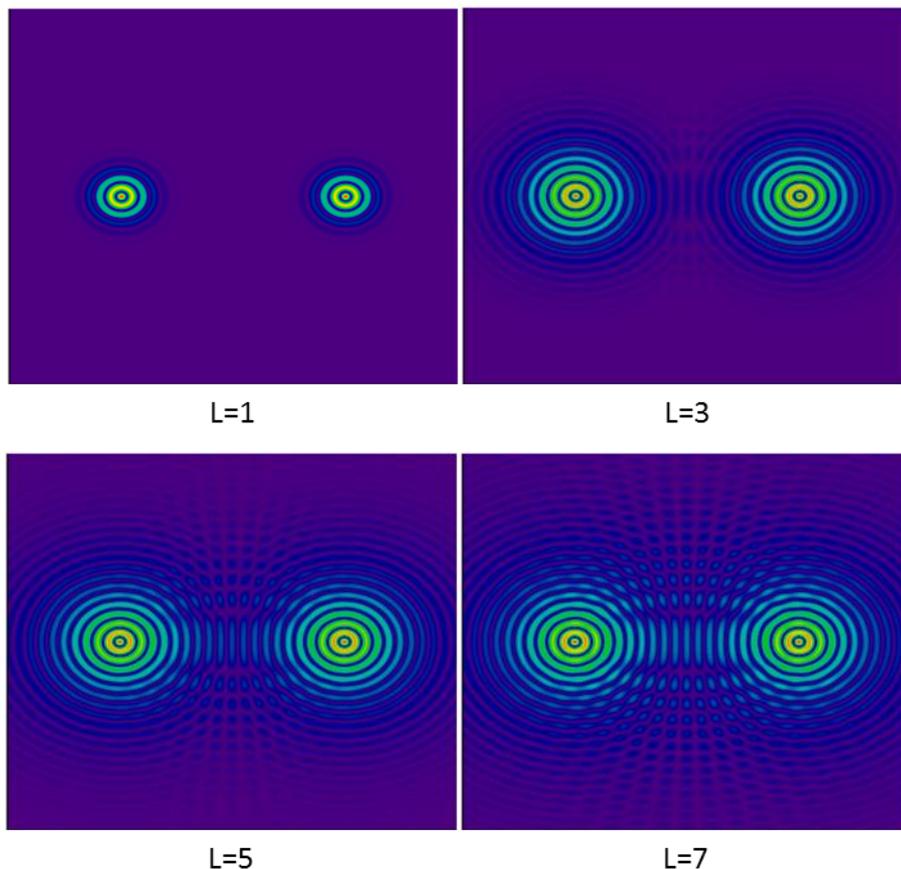


圖 3-5 利用高斯分布控制波函數干涉圖形

相關程式碼如附錄 3.2。

甚至我們可以藉由 mathcad 中的 FRAME 參數來製作成動畫的形式，並用功能列 view 中 Animate 的功能，將我們所要逐步呈現的波的圖形圈選起來，經由輸入所要畫出畫面的值，與每秒所要繪製的張數，經程式運算輸出之後，就會製作成類似影片的樣子，藉由這種方式，我們更可以由影片的畫面中的直接變化，看到波與波之間逐漸干涉的情形，使我們更容易依時間的變化看到波與波之間交疊改變的狀況，程式碼如附錄 3.3。

附錄 3.1

點波源干涉 MathCad 程式碼

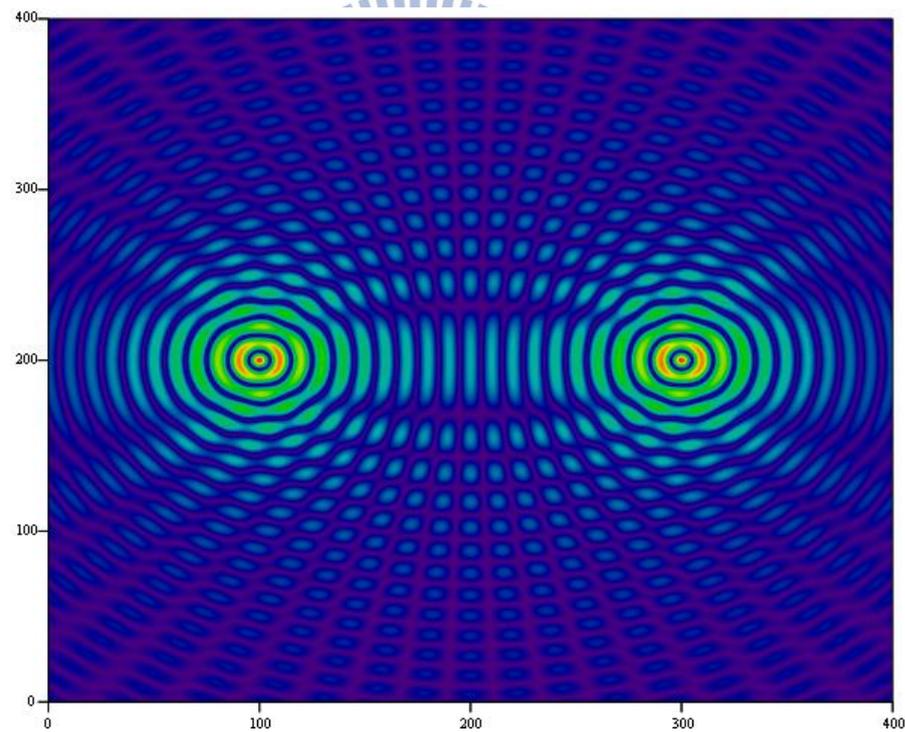
$$\lambda := 1 \quad x1 := -5\lambda \quad x2 := 5\lambda \quad d := 10\lambda$$

$$N := 400 \quad m := 0..N \quad n := 0..N$$

$$X_m := -d + 2\frac{d}{N} \cdot m \quad Y_n := -d + 2\frac{d}{N} \cdot n \quad L := 1\lambda$$

$I_{m,n}$

$$:= \left[\left[\frac{1}{\sqrt{(X_m - x1)^2 + (Y_n)^2 + L^2}} \cdot \cos \left[\frac{2 \cdot \pi}{\lambda} \cdot \sqrt{(X_m - x1)^2 + (Y_n)^2} \right] \right] \right. \\ \left. + \frac{1}{\sqrt{(X_m - x2)^2 + (Y_n)^2 + L^2}} \cdot \cos \left[\frac{2 \cdot \pi}{\lambda} \cdot \sqrt{(X_m - x2)^2 + (Y_n)^2} \right] \right] \left| \right|$$



I

附錄 3.2

波的干涉(時間變動) MathCad 程式碼

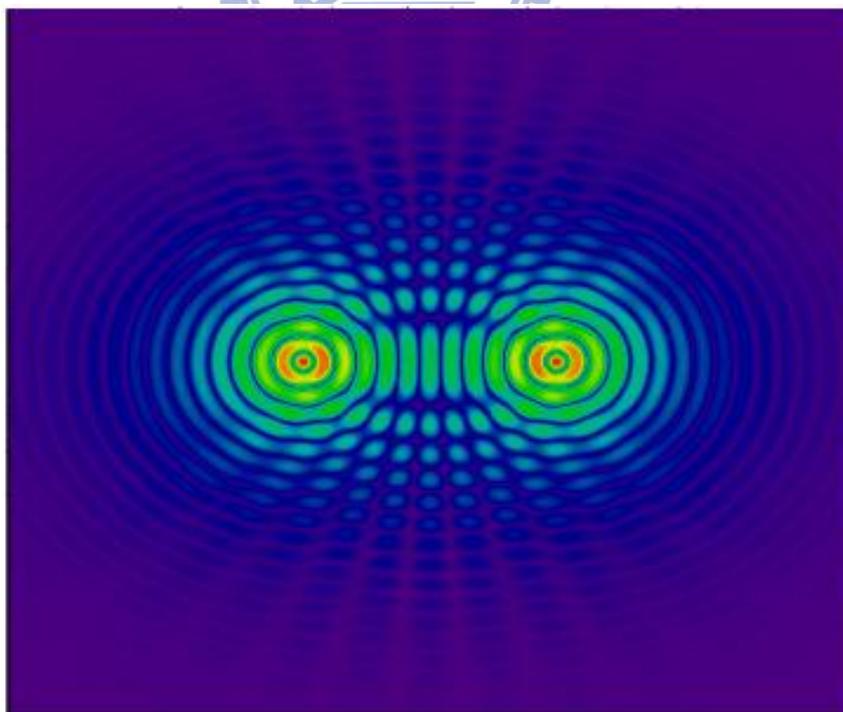
$$\lambda := 1 \quad x1 := -3\lambda \quad y1 := 0\lambda \quad x2 := 3\lambda \quad y2 := 0\lambda \quad d := 10\lambda$$

$$N := 250 \quad m := 0..N \quad n := 0..N$$

$$X_m := -d + 2\frac{d}{N} \cdot m \quad Y_n := -d + 2\frac{d}{N} \cdot n \quad L := 5\lambda \quad L_o := 1\lambda$$

$I_{m,n}$

$$:= \left[\begin{array}{l} \frac{1}{\sqrt{(X_m - x1)^2 + (Y_n - y1)^2 + L_o^2}} \cdot e^{-\left[\frac{((X_m - x1)^2 + (Y_n - y1)^2)}{2L^2}\right]} \cdot \cos\left[\frac{2 \cdot \pi}{\lambda} \cdot \sqrt{(X_m - x1)^2 + (Y_n - y1)^2}\right] \\ + \frac{1}{\sqrt{(X_m - x2)^2 + (Y_n - y2)^2 + L_o^2}} \cdot e^{-\left[\frac{((X_m - x2)^2 + (Y_n - y2)^2)}{2L^2}\right]} \cdot \cos\left[\frac{2 \cdot \pi}{\lambda} \cdot \sqrt{(X_m - x2)^2 + (Y_n - y2)^2}\right] \end{array} \right]^a$$



I

附錄 3.3

波的干涉(隨時間的變動變化) MathCad

程式碼

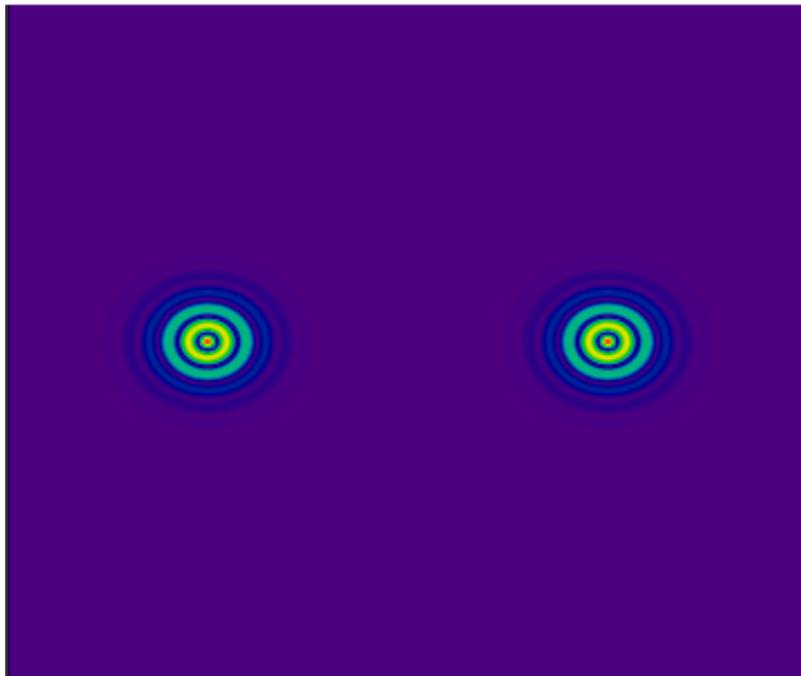
$\lambda := 1$ $x1 := -3\lambda$ $y1 := 0\lambda$ $x2 := 3\lambda$ $y2 := 0\lambda$ $d := 10\lambda$

$N := 250$ $m := 0..N$ $n := 0..N$ $Lo := 1\lambda$

$X_m := -d + 2\frac{d}{N} \cdot m$ $Y_n := -d + 2\frac{d}{N} \cdot n$ $L := 1 + \text{FRAME}$

$I_{m,n}$

$$:= \left[\begin{array}{l} \frac{1}{\sqrt{(X_m - x1)^2 + (Y_n - y1)^2 + Lo^2}} \cdot e^{-\left[\frac{((X_m - x1)^2 + (Y_n - y1)^2)}{2L^2}\right]} \cdot \cos\left[\frac{2 \cdot \pi}{\lambda} \cdot \sqrt{(X_m - x1)^2 + (Y_n - y1)^2}\right] \\ + \frac{1}{\sqrt{(X_m - x2)^2 + (Y_n - y2)^2 + Lo^2}} \cdot e^{-\left[\frac{((X_m - x2)^2 + (Y_n - y2)^2)}{2L^2}\right]} \cdot \cos\left[\frac{2 \cdot \pi}{\lambda} \cdot \sqrt{(X_m - x2)^2 + (Y_n - y2)^2}\right] \end{array} \right]^a$$



第四章 彈簧擺的視覺化

我們所彈簧擺為研究力學現象所產生的運動方式的一種方法，可以利用觀察彈簧擺的運動方式，來觀察擺動現象是否為混沌，還是有規律地擺動，甚至形成共振現象；彈簧擺(spring-pendulum)，依名稱所示可以知道就是在一個單擺上連結一個彈簧，如圖 4-1 所示，



圖 4-1 彈簧擺示意圖

懸掛一個彈簧，在彈簧的末端懸掛一個彈簧，並在彈簧的末端繫上一個質量為 m 的物體，這就構成一個彈簧擺，這是只要輕輕地提供一個方向的力量，彈簧就會拉著因重力下垂的物體擺盪震動起來，如圖 4-2 所示[15]。

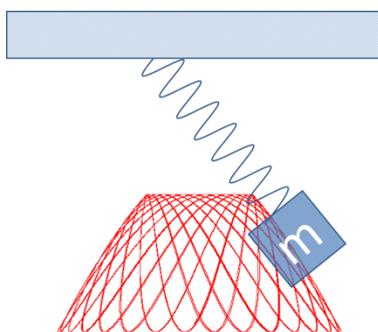


圖 4-2 彈簧擺擺動示意圖

在研究彈簧擺之前，我們先從一般的單擺(pendulum)看起，在一般的單擺中，是將一條長度為 l 線段的末端，懸掛一個質量為 m 的物體，如圖 4-3 所示，則這個物體將會受到重力加速度 g 的影響，擺動的頻率為 $w_p = \sqrt{\frac{g}{l}}$

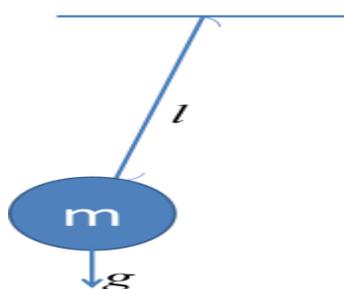


圖 4-3 單擺受力示意圖

在一般的彈簧運動中，是在一個彈簧的末端，放置一個質量為 m 的物體，如圖 4-4 所示，如圖其彈簧係數為 k ，則彈簧的震動幅度為

$w_s = \sqrt{\frac{k}{m}}$ ，將彈簧與單擺結合之後就成為了彈簧擺。



圖 4-4 彈簧運動示意圖

4.1 二維空間彈簧擺

在彈簧擺上，因為懸掛物體，會使彈簧原始的長度 l_0 因為重力而拉長 $\frac{mg}{k}$ ，令 $l_e = l_0 + \frac{mg}{k}$ ，若是在 x - y 平面中表示相關的位置，如圖4-5所示[16]，

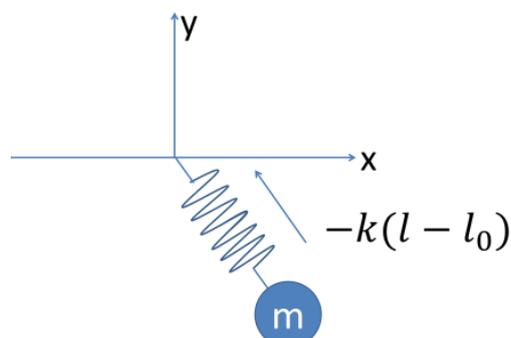


圖 4-5 彈簧擺二維平面示意圖

令 $l = \sqrt{x^2 + y^2}$ ，則可以知道彈簧擺所受到來自單擺的力為 $F_s = -k(\sqrt{x^2 + y^2} - l_0)$ ，若加入考慮方向性，則

$$\begin{aligned}\vec{F}_s &= -k(\sqrt{x^2 + y^2} - l_0) \left(\frac{x\hat{a}_x + y\hat{a}_y}{\sqrt{x^2 + y^2}} \right) \\ &= -k \left(1 - \frac{l_0}{\sqrt{x^2 + y^2}} \right) (x\hat{a}_x + y\hat{a}_y)\end{aligned}\quad (4.1)$$

彈簧擺中來自彈所施予的力則為重力所施予的部分 $\vec{F}_g = -mg\hat{a}_y$

又有關運動的方程式為

$$\vec{F} = m\vec{a} = m \frac{d^2x}{dt^2} \hat{a}_x + m \frac{d^2y}{dt^2} \hat{a}_y \quad (4.2)$$

經移項整理

$$\hat{x}: \frac{d^2x}{dt^2} = -\frac{kx}{m} \left(1 - \frac{l_0}{\sqrt{x^2+y^2}}\right) \quad (4.3)$$

$$\hat{y}: \frac{d^2y}{dt^2} = -g - \frac{ky}{m} \left(1 - \frac{l_0}{\sqrt{x^2+y^2}}\right) \quad (4.4)$$

4.3 式與 4.4 式皆為耦合方程式，在任何一個 x, y 方程式中，都含有 x, y 的關係式，所以我們利用有限差分法，來分析這個方程式

$$\left(\frac{dx}{dt}\right) \Big|_i = \frac{x_i - x_{i-1}}{\Delta t} \quad (4.5)$$

$$\left(\frac{d^2x}{dt^2}\right) \Big|_i = \frac{\left(\frac{dx}{dt}\right)_i - \left(\frac{dx}{dt}\right)_{i-1}}{\Delta t} = \frac{x_i - 2x_{i-1} + x_{i-2}}{(\Delta t)^2} \quad (4.6)$$

為了配合疊代的程式，將下標移項，

$$\frac{x_i - 2x_{i-1} + x_{i-2}}{(\Delta t)^2} = \frac{x_{i+1} - 2x_i + x_{i-1}}{(\Delta t)^2} \quad (4.7)$$

則

$$\left(\frac{d^2x}{dt^2}\right) \Big|_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{(\Delta t)^2} \quad (4.8)$$

經移項整理之後得到，

$$x_{i+1} = \left[-\frac{kx_i}{m} \left(1 - \frac{l_0}{\sqrt{x^2+y^2}}\right) \right] (\Delta t)^2 + 2x_i - x_{i-1} \quad (4.9)$$

同理，

$$y_{i+1} = \left[-g - \frac{ky_i}{m} \left(1 - \frac{l_0}{\sqrt{x^2 + y^2}} \right) \right] (\Delta t)^2 + 2y_i - y_{i-1} \quad (4.10)$$

當我們給彈簧擺一個初始條件之後，就會使彈簧擺產生運動。令彈簧擺的初始位置為 x_0 ， y_0 以及給予一個 x 方向的初速度 v_{x_0} ， y 方向的初速度 v_{y_0} ，則下一個位置的方程式為

$$\begin{aligned} x_1 &= x_0 + v_{x_0} \cdot \Delta t \\ y_1 &= y_0 + v_{y_0} \cdot \Delta t \end{aligned} \quad (4.11)$$

由上面的式子，我們可以歸納出總能量為位能和動能的總和，

$$E = mgy + \frac{1}{2}k \left(\sqrt{x^2 + y^2} - l_0 \right)^2 + \frac{1}{2}mv_x^2 + \frac{1}{2}mv_y^2 \quad (4.12)$$

而當彈簧擺靜止不動時，具有最小的能量

$$E_{min} = -mg \left(l_0 + \frac{mg}{k} \right) y + \frac{1}{2}k \left(\frac{mg}{k} \right)^2 = -mg \left(l_0 + \frac{1}{2} \cdot \frac{mg}{k} \right) \quad (4.13)$$

在繪製圖形之前，我們先定義能量比值(energy ratio) R ，

$$R \equiv -\frac{E}{E_{min}} \quad (4.14)$$

由整理 4.14 式我們可以得到

$$R = \frac{(-m \cdot g \cdot l_0) + \frac{1}{2} \cdot m \cdot v_{x_0}^2}{m \cdot g \cdot \left(l_0 + \frac{1}{2} \frac{m \cdot g}{k} \right)} \quad (4.15)$$

因此當我們決定 x 方向的初始速度，就會決定 R 的大小，同時當彈簧擺的擺動是處於小振幅時，令 μ 為單擺和彈簧之間的頻率平方比，

$$\mu \equiv \frac{w_s^2}{w_p^2} \quad (4.16)$$

或是另一個呈現 μ 的關係式為

$$\mu = 1 + \frac{k \cdot l_0}{mg} \quad (4.17)$$

經移項整理

$$k = (\mu - 1) \frac{mg}{l_0} \quad (4.18)$$

由 4.18 式我們可以得到 k 和 μ 的關係式，因此在繪製圖形時，我們只要先決定了係數 μ 的值，即得到相對應的 k 的係數(彈簧的粗細、硬度)。下面的程式碼中，首先先給予初始條件包含所要灑的點數 N ，初始位置 x_0, y_0 ，初始速度 v_{x_0}, v_{y_0} ，頻率平方比 μ ，懸掛物體質量 m ，重力 g ，彈簧原始長度 l_0 ，繪製的時間長度 T ，接下來我們使用另一種程式的編寫方式，首先先將 x, y 之值都輸入於矩陣 z 之中，在程式計算完之後再取出相對應的 x, y 值並繪圖，如此即可產生我們要的彈簧擺動變化，例如給定 $N=400000, g=9.8, l_0=1, m=0.02, T=300, x_0=0, y_0=-l_0, v_{y_0}=0$ ，只改變 μ 和 v_{x_0} 的圖形如圖 4-6，圖 4-7，相關程式碼如附錄 4.1。

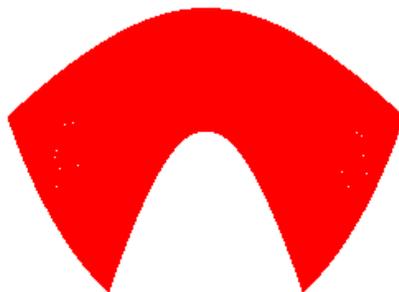


圖 4-6 彈簧擺動軌跡($\mu=4, v_{x_0}=1$)

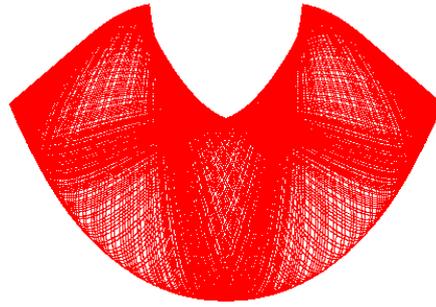


圖 4-7 彈簧擺擺動軌跡($u=7, v_{x_0}=3.5432$)

在繪製彈簧擺的擺動路徑時，由於灑點的因素，會使圖形較為不清晰，因此我們可以同樣的利用取等值線的方式，將圖形變得更清晰，使我們的觀察可以更容易呈現，同時也可以呈現出不同的顏色變化，讓圖形更漂亮，如圖 4-8。

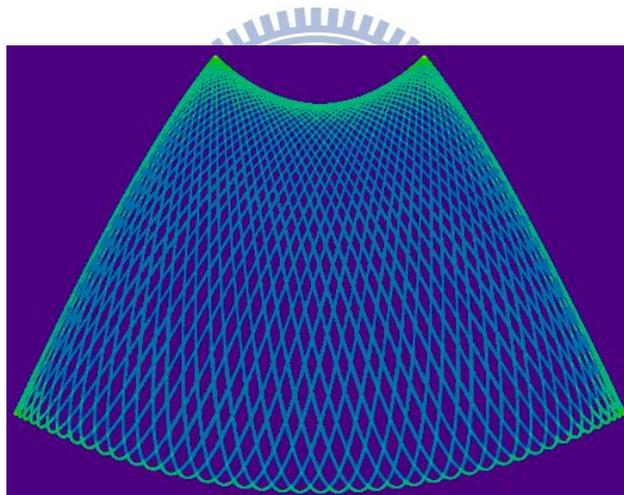


圖 4-8 彈簧擺擺動軌跡等值線圖($u=6, v_{x_0}=1.1001$)

4.2 三維空間彈簧擺

彈簧擺運動不單只是平面的擺動，我們可以藉由加入 z 方向的考量，將 x, y 平面圖形變成三維的空間，將原來的 x 方向推廣 x, y 平面，而原來的 y 方向，變成 z 方向，當中只需要將變數 x, y, z 三個方向的位

置，都會隨著初始條件做一些改變即可，如圖 4-9。

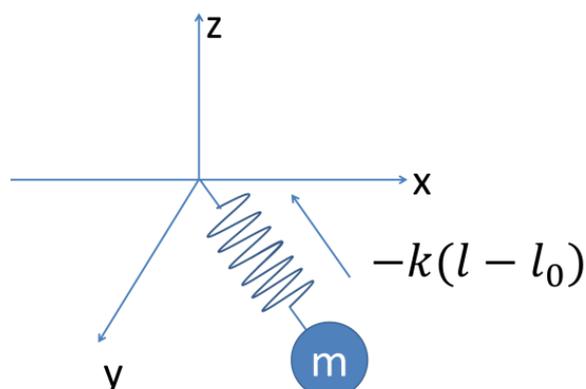


圖 4-9 彈簧擺三維空間示意圖

令 $l = \sqrt{x^2 + y^2 + z^2}$ ，則可以知道彈簧擺所受到來自單擺的力為 $F_s = -k(\sqrt{x^2 + y^2 + z^2} - l_0)$ ，若加入考慮方向性，則

$$\begin{aligned}\vec{F}_s &= -k \left(\sqrt{x^2 + y^2 + z^2} - l_0 \right) \left(\frac{x\hat{a}_x + y\hat{a}_y + z\hat{a}_z}{\sqrt{x^2 + y^2 + z^2}} \right) \\ &= -k \left(1 - \frac{l_0}{\sqrt{x^2 + y^2 + z^2}} \right) (x\hat{a}_x + y\hat{a}_y + z\hat{a}_z)\end{aligned}$$

彈簧擺中來自彈所施予的力則為重力所施予的部分

$$\vec{F}_g = -mg\hat{a}_z \quad (4.19)$$

有關運動的方程式為

$$\vec{F} = m\vec{a} = m \frac{d^2x}{dt^2} \hat{a}_x + m \frac{d^2y}{dt^2} \hat{a}_y + m \frac{d^2z}{dt^2} \hat{a}_z \quad (4.20)$$

$$\hat{x}: \frac{d^2x}{dt^2} = -\frac{kx}{m} \left(1 - \frac{l_0}{\sqrt{x^2 + y^2 + z^2}} \right) \quad (4.21)$$

$$\hat{y}: \frac{d^2 y}{dt^2} = -\frac{ky}{m} \left(1 - \frac{l_0}{\sqrt{x^2 + y^2 + z^2}} \right) \quad (4.22)$$

$$\hat{z}: \frac{d^2 z}{dt^2} = -g - \frac{ky}{m} \left(1 - \frac{l_0}{\sqrt{x^2 + y^2 + z^2}} \right) \quad (4.23)$$

4.21 式到 4.23 式為耦合方程式，我們利用有限差分法與移項整理，來分析這個 4.21 式到 4.23 式方程式，我們可以得到

$$\begin{aligned} x_{i+1} &= \left[-\frac{kx_i}{m} \left(1 - \frac{l_0}{\sqrt{x^2 + y^2 + z^2}} \right) \right] (\Delta t)^2 + 2x_i - x_{i-1} \\ y_{i+1} &= \left[-\frac{ky_i}{m} \left(1 - \frac{l_0}{\sqrt{x^2 + y^2 + z^2}} \right) \right] (\Delta t)^2 + 2y_i - y_{i-1} \\ z_{i+1} &= \left[-g - \frac{kz_i}{m} \left(1 - \frac{l_0}{\sqrt{x^2 + y^2 + z^2}} \right) \right] (\Delta t)^2 + 2z_i - z_{i-1} \quad (4.24) \end{aligned}$$

若是我們給彈簧擺一個初始條件之後，令彈簧擺的初始位置為 x_0 ， y_0 ， z_0 以及給予一個 x 方向的初速度 v_{x_0} ， y 方向的初速度 v_{y_0} ， z 方向的初速度 v_{z_0} 則下一個位置的方程式為

$$\begin{aligned} x_1 &= x_0 + v_{x_0} \cdot \Delta t \\ y_1 &= y_0 + v_{y_0} \cdot \Delta t \\ z_1 &= z_0 + v_{z_0} \cdot \Delta t \end{aligned} \quad (4.25)$$

由 4.25 式，我們可以歸納出彈簧擺在擺動的過程中的總能量為位能和動能的總和為 4.26 式，

$$E = mgz + \frac{1}{2}k \left(\sqrt{x^2 + y^2 + z^2} - l_0 \right)^2 + \frac{1}{2}mv_x^2 + \frac{1}{2}mv_y^2 + \frac{1}{2}mv_z^2 \quad (4.26)$$

而當彈簧擺靜止不動時，具有最小的能量

$$\begin{aligned} E_{min} &= -mg \left(l_0 + \frac{mg}{k} \right) z + \frac{1}{2}k \left(\frac{mg}{k} \right)^2 \\ &= -mg \left(l_0 + \frac{1}{2} \cdot \frac{mg}{k} \right) \end{aligned} \quad (4.27)$$

而能量比值 R ，也和之前二維平面相同，[16]

$$R \equiv -\frac{E}{E_{min}} \quad (4.28)$$

由 4.28 式我們可以整理得到

$$R = \frac{(-m \cdot g \cdot l_0) + \frac{1}{2} \cdot m \cdot v_{x_0}^2 + \frac{1}{2} \cdot m \cdot v_{y_0}^2}{m \cdot g \cdot \left(l_0 + \frac{1}{2} \cdot \frac{m \cdot g}{k} \right)} \quad (4.29)$$

因此當我們決定 x, y 方向的初始速度，就會決定 R 的大小，而頻率平方比 μ ，也和二維空間相同，所以當我們分別在不同的初始條件之下，就可以繪製出許多彈簧擺的路徑圖，我們給予 $N = 2^{19}$ ， $m = 0.02, l_0 = 1, g = 9.8, x_0 = 0.01, y_0 = 0.01, z_0 = -l_0, v_{y_0} = 0, v_{z_0} = 0, T = 400, \mu = \{3, 10, 22, 50, 500\}$ ， $v_{x_0} = \{0.41243, 4.31243, 5.41243, 40.41243\}$ ，即可繪出許多彈簧擺的擺盪路徑圖形，如圖 4-10(a)，(b)，(c)，(d)，(e)，(f)，相關程式碼如附錄 4.2。

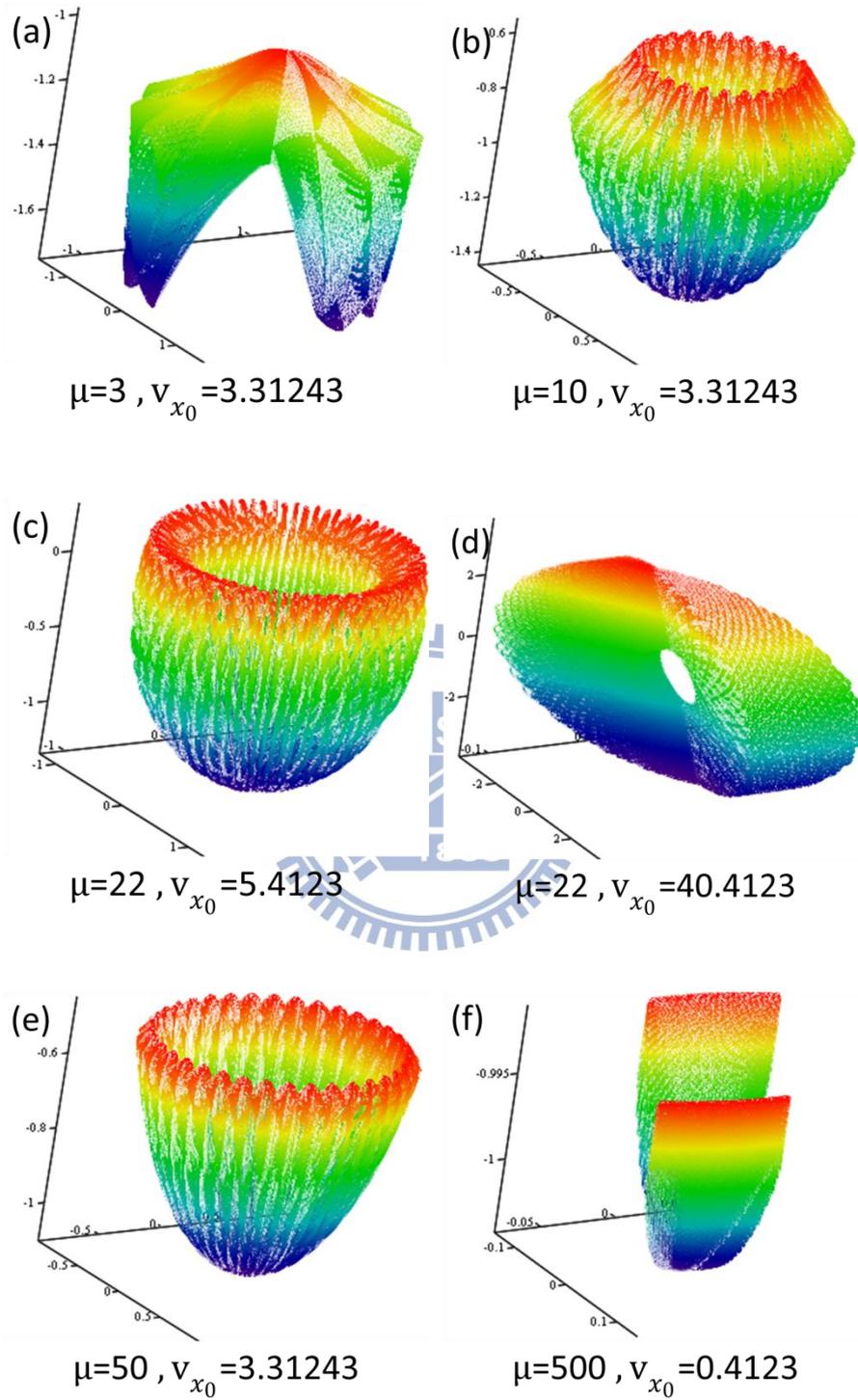


圖 4-10 不同初始條件下的彈簧擺軌跡

4.3 傅立葉轉換

傅立葉變換在物理學、聲學、光學、結構動力學、數論、組合數學、機率論、統計學、信號處理、密碼學、海洋學、通訊等領域都有著廣泛的應用。例如在信號處理中，傅立葉變換的典型用途是將信號分解成振幅分量和頻率分量，因此傅立葉轉換，是一種分析的工具，我們可以利用傅立葉轉換來看任何一個訊號內究竟有什麼樣的訊號。

首先我們來看離散型的傅立葉轉換(DFT, Discrete time Fourier Transform)。假設我們有一個訊號 $X(n)$ ，將它乘上 $e^{-j\omega n}$ ，將它從 $n = -\infty$ 到 $+\infty$ 加起來，用 $X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$ 來表示它的結果，它就是一個連續變數 ω 的函數，同時我們也可以從理論證明 $x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega$ ，而 $X(e^{j\omega})$ 就稱為 $x(n)$ 的離散型傅立葉轉換[17]。

早期的傅利葉轉換大都利用手算來完成，因此用傅利葉轉換的數值運算數量都很少。但在 1865 年 Gauss 就提出有關於如何更快的計算傅利葉轉換的演算技巧，而快速傅利葉轉換的演算法真正開始受到重視是在 1965 年 Cooley 及 Tukey 發表一系列快速傅利葉轉換的論文。現代由於電腦的計算能力加強，傅利葉轉換的數值數量可以大量的增加，於是大家開始關心如何使用更快的計算方式來節省計算機的時間，因此我們將使用快速傅利葉轉換(FFT, Fast Fourier Transform)來協助我們計算。當我們以傳統的計算方式進行 DFT 運算時，對一個 N 點的序列而言，DFT 運算需要 N^2 的計算量，但是如果使用快速傅立葉轉換(FFT)時，我們則只需要 $N \cdot \log_2 N$ 個計算量，當 N 很大時，

這樣節省下來的計算時間是相當可觀的。

我們也可利用 DFT 來求得信號的頻譜。然而 DFT 的缺點為計算量過於龐大。FFT 則是以 DFT 為基礎，大量節省計算量，也因此 FFT 沿襲著 DFT 之限制條件，並且增加了 FFT 的限制條件，而這些 FFT 的限制條件為：

(1) 受測信號須為週期性。(2) 取樣數須為 $2r$ 個資料[18]。

因此在我們執行程式運算時，要注意上面的兩個 FFT 的條件。

在觀察彈簧擺的軌跡與運動方式時，在某些條件是處於較簡單而有規律的線條呈現，而在某些條件下所呈現的圖形則較為混亂，甚至出現混沌的情況，此時我們可以利用 mathcad 中傅立葉轉換的參數 `fft`，藉由這個參數，我們可以將原來的 x, y, z 方向的頻率分別用傅立葉轉換，特徵化，若是屬於有規律的圖形，我們所得到的傅立葉轉換會比較少的特徵值，若是出現許多高低不同的特徵值在某一個方向軸裡，則就是很明顯的渾沌現象[19]，其中一般的情形(非混沌)如圖 4-11

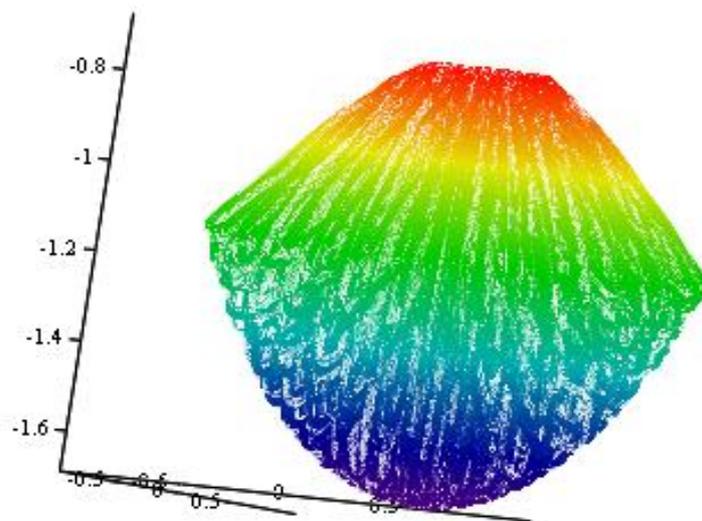


圖 4-11 非混沌彈簧擺軌跡

將原來的 x, z 方向的頻率經快速傅立葉轉換(FFT)後數值如圖 4-12 所示

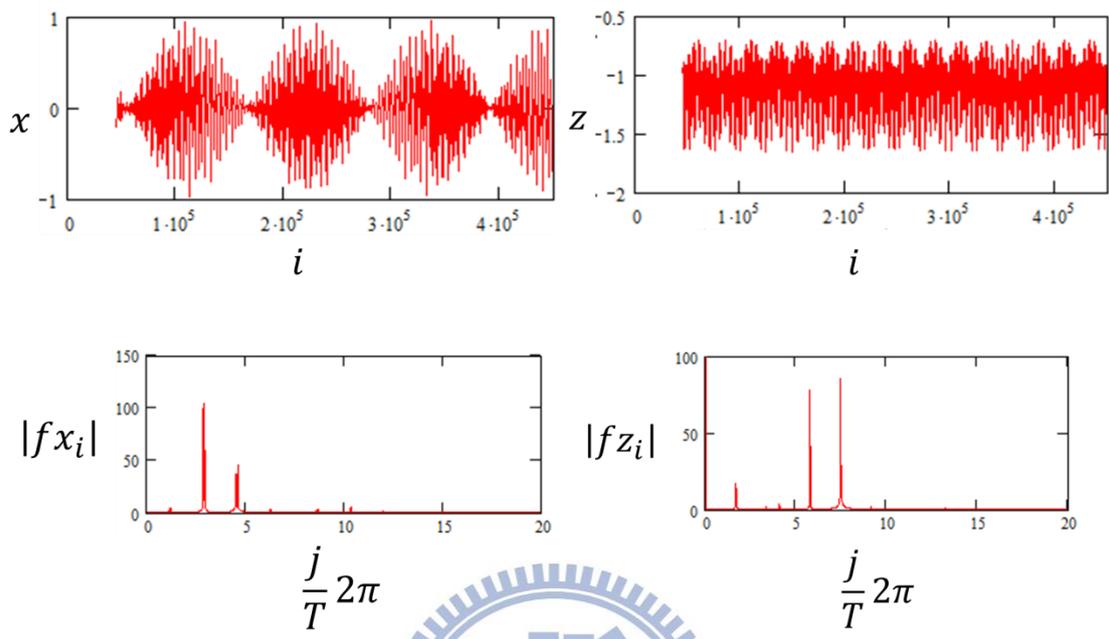


圖 4-12 快速傅立葉轉換圖 4-11 的 x, z 方向結果

混沌的情形，如圖 4-13

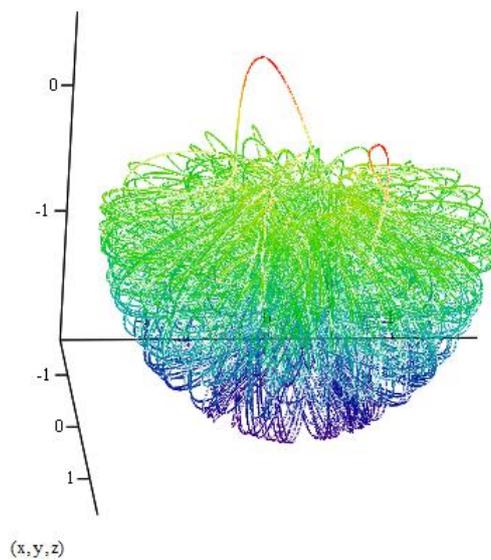


圖 4-13 混沌彈簧擺軌跡

將原來的 x, z 方向的頻率經快速傅立葉轉換(FFT)後數值如圖 4-14 所示

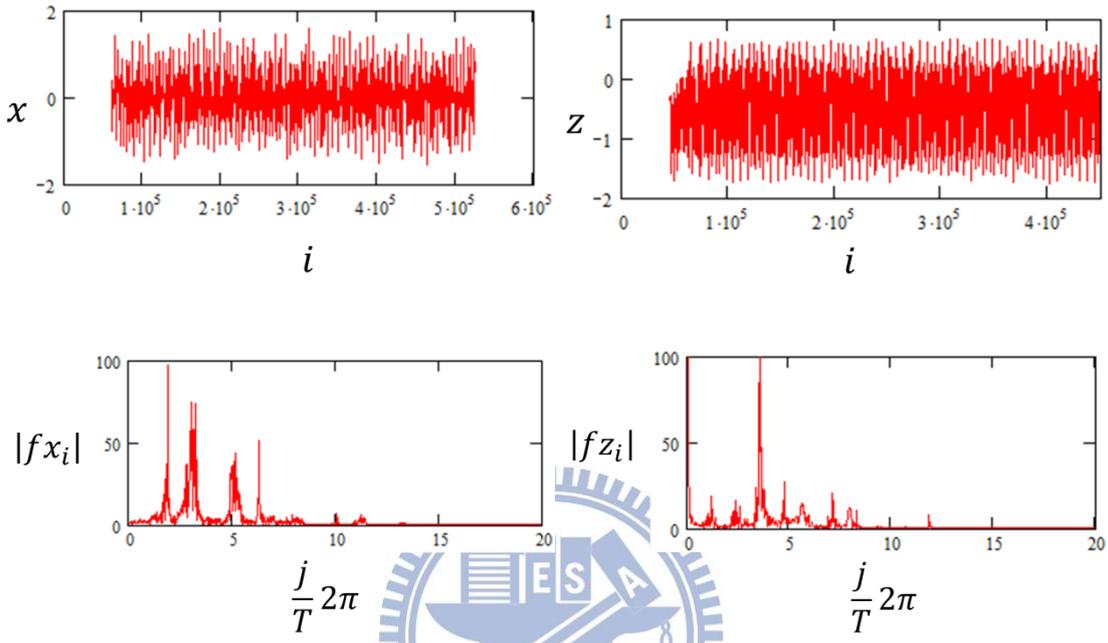


圖 4-14 快速傅立葉轉換圖 4-13 的 x, z 方向結果

經由觀察上面的傅立葉轉換的圖形可以發現到，當彈簧擺的圖形經傅立葉轉換後呈現的特徵值，若是較為混亂且多峰的，則原來的圖形結果所出現的也是較為混亂的。

在程式碼的部分，要加入的程式碼只要在之前的程式碼之後鍵入 $fx := \text{fft}(x)$ ，並繪製出相對應的圖形，就可以看到 x 方向的傅立葉轉換特徵值，同理 $fz := \text{fft}(z)$ ，並繪製出來，可以看到 z 方向特徵值的樣子；經由特徵值的判斷，我們可以更容易地去判斷初始條件是否會導致混沌現象的產生。

附錄 4.1

二維彈簧擺 MathCad 程式碼

$$\begin{aligned}
 N &:= 400000 & g &:= 9.8 & L_0 &:= 1 & m &:= 0.02 & \mu &:= 6 \\
 T &:= 300 & x_0 &:= 0 & y_0 &:= -L_0 & v_{x0} &:= 1.1001 & v_{y0} &:= 0
 \end{aligned}$$

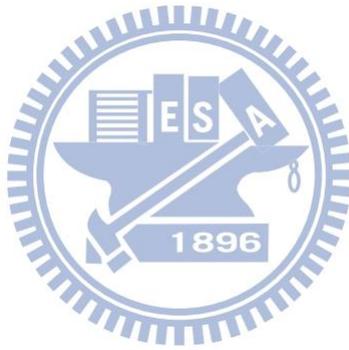
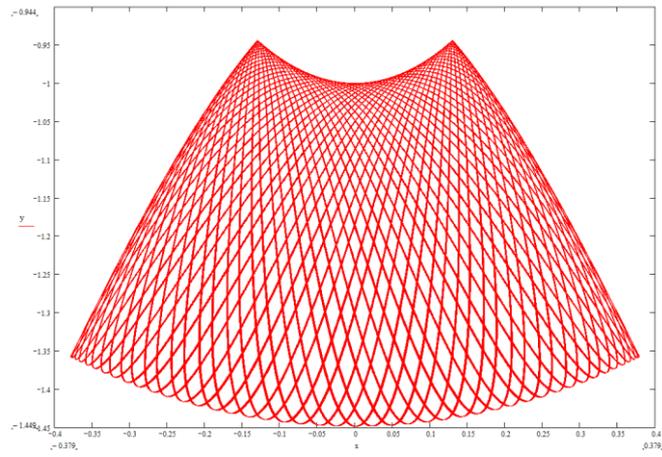
$$k := (\mu - 1) \frac{mg}{L_0} \quad R := \frac{(-m \cdot g \cdot L_0) + \frac{1}{2} m \cdot v_{x0}^2}{m \cdot g \cdot \left(L_0 + \frac{1}{2} \frac{m \cdot g}{k} \right)}$$

$$\text{CP}(x_0, y_0, v_{x0}, v_{y0}, T, N) :=$$

$$\begin{aligned}
 & \Delta t \leftarrow \frac{T}{N} \\
 & x_0 \leftarrow x_0 \\
 & y_0 \leftarrow y_0 \\
 & x_1 \leftarrow x_0 + v_{x0} \cdot \Delta t \\
 & y_1 \leftarrow y_0 + v_{y0} \cdot \Delta t \\
 & \text{for } j \in 1..N-1 \\
 := & \left[\begin{array}{l}
 x_{j+1} \leftarrow \left[\left[-\frac{k}{m} \cdot \left[1 - \frac{L_0}{\sqrt{(x_j)^2 + (y_j)^2}} \right] \cdot x_j \right] \cdot \Delta t^2 \right] + 2 \cdot x_j - x_{j-1} \\
 y_{j+1} \leftarrow \left[\left[-g - \frac{k}{m} \cdot \left[1 - \frac{L_0}{\sqrt{(x_j)^2 + (y_j)^2}} \right] \cdot y_j \right] \cdot \Delta t^2 \right] + 2 \cdot y_j - y_{j-1}
 \end{array} \right] \\
 & \text{augment}(x, y)
 \end{aligned}$$

$$z := \text{CP}(x_0, y_0, v_{x0}, v_{y0}, T, N)$$

$$x := z^{<0>} \quad y := z^{<1>} \quad i := 0..N$$



附錄 4.2

彈簧擺三維擺動 MathCad 程式碼

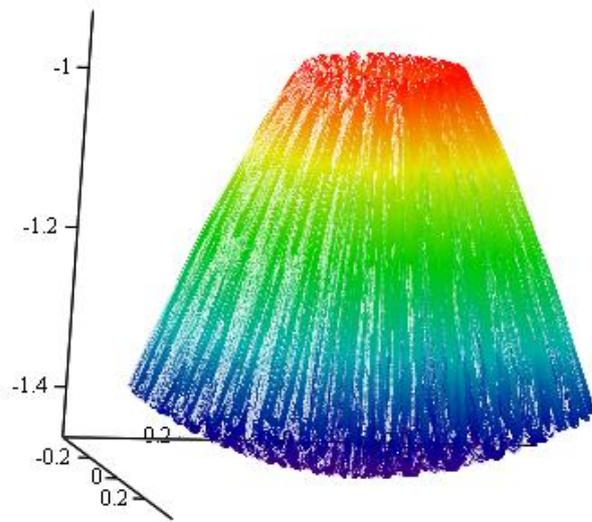
$$N := 2^{19} \quad g := 9.8 \quad L_0 := 1 \quad m := 0.02 \quad u := 6 \quad T := 400$$

$$x_0 := 0 \quad z_0 := -L_0 \quad y_0 := 0.01 \quad v_{x0} := 1.101 \quad v_{y0} := 0$$

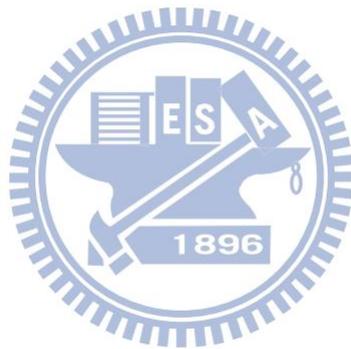
$$v_{z0} := 0 \quad k := (u - 1) \frac{m \cdot g}{L_0} \quad R := \frac{(-m \cdot g \cdot L_0) + \frac{1}{2} \cdot m \cdot v_{x0}^2}{m \cdot g \cdot \left(L_0 + \frac{1}{2} \cdot \frac{m \cdot g}{k} \right)}$$

$$CP(x_0, y_0, z_0, v_{x0}, v_{y0}, v_{z0}, T, N) :=$$

$$\left| \begin{array}{l} \Delta t \leftarrow \frac{T}{N} \\ x_0 \leftarrow x_0 \\ y_0 \leftarrow y_0 \\ z_0 \leftarrow z_0 \\ x_1 \leftarrow x_0 + v_{x0} \cdot \Delta t \\ y_1 \leftarrow y_0 + v_{y0} \cdot \Delta t \\ z_1 \leftarrow z_0 + v_{z0} \cdot \Delta t \\ \text{for } j \in 1..N - 2 \\ x_{j+1} \leftarrow \left[\left[\frac{-k}{m} \cdot \left[1 - \frac{L_0}{\sqrt{(x_j)^2 + (y_j)^2 + (z_j)^2}} \right] \cdot x_j \right] \cdot \Delta t^2 \right] + 2 \cdot x_j - x_{j-1} \\ y_{j+1} \leftarrow \left[\left[\frac{-k}{m} \cdot \left[1 - \frac{L_0}{\sqrt{(x_j)^2 + (y_j)^2 + (z_j)^2}} \right] \cdot y_j \right] \cdot \Delta t^2 \right] + 2 \cdot y_j - y_{j-1} \\ z_{j+1} \leftarrow \left[\left[-g - \frac{k}{m} \cdot \left[1 - \frac{L_0}{\sqrt{(x_j)^2 + (y_j)^2 + (z_j)^2}} \right] \cdot z_j \right] \cdot \Delta t^2 \right] + 2 \cdot z_j - z_{j-1} \\ \text{augment}(x, y, z) \end{array} \right.$$



(x,y,z)



第五章 Knot & Torus 的視覺化

在考古學家研究中打結現象是自史前時代就開始的，無論中西方都有利用繩子打結的生活經驗，在打結之前，我們要先從基本的環面看起，在數學幾何上，一個環面，由一個圓和該圓的共生軸迴旋旋轉而成。我們可以給這個環面一個參數式(5.1)[20]，

$$\begin{cases} x(u, v) = (d + a \cdot \cos v) \cdot \cos u \\ y(u, v) = (d + a \cdot \cos v) \cdot \sin u \\ z(u, v) = a \cdot \sin v \end{cases} \quad (5.1)$$

其中 $u, v \in [0, 2\pi]$ ， d 是管子中心(圓面的圓心)到輪胎中心的距離， a 為圓管(圓環)的半徑，藉由繩結在圓圈上旋轉交疊就可以形成相關的 knot 圖形，如圖 5-1，

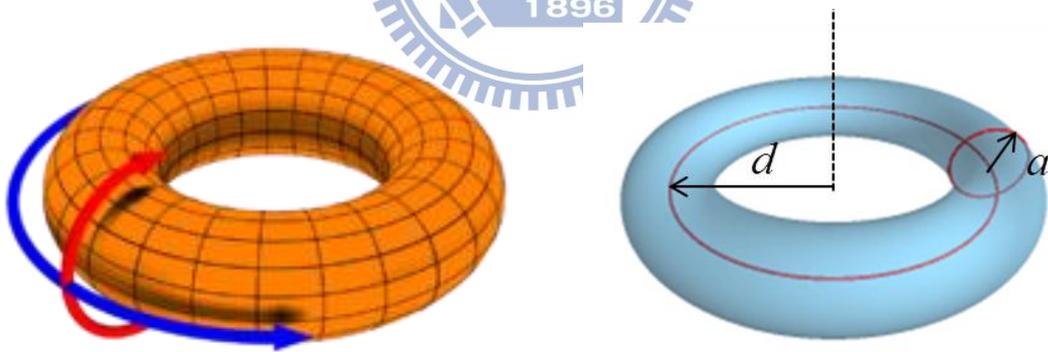


圖 5-1 Torus 形成示意圖

而球面可以視為一個環面的特殊情況，也就是旋轉軸是該圓的直徑， $d = 0$ ，如圖 5-2，

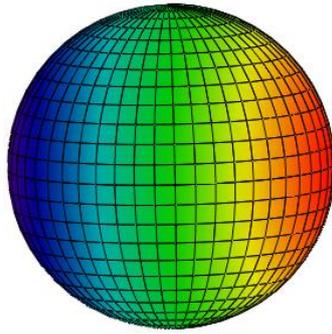


圖 5-2 Torus 旋轉軸 $d = 0$ 示意圖

若轉軸和圓面不相交，則圓面的中心會形成一個洞，就像一個甜甜圈的樣子，或是一個輪胎充飽了氣一樣，如圖 5-3

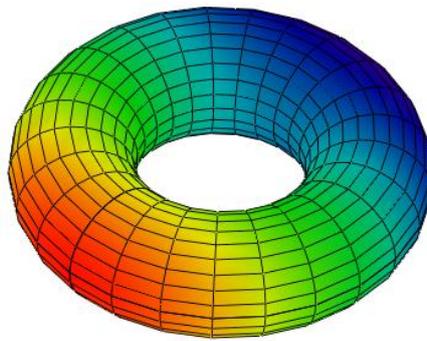


圖 5-3 Torus 示意圖

5.1 Torus Knot

想像我們用一條繩子在 Torus 上旋轉纏繞，所產生的結果就會是環面扭結(Torus Knot)，而方程式如(5.2)：

$$\begin{cases} x(\theta) = [a \sin(q\theta) + d] \cdot \sin(p\theta) \\ y(\theta) = [a \sin(q\theta) + d] \cdot \cos(p\theta) \\ z(\theta) = a \cdot \cos(q\theta) \end{cases} \quad (5.2)$$

其中我們以 (p, q) 代表扭結纏繞變化的係數， p 為繩子環繞的程度， q 為繩子圍繞的圈數。

在繪圖時，為了增加圖形線條的寬度更好觀察，我們可以用圖形中的 appearance 中將原來灑點的 dots 加寬，增加點和點之間的密集度，呈現出類似線的線條，讓我們更容易看到線條環繞旋轉的樣子，例如，以 $p = 1$ 為例，如圖 5-4：

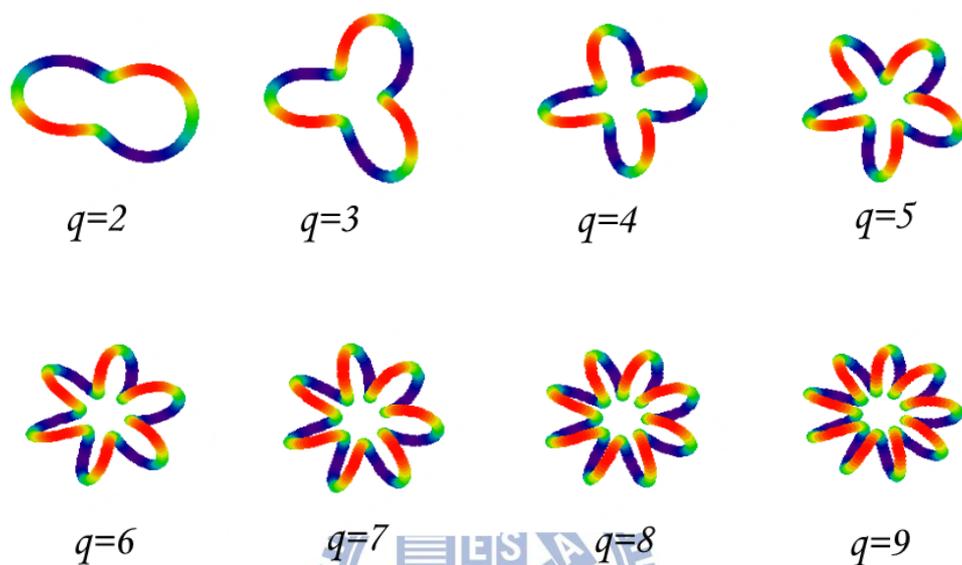


圖 5-4 $(1, q)$ ， q 變化 Torus Knot 圖

以 $p = 2$ 為例，如圖 5-5：

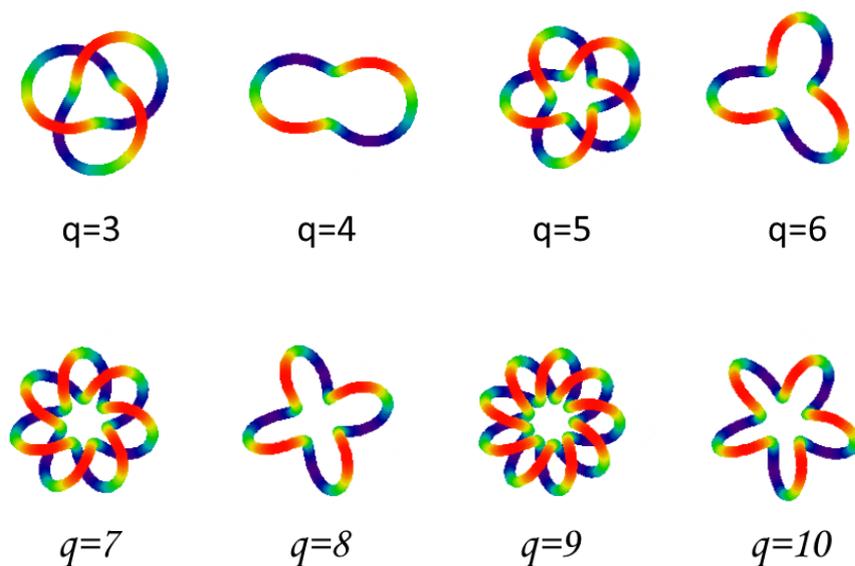


圖 5-5 $(2, q)$ ， q 變化 Torus Knot 圖

以 $p = 3$ 為例，如圖 5-6

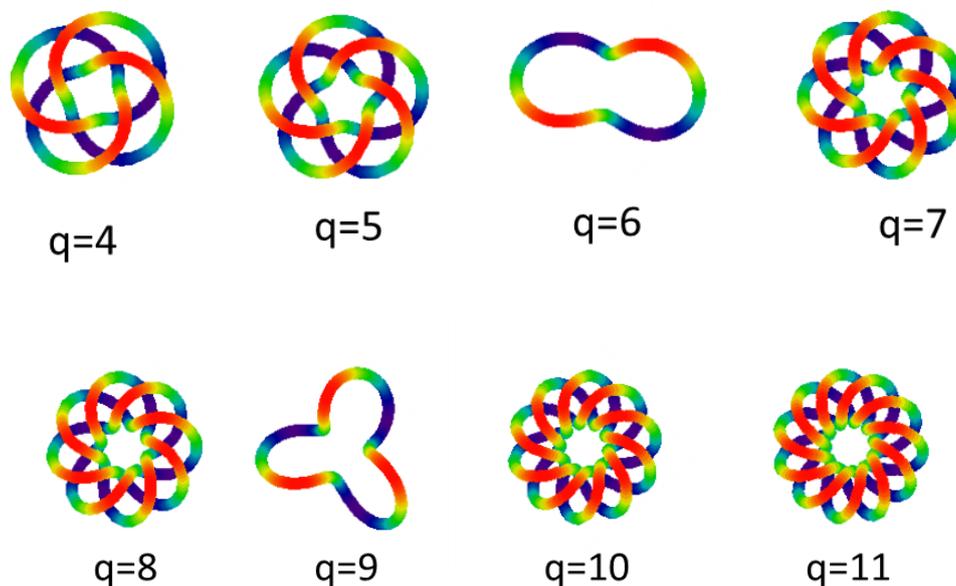


圖 5-6 $(3, q)$ ， q 變化 Torus Knot 圖

在確定 (p, q) 之後，若是改變圓面與圓中心的距離，也可以產生不同的變化的扭結圖形。

例如 $(p, q) = (1, 3)$ ，如圖 5-7，

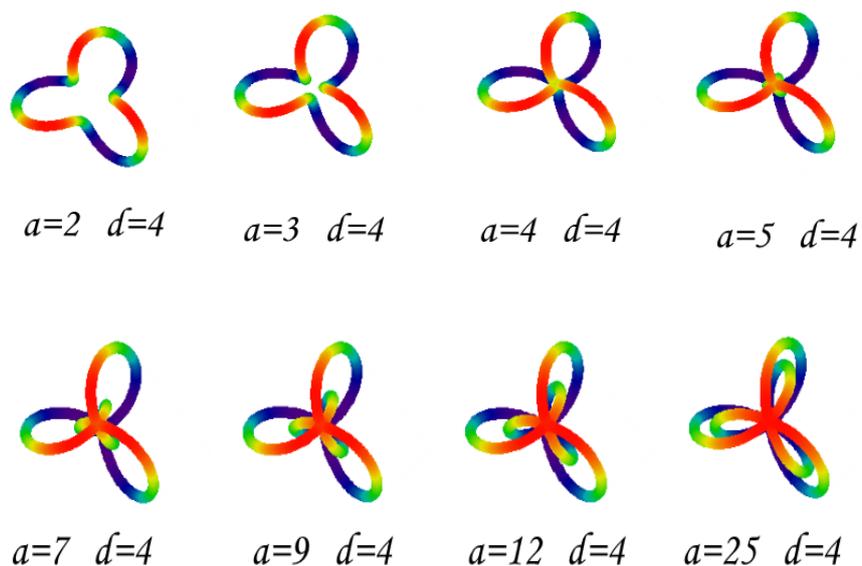


圖 5-7 $(p, q) = (1, 3)$ a, d 變化 Torus Knot 圖

經由圖 5-7 的結果，我們可以發現到，不同的條件所產生的圖形會有很多不同的變化，相關的程式碼如附錄 5.1；而在繪製最後的結果時，因為是立體的圖形，產生的圖形角度會和想像有些差異，此時只要用滑鼠調整圖形轉動到想要呈現的視角，即可產生想要產生的扭結圖案。

例如圖 5-8 為一開始產生的圖案，

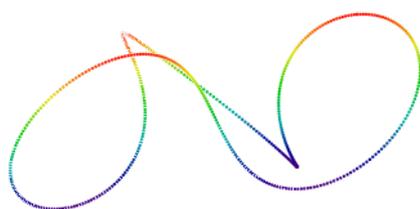


圖 5-8 mathcad 生成圖(未旋轉)

將上面的圖形分別投影在 x - y 平面，如圖 5-9(a)， y - z 平面，如圖 5-9(b)， x - z 平面上如圖 5-9(c)，所呈現的圖案，

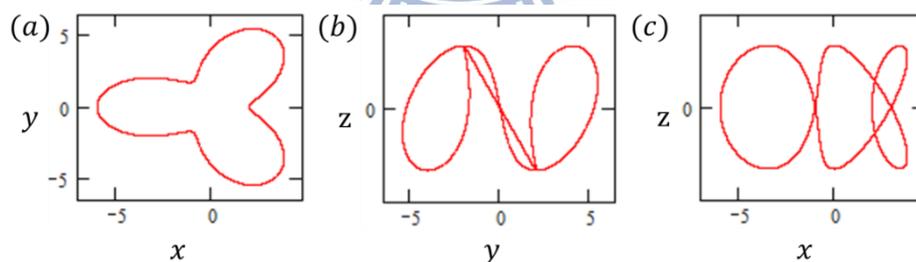


圖 5-9 mathcad 生成圖投影

經滑鼠調整旋轉，可以看到我們想要看到的 Torus Kno 圖形 t，如圖 5-10



圖 5-10 mathcad 生成圖(經旋轉)

5.2 Decorative Knot

Torus knot 所產生的結也可以產生各種不同的變化，我們稱他為裝飾結(Decorative Knot)，經由改變三角函數的參數，在三維的空間之中，利用圓的內擺線與外擺線的概念，在許多的交錯變化下，產生不同的結的結構，其參數式如下[21]：

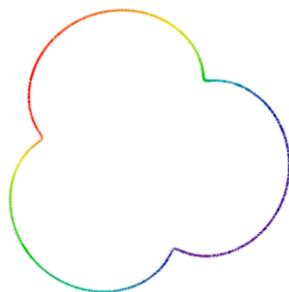
$$\begin{cases} x = m \cdot \cos(p \cdot \theta) + n \cdot \cos(q \cdot \theta) \\ y = m \cdot \sin(p \cdot \theta) + n \cdot \sin(q \cdot \theta) \\ z = h \cdot \sin(t \cdot \theta) \end{cases} \quad \text{for } 0 < \theta < 2\pi \quad (5.3)$$

其中， $p > 0$ 且 p, q 為非 0 的整數。 p, q 為週期的對應關係， t 則為皺褶的層數。而根據裝飾結的參數是，當 $z = 0$ 時，只有看 x, y 方向時，

$$\begin{cases} x = m \cdot \cos(p \cdot \theta) + n \cdot \cos(q \cdot \theta) \\ y = m \cdot \sin(p \cdot \theta) + n \cdot \sin(q \cdot \theta) \end{cases} \quad \text{for } 0 < \theta < 2\pi \quad (5.4)$$

根據不同的 m, n, p, q 所呈現出來的結果會是圓的內擺線與外擺線的線條。

由裝飾結的參數式，在某些狀況下，可以得到類似於 Torus knot 的結果如圖 5-11，



$$m = 1, n = 5, p = 4, q = 1, h = 1, t = 1$$

圖 5-11 Decorative Knot 相似於 Torus knot 的結果

同時，因為參數式的變化，經由改變 p, q, m, n, h, t 可以產生許多有趣的圖形，在下面的圖形中，我們固定 $m = 1, n = 0.45, h = 1, t = 1$ ， $p - q = 5$ ，等參數係數並分別繪製圖形，以圖 5-12 為例。

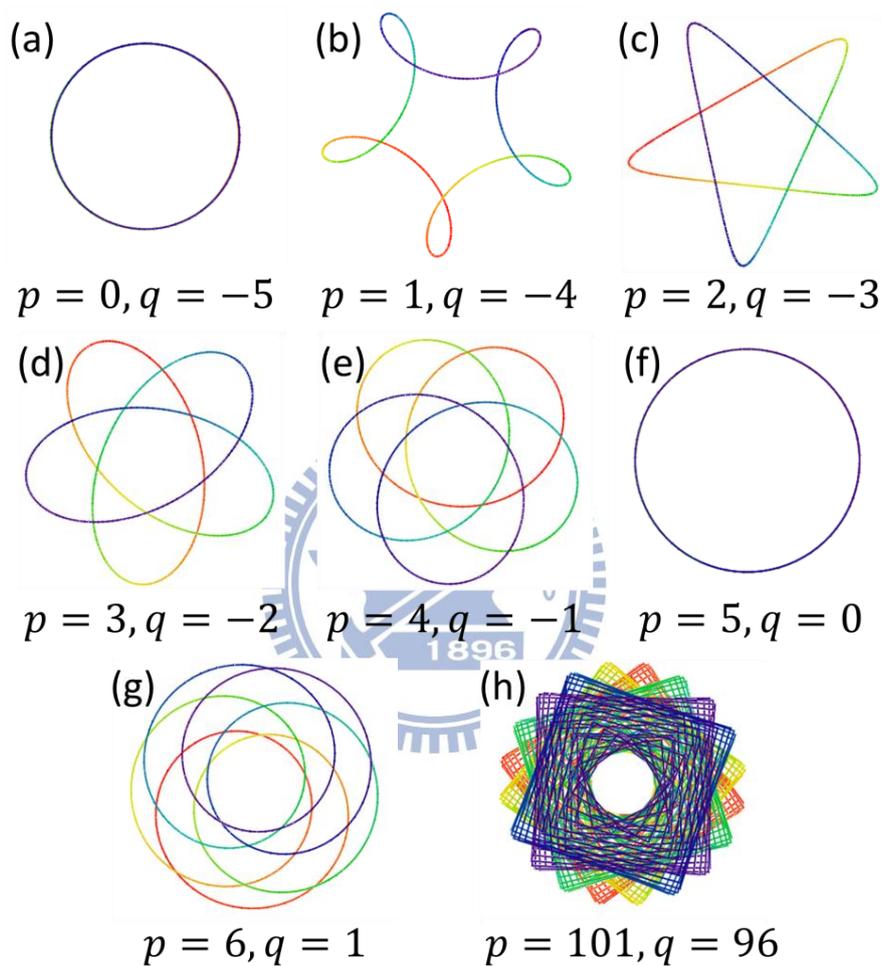


圖 5-12 Decorative Knot $m > n, p - q = 5$ 圖

在程式運算時，除了修改參數的數值，來繪製圖形，我們也可以在原來裝飾結參數式的後面再加上一些 \cos 和 \sin 的乘積，讓圖形產生更多不同的變化，例如我們在原來的參數式後面再加上

$$\begin{cases} x = m \cdot \cos(p \cdot \theta) + n \cdot \cos(q \cdot \theta) + a \cdot \cos(b \cdot \theta) \\ y = m \cdot \sin(p \cdot \theta) + n \cdot \sin(q \cdot \theta) + a \cdot \sin(b \cdot \theta) \\ z = h \cdot \sin(t \cdot \theta) + c \cdot \cos(d \cdot \theta) \end{cases}$$

$$\text{for } 0 < \theta < 2\pi \quad (5.5)$$

其中 m, n, a, b, c, d, p, q 皆為常數，如此可以產生更多不同的結的結果，如圖 5-13(a)，(b)，相關程式碼如附錄 5.2。

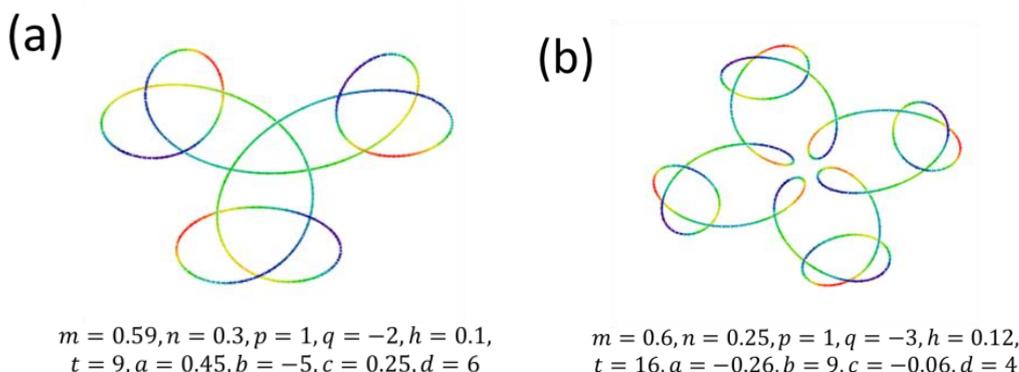


圖 5-13 Decorative Knot 加入不同參數示意圖

5.3 Knotted Tori

在前面介紹過 torus knot 與 decorative knot，這兩個圖形都是藉由點的軌跡的變動方式來環繞，其中我們又會以加寬點的大小的方式，來繪製出線的感覺，而 knotted tori 則以曲面環繞 torus 的方式來呈現打結的結構，因此我們需要在每一個方向變化上都加上另一個角度參數 ϕ 的變化，讓原來的線變化產生曲面的樣子，讓結的產生更具變化性的圖案，knotted tori 的參數式如下[22]：

$$\begin{cases} x(\theta, \phi) = [(a \cdot \sin(q \cdot \theta) + d) \cdot \sin(p \cdot \theta)] + b \cdot \sin(p \cdot \theta + \phi) \\ y(\theta, \phi) = [(a \cdot \sin(q \cdot \theta) + d) \cdot \cos(p \cdot \theta)] + b \cdot \cos(p \cdot \theta + \phi) \\ z(\theta, \phi) = a \cdot \cos(q \cdot \theta) + b \cdot \cos(q \cdot \theta + \phi) \end{cases} \quad (5.6)$$

在 $0 < \theta < 2\pi$ ，當 b 越大時，則曲面越大， b 越小時，則越接近原來的 knot 的圖案，如圖 5-14(a),(b)

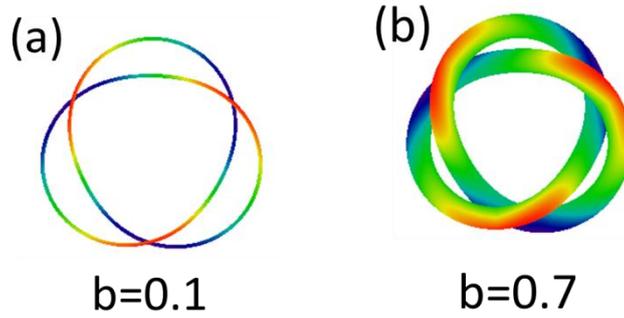


圖 5-14 knotted tori 不同粗細圖

在 mathcad 程式中，利用 Fill surface 介面繪圖，藉由改變 a, d, b, p, q 係數環繞出不同的結的圖案。例如在固定 $a = 1, d = 5, b = 0.7, p = 2$ 的情況下改變 q 如圖 5-15，相關程式碼如附錄 5.3。

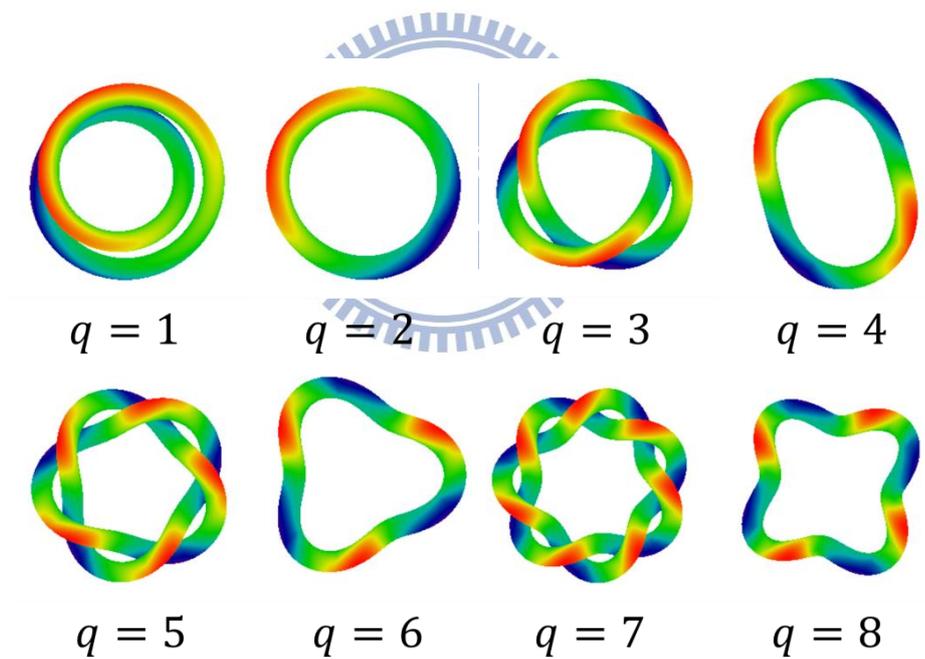


圖 5-15 $(2, q)$ q 變化 knotted tori 示意圖

附錄 5.1

Torus knot MathCad 程式碼

$a := 3$ $d := 5$ $p := 2$ $q := 3$ $N := 500$ $j := 0..N$

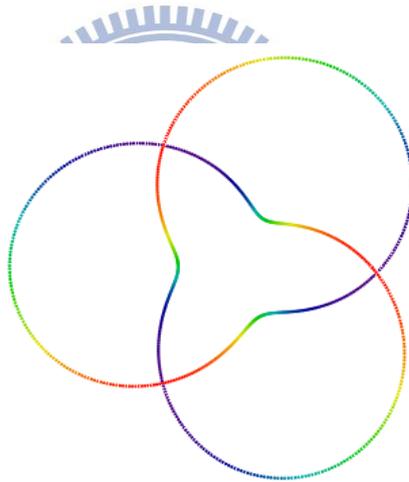
$m := 0..N$ $n := 0..N$

$$\theta_j := \frac{2\pi}{N} \cdot j$$

$$x := (d + a \cdot \cos(q \cdot \theta_j)) \cdot \sin(p \cdot \theta_j)$$

$$y_j := (d + a \cdot \cos(q \cdot \theta_j)) \cdot \cos(p \cdot \theta_j)$$

$$z_j := a \cdot \cos(q \cdot \theta_j)$$



附錄 5.2

Decorative Knot MathCad 程式碼

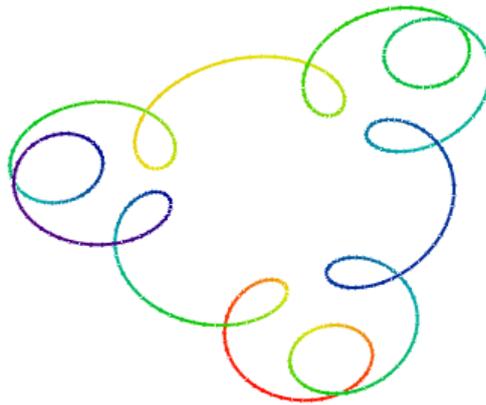
```
N := 500  m := 0.8  n := 0.25  p := 1  q := -2  h := 1  
t := 3  a := -0.24  b := 10  c := -2  d := 2  j := 0..N
```

$$\theta_j := \frac{2\pi}{N} \cdot j$$

$$x := m \cdot \cos(p \cdot \theta_j) + n \cdot \cos(q \cdot \theta_j) + a \cdot \cos(b \cdot \theta_j)$$

$$y_j := m \cdot \sin(p \cdot \theta_j) + n \cdot \sin(q \cdot \theta_j) + a \cdot \sin(b \cdot \theta_j)$$

$$z_j := h \cdot \sin(t \cdot \theta_j) + c \cdot \cos(d \cdot \theta_j)$$



(x,y,z)

附錄 5.3

Knotted tori MathCad 程式碼

$N := 500$ $a := 6$ $b := 1$ $d := 12$ $p := 4$ $q := 15$

$x(\theta, \phi) := [(a \cdot \sin(q \cdot \theta) + d) \cdot \sin(p \cdot \theta)] + b \cdot \sin(p \cdot \theta + \phi)$

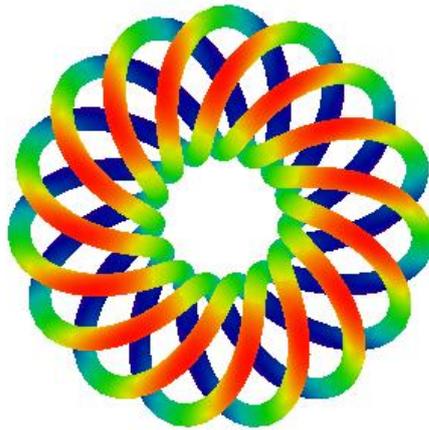
$y(\theta, \phi) := [(a \cdot \sin(q \cdot \theta) + d) \cdot \cos(p \cdot \theta)] + b \cdot \cos(p \cdot \theta + \phi)$

$z(\theta, \phi) := a \cdot \cos(q \cdot \theta) + b \cdot \cos(q \cdot \theta + \phi)$

$i := 0..N$ $j := 0..N$

$\theta_i := \frac{2\pi}{N} \cdot i$ $\phi_j := \frac{2\pi}{N} \cdot j$

$X_{i,j} := x(\theta_i, \phi_j)$ $Y_{i,j} := y(\theta_i, \phi_j)$ $Z_{i,j} := z(\theta_i, \phi_j)$



(X,Y,Z)

Reference

- [1]B. B. Mandelbrot, "How Long is the Coast of Britain? Statistical Self-similarity and Fractal Dimension ," Science, **156**, pp.636(1967).
- [2]<http://en.wikipedia.org/wiki/Fractal>
- [3]張詩言，Fractal 入門—電腦繪圖與自然藝術的結合理論，全欣資訊圖書股份有限公司（1995）
- [4]彭芳麟，計算物理基礎，高等教育出版社(2010)
- [5]廖思善，動手玩碎形，天下遠見出版股份有限公司(2006)
- [6]念家興，碎形與動態系統，數學傳播第 25 卷 1 期(2001)
- [7]Barnsley, Michael Fielding "Fractals everywhere ," Boston: Academic Press(1993).
- [8]Briggs, John, "Fractals : the patterns of chaos : a new aesthetic of art, science, and nature ," New York: Simon & Schuster(1992)
- [9]<http://classes.yale.edu/fractals/>
- [10]http://en.wikipedia.org/wiki/Iterated_function_system
- [11]L. Bindi, P. J. Steinhardt, N. Yao, and P. J. Lu, "Natural quasi-crystals ," Science **324**, pp.1306(2009).
- [12][http://en.wikipedia.org/wiki/Interference_\(wave_propagation\)](http://en.wikipedia.org/wiki/Interference_(wave_propagation))
- [13]E. Makovicky, "Five-fold Symmetry", World Scientific, Singapore-London ,pp.67(1992)
- [14]R. Goodman, "The Mathematics of a Mirrors and Kaleidoscopes", The Mathematics of Mirrors and Kaleidoscopes, pp.281(2004)
- [15]<http://jpkc.zju.edu.cn/k/526/xgzy/thb.htm>
- [16]Broucke, R.; Baxa, P. A. "Periodic solutions of a spring-pendulum system", Celestial Mechanics, pp. 8: 261 (1973)
- [17]祈忠勇，FFT 與訊號處理簡介，數學傳播第 18 卷第 4 期
- [18]<http://en.wikipedia.org/wiki/FFT>

- [19]Weele van der, J.P. and Kleine de, E. "The order-chaos-order sequence in the spring pendulum," *Physica A: Theoretical and statistical physics*, pp. 245(1996).
- [20]http://en.wikipedia.org/wiki/Torus_knot
- [21]Lindsay D Taylor, "Decorative Knotting using Trigonometric Parametrizations," *Visual Mathematics*, Vol.11,No.2(2009)
- [22]Daniela Velichová "Geometric Modelling Of Knotted Tori," *The Journal of Polish Society for Geometry and Engineering Graphics* Vol. 18,pp.91(2008)

