分形艺术程序设计

作者：潘金贵

出版：南京大学出版社

出版时间：1998 年 3 月

目录

§1.1 分形起源

1.1.1 分形现象很常见

在经典的欧氏几何中，我们可以用直线、圆锥、球等这一类规则的形状去描述如墙、车轮、道路、建筑物等人造物体，因为这些物体就是按欧氏几何的规则图形生成的。目前，几何学里所研究的对象，大体上是“规则”的，但是，自然界中，却存在很多“不规则”的复杂的几何对象，如山脉、云烟、波浪、树木、闪电，以及星团、短痕、浸润、冲积扇、泥裂、冻豆腐、水系、晶簇、蜂窝石、小麦须根系、树冠、支气管、星 系、材料断口、小肠绒毛、大脑皮层……等，它们无法用经典几何图形来描述，人们发现，没有传统的数学模型可以对它们进行研究，因为它们不再具有我们所早已熟知的连续、光滑可微这一基本性质了。这一大类形状奇怪的图形长期以来被认为是“不可名状的”、“病态的”而很容易被人们忽视了。

分形指具有多重自相似的对象，它可以是自然存在的，也可以是人造的。花椰菜、树木、山 川、云朵、脑电图、材料断口等都是典型的分形。根据英国儿歌改编的一首小诗可以说明分形之普遍:

一个分形的人，

穿过分形的森林，

走过分形的一英里，

分形地捡到了一枚分形的六便士。

买了一只分形的猫，

抓了一只分形的老鼠。

分形的人，

分形的猫，

分形的老鼠，

都挤在分形的小屋里。

分形人分形的大脑皮层里，

构思着分形猫分形地吞下分形老鼠，

分形老鼠被分形猫分形的小肠壁分形地吸收着……

如果您从未听说过“分形”，一时又很难搞清楚分形是什么，有一个简单迅捷的办法：去市场买一个新鲜的菜花(花椰菜)，掰下一枝，切开，仔细观察，思考其组织结构。这就是分形！好了，分形概念虽然极有价值，但它并不神秘，人人都能明白它的基本含义。

1.1.2 "fractal"的由来

著名理论物理学家约翰·惠勒(J.Wheeler)说过，在过去，一个人如果不懂得“熵”是怎么回事，就不能说是科学上有教养的人；在将来，一个人如果不能同样熟悉分形，他就不能被 认为是科学上的文化人。

分形不但抓住了浑沌与噪声的实质，而且抓住了范围更广的一系列自然形式的本质，这些形式的几何在过去相当长的时间里是没办法描述

的，或者被高贵的科学认为是不屑于研究的， 它们包括:海岸线、树枝、山脉、星系分布、云朵、聚合物、天气模式、大脑皮层褶皱、肺部支气管分支以及血液微循环管道等等。

分形在自然界中太普遍了，用分形语言去描绘大自然丰富多彩的面貌，应当是最方便、最适宜的。

有人会说了，不知道“宇称”、“对称破缺”、“耗散结构”等，算不上科学上的文化人， 但不知道 fractal 又有什么关系呢? fractal 是干什么的，怎么我们从词典上找不到这个词?

80 年代中期以前的辞书、词典，基本上没有收 fractal 这个词(1991 年版《形而上学与本体论手册》已收入“浑沌”、“分形”条目)。实际上分形一词“Fractal”是 1975 年由 Mandelbrot(B.B.Mandelbrot, 1924-,)首先拼造的新词，它来自拉丁语的形容词“Fractus”(frangere 的形容词形式)，含有碎片，不规则的含义，又有分数分级的意思。Mandelbrot 1924 年生于波兰华沙，后移居法国和美国，现为 Yele 大学数学教授。因为对分形几何理论的贡献他荣获了 1985 年的 Barnad 奖和 1986 年的 Franklin 奖，历史上爱因斯坦(A.Einstein,1879-1955)、 费米(E.Fermi,1901-1954)、卢瑟福(L.Rutherford,1871-1937)等人获得过此殊荣。

Mandelbrot 很早就开始对传统数学理论无法解释和研究的一些自然现象产生了兴趣并加以研究，从 50 年代起，他孤身一人，整日思索着一种新的几何学 。他试图通过这种几何学统一描述自然界、人类社会

中普遍存在的各种不规则现象，如流体 湍动、曲折的海岸线、多变的天气、动荡的股市、经济收入分配关系、棉花的价格波动等等 。严格说，那时候他自己也不明确自己在找什么，甚至不知道要找的是一种新的几何学。后来他把他的研究成果汇集成书出版，由于他所研究的几何对象往往具有非整数的 Hausdoff-Besicovitch 维数，因而他把书取名为“Fractal Geometry”，中文译为“分形几何”。1982 年他又出版了一本更为系统的经典著作“大自然的分形几何学”(The fractal geometry of nature)。

对于为什么取“Fractal”,有这么一个故事。 1975 年的一天，Mandelbrot 翻看儿子的拉丁语课本，突然受到启发，决定根据 fractus 创造一个新词，于是有了 fractal 这个英文词。后来法文词、德文词也都这样写；名词和形容词也都一样。同年他用法文出版了专著《分形对象:形、机遇与维数》(Les objets fractals: forme, hasard et dimension)，1977 年出版了此书的英译本《分形：形、 机遇与维数》。1982 年又出版了此书的增补本，改名为《大自然的分形几何学》。

还有一个有趣的故事是，70 年代末Mandelbrot的《分形：形、机遇和维数》(Fractals: Form,Chance,and Dimension)英文版在北京中关村一带的地摊上便可见到数十部,当时 北京大学力学系黄永念(1939-)教授和 朱照宣(1930-)教授每人买了一部，据说只花了几元钱。十多年后，当分形理论被科学界认同、热起来时，在世界上再去寻找这部原版名著，则几乎不可能了。当时，国际、国内科学界基本上不知道分形是怎么回事。

过了不久，分形如雨后春笋，在科学界流行起来，两位先生庆幸无意间(也许是上苍的有意安排)买到了一部世界名著。后来的事情大家都知道了，黄先生 和朱先生都是非线性科学领域的专家，与 北京大学其他同志于 1986 年创立了北 京大学非线性科学中心，为推动我国浑沌/分形研究作出了重要贡献。

现在，对分形的研究已远远超出了几何学的范围，自然界有许多现象与分形有关，如海岸线，河流网络等，它出现在许多物理，化学，生物以及诸多动力系统，甚至社会经济的理论和实际课题中。分形几何在揭示客观世界的许多复杂结构方面是一个有力的工具。著名科学家 John A。Wheeler 说:“谁不熟悉分形，谁就不算是科学家”。当然，他说这话也许有些过分，但我们应该重视分形理论是没错的。

1.1.3 ″分形″的由来

Mandelbrot 有着不平凡的人生经历，他 1924 年 11 月生于波兰华沙，1936 年搬到法国巴黎，1958 年去美国，1974 年成为 IBM 的一位研究人员至今。1985 年获巴纳奖章,1986 年获富兰克林奖章 (Franklin Medal)。现在他是美国艺术与科学学院(American Academy of Arts and Sciences)院士、美国国家科学院(U.S.National Academy of Sciences)院士,现任职于耶鲁大学。

Mandelbrot 所受的教育不很规则，他甚至声称背字母表都有困难，但他善于以图形化的方式思 维。据他本人讲，当初参加法国著名的高等工业学院(Ecole Polytechnique)关键性的入学 考试时，他不能很好地对

付代数题，但是他却成功地在头脑中通过把代数问题转化为图形而取得高分。他不但对几何形状感兴趣，而且特别关注“不规则”的形状。

接受大学教育以后,Mandelbrot 的生涯变得与他所感兴趣的形状一样无规则。他在加州理工大学研究航空学，受到普林斯顿高等研究院的杰出数学家约翰·冯·诺伊曼(J. von Neumann, 1903-1957)的支持，并在一系列领域做着研究工作。“我时不时被某种突如其来的力量所驱使，恰好在撰写研究论文的当中放下这个领域的研究。我兴冲冲奔向另一个感兴趣的新课题，而我以前对此领域什么也不知道。我按本能行事，却说不大清楚为什么，直到很久很久以后。”

在 1982 年出版的《大自然的分形几何学》一书的开头，作者写到:“我的第一篇科学论文发表于 1951 年 4 月 30 日。多年来，许多人觉得我的每项研究所取的方向都不相同。但这种表面上的无序性只是一种错觉，在其背后有明确的统一目标，本书及以前的两个版本正是试图阐明这个目标。聚沙成塔，我的大多数工作成了一门新学科的产前阵痛。”

1958 年芒德勃罗已是一名专业研究人员，1974 年在 IBM 很有威望的纽约约克郡高地托马斯· 沃森(Thomas J.Watson)研究中心任职。在那里，一种新的几何学在他的头脑中萌生了，它不同于以前所知道的任何几何学。(按陈省身(Chern Shing-Shen,1911-)的观点，历史上几 何学可分为六个时期: 1)公理(欧几里德); 2)坐标(笛卡尔，费马); 3)微积分(牛顿，莱布 尼兹); 4)群(克莱因，李); 5)流形(黎曼); 6)纤维丛(嘉当，惠特尼)。当然，这也只是一 种说法。)芒德勃罗创立了“分形”理论。

在这个思想萌发的初期，他就用分形来刻画股票价格，竟产生了足以愚弄这一行当专家的数学赝品。他的分形显示，大的涨、跌期模仿着每月、每天的价格波动，于是整个市场从它的最大尺度到最小尺度是自相似的。

Mandelbrot 转向数据传输中的噪声问题,用他的新几何学创造了一个可操作的模型。他不使用天文学数据，竟通过数学用图形显示了天体物理学家刚刚证实的宇宙星系分布。“我很清楚，自相似决不是一种平淡无奇的、无意义的性质，它是生成图形的一种非常有力的方法。” 芒德勃罗说的“自相似”指细节在递降尺度上能够复现。

据《湍鉴》(Turbulent Mirror)一书介绍，尽管芒德勃罗 对他的分形倾注了传教士般诲人不倦的热情，但这时候分形概念还是不能惹人注意。物理学 过去总是设法把自然界许多精致性质堆放在泛泛的“浑沌”与“无序”标题之下，与此类似 ，自然界的许多精致形式及其丰富细节过去也被常规几何学所忽略了。

进入 80 年代中期，各个数理学科几乎同时认识到了分形概念的价值，人们惊奇地发现，哪里有浑沌、湍动和混乱，分形几何学就在哪里登场。新学科的创立充满艰辛,但也充满乐趣， 有词为证:

迭代风流，奇点依旧。

海岸弯弯几何？芒氏分形秀。

罗素邂逅孤子，劳苦何所求？

反馈分岔浑沌路，人云小费谬!谬!

科海逐浪异宿，莫随当年老 Q。

(应稍作说明的是，“小费”指费根鲍姆(后文要专讲他的工作)，“罗素”(J.S.Russell)是指发现孤波的那位造船工程师，而不是那位有名的哲学家罗素(B.A.W. Russell, 1872-1970)，“迭代”、“奇点”、“孤子”、“分岔”、“异宿”等都是专有科学名词。)

70 年代末fractal传到中国，一时难以定译。一日，中国科学院物理所李荫远(1919-)院士说，fractal应当译成“分形”，郝柏林(1934-)、张恭庆(1936-)、黄(田+匀)(女，1935-)、赵凯华(1930-)、朱照宣等科学家表示赞同，于是在中国大陆fractal逐渐定译为“分形 ”，如今台湾还译“碎形”，显然不如“分形”好。

分形的特点是，整体与部分之间存在某种自相似性，整体具有多种层次结构。“分形”之译的确抓住了fractal的本质--科学本质、哲学本质和艺术本质。中国传统文化中关于“分” 与“形”有丰富的论述，想必李荫远院士极为熟悉。李院士是物理学名词审定委员会三名顾问之一，另两位是钱临照(1906-1999)和马大猷(1915-)。

宋明理学关于“理”(“理念”、“一”或者“太极”)与“万物”、整体与部分、一般与具体的关系的思想吸收了佛家观念，特别是华严宗和禅宗的观念，篇首引文就是一例。此外佛家还有“月印万川”的形象说法。

哲学史上更有“理一分殊” 的著名命题，对其解释有若干种，宋代的朱熹(1130-1200)是从唯心的方面说，明代罗钦顺(1465-1547)则是从唯物的方面说，都很深刻。罗氏特别指出： “盖一物之生，受气之初，其理惟一；成形之后，其分则殊。其分之殊，莫非自然之理，其理之一，常在分殊之中，此所以为性命之妙也。”我们不得不佩服古人的智慧，罗氏已将哲学道理讲得非常透彻了。

回头再看，李荫远的译名实在于平凡处见功力，如李善兰(1811-1882)译“微分” (differentiation)、“积分”(integration)，王竹溪(1911-1983)译“湍流”(turbulence)、“逾渗”(percolation)和“输运”(transportation)。

1.1.4 分形与数学的关系

我们在学微积分时，老师常会提到Weierstrass的处处连续而处处不可导的例子以及Peano曲线，Koch曲线等在经典数学范畴让数学家们无从下手，因而把它们作为“怪物”“反例”的事情，其实，这些图形正是分形！所以，分形图形很早就有描述，只是数学家们还没有认识到其巨大的应用背景而加以注意和研究。Mandelbrot在他的文章“Fractals and the rebirth of iteration theory”中特别评论到早年数学家们的这些工作：“值得赞扬的是由于他们发明了如此的结构，使我能最后把它们串连在一起，从而找到其宝贵的价值。应该受到指责的是由于他们没能在这些结构之间看到并开发出一种密切的内在联系。他们对待每一个结构，就象是对待一个畸形怪胎或不受欢迎的反例，这就从根本上忽视和疏漏了它们真实和深刻的内涵。”

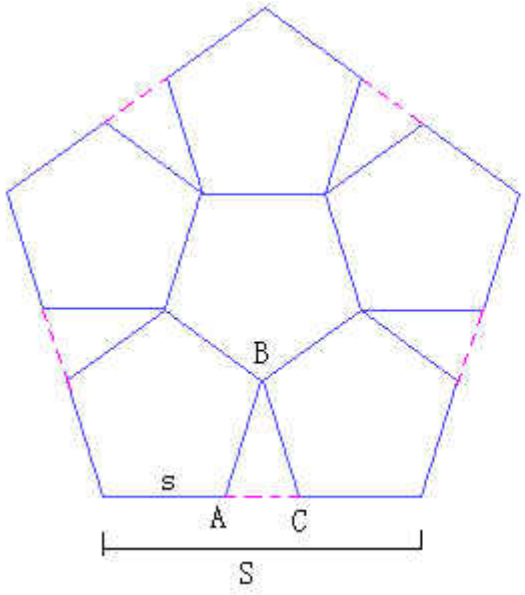
在近代科学中，分形常和分歧（Bifurcation）、孤粒子(Soliton)、混沌（Chaos）相提并论，因为物理学家特别关注这些现象常常交织在一起。分形混沌现象常常产生一幅幅变化莫测奇境般的图象，令理论科学家叹为观止。可以说，计算机对图形的作用，绝不亚于显微镜对生物和医学的作用。就连科学家一般不轻易涉足的艺术领域，分形也大有用武之地，因为在计算机上产生的山脉，彩云，花草，树木等画面，已达到了以假乱真的地步。近年来，计算机动画等在电影特技上的表现常常让人们拍案叫绝，计算机动画中就常有分形的应用。

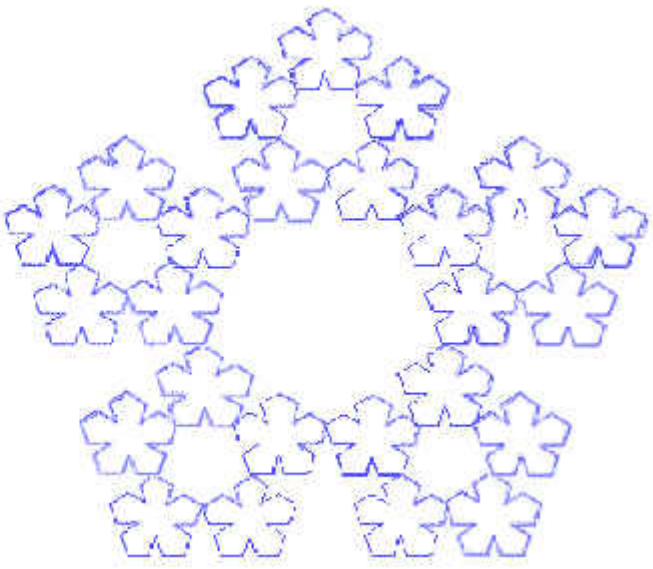
称之为分形的结构一般都有内在的几何规律性，即比例自相似性，并不是杂乱无章的，就象混沌一样，在无序中含有有序的结构。大多数分形在一定的标度范围内是不变的，在这个范围内，不断地显现放大任何部分，其不规则程度都是一样的，这个性质称为比例性。按照统计的观点，几乎所有的分形又是置换不变的，即它的每一部分移位、旋转、缩放等在统计的意义下与其它任意部分相似。这两个性质表明分形决不是完全的混乱，在它的不规则性中存在着一定的规则性。它同时暗示着自然界中一切形状及现象都能以较小或部分的细节反映出整体的不规则性。

1.1.5 分形例子

“分形”这个词是 Mandelbrot 首先创造的，自然界许多物体，如树、海岸线、云等，现在看来都具有分形的性质。可是在历史上，早就有艺术家和数学家创造出来过一些抽象的分形形式的物体，只是他们还没有

意识到这里面包含的更深层次的理论。在 Mandelbrot 的书中就提到过一些历史上的分形例子，如荷兰艺术家 Maurits Escher (1902-1970) 在其拼贴画中使用过一种技术：把身体的拷贝作为身体的一部分。这些图画及 Heri Poincar é 的关于动力系统的一些开拓性工作给了 Mandelbrot 极大的启发。更早时候，Albert D ü rer(1471-1528)就基于一个正五边形生成了一个分形体：对一个正五边形的每条边，向外生成另外五个正五边形，构成了一个大五边形的轮廓。我们可以发现，两个小五边形之间的等腰三角形的底边和腰长度之比为 AC:BC=2cos72"=0.618，正是黄金分割点。小大五边形边长之比为 s:S=1:(2+2cos72")，只要 S 知道了，就可以算出 s。产生分形体的做法是：给出一个边长为 S 的大正五边形，在里面放进五个边长为 s 的小的正五边形，再在每个五边形里面各放进五个更小的边长比为 S:s 的正五边形，重复该过程以至无穷，就得到一个分形体，它具有无限精细的结构。



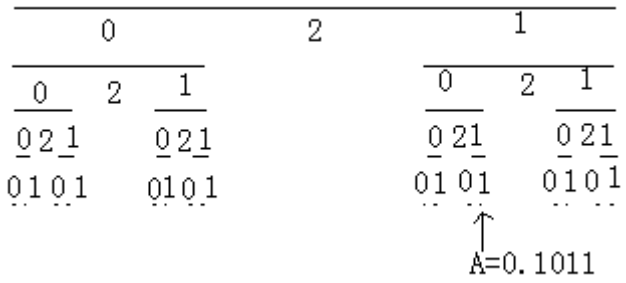


D ü rer 五边形与分形

下面我们来看几个经典的分形例子。

(A) Cantor 三分集

19 世纪末，数学集合论发展起来了，数学家们创造出了一些及其怪诞的集合，其中一个最为著名的就是 Cantor 三分集，它是著名集合论数学家 Georg Cantor(1845-1918)在对 Fourier 级数的收敛性研究中构造出来的。Cantor 三分集的构造非常简单：从闭区间[0,1]的实线段开始，去掉中间三分之一长度的部分(1/3,2/3)，留下了闭区间[0,1/3]、[2/3,1]，对每个闭区间重复上述过程以至无穷，其留下的点就是典型的 Cantor 三分集:



Cantor 三分集除了比例自相似外，还有一些很有趣的性质。(1) 我们来看 Cantor 三分集的长度：第一次截去 1/3，第二次截去剩下的 2/3 的 1/3…，截去的总长度为一个简单几何级数之和 {1+2/3+(2/3)2+(2/3)3+…}/3=(1/3)/(1-2/3)=1。这意谓着剩下的点尽管有无穷多，但仅仅“拥挤”在一个长度为 0 的空间，但显然这些点是不连通的，任意两个点之间都有未被填充的空间，就象“灰尘”一样。(2) 如果我们对留下的点进行编码，设[0,1]区间被分为三部分，分别以 0,2,1 表示，每次以 2 表示截去的部分，0 表示留下的左边部分，1 表示留下

的右边部分，则对 Cantor 三分集上的每个点都可以用 0 和 1 两个数编码表示，如图 1.1.2 中 A 点可表示为 0.101110110。这样一个二值数，正如计算机里二进制的表示法，可以表示[0,1]中的所有的数，即它能够“充满”整个区间，可见它包含无穷的点，总长度为 0，但却一一对应区间中所有的实数。

（B）海岸线模型：

到过海边的人都见过海岸线，我们也常常见到介绍某国的文字中有“某某国有海岸线多长多长”的字样，那么，这个数字准确吗？它是怎么算出来的？Mandelbrot于 1967 年在《科学》（Science）杂志第 155 期上发表了一篇富有启发性的文章，题目就叫“英国的海岸线有多长？”（How long is the coast of Britain? ）。文章中说，海岸线这种曲线的度量是无法得到一个确定的答案的，测定的长度依赖于所采用的测量的尺度。如果测量单位较大，就如同远距离观察的效果，较小的海湾是不可能看到的，所以在测量中无法得到反映。当你接近海岸线时，相当于以较小的度量单位作出测量，一些较小的海湾就变得清晰了。如果再缩小尺度，那么海湾的细节也可以观测到，尺度足够小时，海岸线的更精细的凹凸波动都可以清晰地显现出来，因此，Mandelbrot断言，海岸线的长度是不确定的。

海岸线的这种特性正是分形所具有的，也是促使 Mandelbrot 去研究分形现象的重要原因。我们稍加分析就会发现这种现象与我们熟知的经典几何图形如直线，圆周等完全不一样。为了更好地说明这一点，我们

先引进维数的概念，具体讨论我们在后面会涉及到。我们都知道，线度放大到原来的 2 倍，线段的长度也扩大到原来的 2 倍，而平面图形的面积扩大到原来的 4 = 2x2 倍，空间物体的体积扩大到原来的 8 = 2x2x2 倍，即对经典的几何对象，度量单位 û 与测量结果 L 之间的关系依赖于某个常数 D：线度放大 k 倍，整个对象就放大到原来的 p=k^D 倍，这个 D 就是我们常说的空间维数(点是 0 维的，线是 1 维的，面是 2 维的，体积是 3 维的)。我们把上面关系改写一下：D = lnp/lnk，这时，D 已经不必规定为整数了。

在度量海岸线长度时，测量结果L与度量单位r之间的依赖关系中，就要考虑常数D的影响，经过数学分析，L(r)正比于$r^{(1-D)}$,当L(r)=r·N(r)时,N(r)正比于$r^{(-D)}$。在数学上，对于m维空间点集X，记N(r)是覆盖X所需的半径为r的m维球的个数，则当-r>0 时，如果N(r)的增长规律服从N(r)正比于$r^{(-D)}$，就称点集X的Hausdoff维数为D。

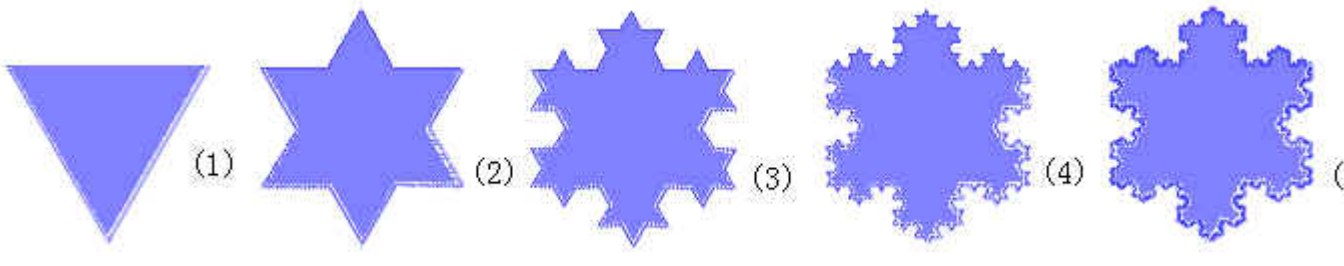
我们这儿要指出的是，关于维数的定义可以有多种方式，上面定义只是其中最为简单的一种。一般来说，对分形维数的计算是很复杂的，应用不同的定义算出的维数并不相同，所以对分形维数的近似计算方法的研究也有很多的讨论。一般来说，规则几何对象的维数总是整数的，而分形图形的维数一般不是整数，所以，Mandelbrot 在一开始给分形下定义时就简单地说：“所谓分形，指的就是其 Hausdoff 维数不是整数的几何对象。”尽管这个定义后来证明过于简单和武断，但确实是一个识别分形的好方法。

应用海岸线模型，人们做了很多实验，发现了许多类似于海岸线的分形现象。例如，语音波形是相当复杂的，如果将实验记录下来的语音波形的振幅与时间的关系转换成向径与幅角，则在极坐标下，语音波形变成了海岸线模型。经过计算发现不同动物的不同状态对应于相应稳定的分形维数：海豚 1.90，猫 1.74，生气的猫 1.78，人的耳语 1.49（见 Pickover CA & Khorasani AL. Fractal characterization of speech waveform graphs. Computer & Graphics. 10(1986), No.1).

类似地，云彩边界、流体的湍流和山地的轮廓等图形也是分形。

(C) Koch 曲线

1904 年，德国数学家 Helge von Koch(1870-1924)研究了一个处处连续而处处不可导的曲线例子，后人一般称为 Koch 曲线(Koch 雪花)，并吸引了一批著名数学家如 Giuseppe Peano(1859-1932)、David Hilbert(1862-1943)致力于构造和研究类似的分形曲线：第一步，取边长为 1 的正三角形，将每边三等分，以各边的中段为边，向外作小的正三角形，并去掉原来的中段，得到一个星形十二边形；第二步，继续将此十二边形的每条边三等分，以各边中段为边向外作更小的正三角形并去掉原来的各个中段，生成一个 48 边形；如此继续下去以至无穷，得到的极限曲线是连续的但在任何地方都没有确定的切线，如图所示。

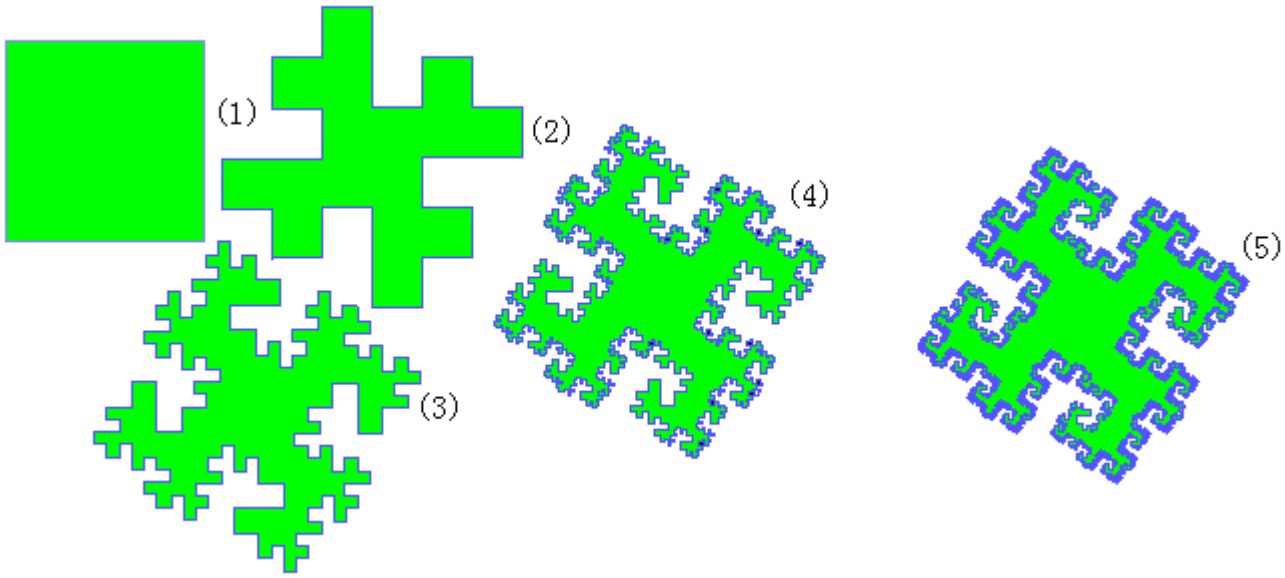


Koch 雪花

从上面叙述的曲线生成过程可以知道，从任何一步到下一步，曲线(准确地说是折线段)的周长就增长到原来的 4/3 倍，所以，曲线系列的周长趋于无穷大。我们仍然套用前面的维数定义，记正三角形边长长度为 r 时曲线的长度 L(r)，则有递推公式 L(r/3)=4L(r)/3，这时我们可以找到形如的 $L(r)=r^{(-D)}$解，只要 D 满足 $3^{(D-1)}=4/3$，所以有 Koch 曲线的 Hausdoff 维数为

$$D=\ln 4/\ln 3=1.2618\ldots$$

为了更好地说明问题，我们只考虑正三角形的每条边的变化过程(其余两边完全一样)，每条边对应一段线段，称为 Koch 多边形的边，将线段三等分，以中间一段为边作正三角形并去掉中间这段，对应的多边形图形一般称为“生成元”(或称为“发生器”)，以代替前一步 Koch 多边形的各条边。我们很容易发现，生成元的选取是有很大的灵活性的，对应的 Koch 曲线也很不相同，对应的分形维数也不一样。生成元的波动越大，对应的 Koch 曲线越复杂，其分形维数也可以很接近于 2，甚至维数就是 2 的，下图是其中的一些例子。

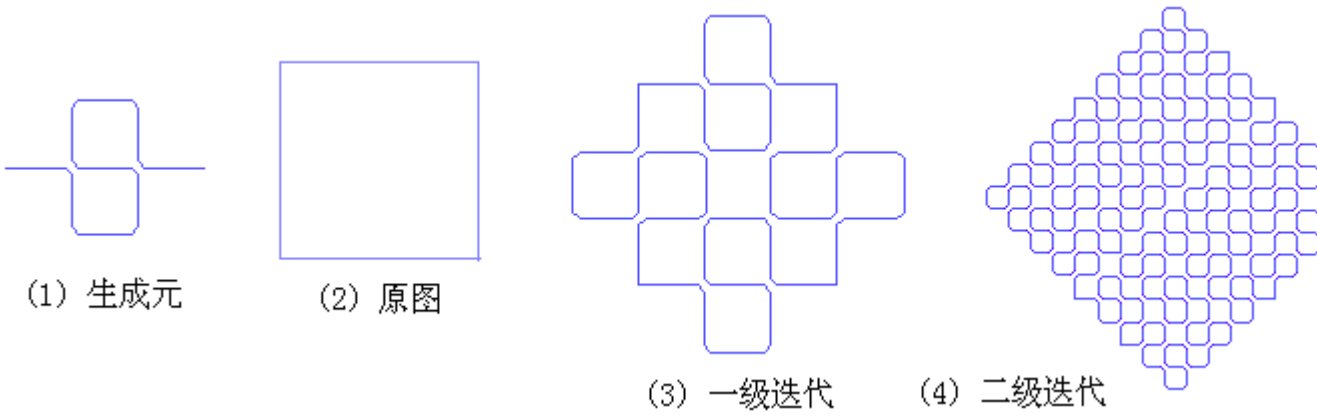


Koch 曲线的一些推广

（D）Peano 曲线

1890 年，数学家 Peano 设计了一个 Hausdoff 维数为 2 的曲线例子，后人称之为 Peano 曲线。在数学史上，Peano 曲线曾引起极大的关注。我们都知道，经典曲线是 1 维的，相对于平面，曲线的空隙是很大的，你见过几乎充满平面的曲线吗？在曲线理论建立以前，人们对“什么是曲线”认识不尽相同，Peano 曲线的提出是为了作为曲线的一个特例而构造的，在分形理论出现以后，人们就很容易发现它是一个典型的分形图形。

Peano 曲线构造时取初始图形为单位正方形，对正方形的每条边，取生成元为下图(1)，生成的图形象蜂窝，如下图(4)，容易验证其分形维数均为 2。



Peano 曲线

此外还有 Julia 集及平面 Brown 运动轨迹等.

Mandelbrot的学术简历

Mandelbrot评传

芒德勃罗与北京大学非线性科学中心主任赵凯华教 授[刘华杰摄]

分形理论是一门交叉性的横断学科，从振动力学到流体力学、天文学和计算机图形学，从分 子生物学到生理学、生物形态学，从材料科学到地球科学、地理科学，从经济学到语言学、 社会学等等，无不闪现着分形的身影。分形理论已经对方法论和自然观产生强烈影响，从分形的观点看世界，我们发现，这个世界是以分形的方式存在和演化着的世界。

下面引用 Mandelbrot 的一段话，也许可以使您很快进入状态:

为什么几何学常常被说成是“冷酷无情”和“枯燥乏味”的?原因之一在于它无力描写云彩 、山岭、海岸线或树木的形状。云彩不是球体，山岭不是锥体，海岸线不是圆周，树皮并不 光滑，闪电更不是沿着直线传播的。

更为一般地，我要指出，自然界的许多图样是如此地不规则和支离破碎,以致与欧几里得(几何)——本书中用这个术语来称呼所有标准的几何学——相比，自然界不只具有较高程度的复杂性，而且拥有完全不同层次上的复杂度。自然界图样的长度，在不同标度下的数目,在所有实际情况下都是无限的。

这些图样的存在,激励着我们去探索那些被欧几里得搁置在一边，被认为是“无形状可言的 ”形状，去研究“无定形”的形态学。

然而数学家蔑视这种挑战，他们想出种种与我们看得见或感觉到的任何东西都无关的理论，却回避从大自然提出的问题。

作为对这个挑战的回答，我构思和发展了大自然的一种新的几何学，并在许多不同领域中找 到了用途。它描述了我们周围的许多不规则和支离破碎的形状，并通过鉴别出一族我称为分形的形状，创立了相当成熟的理论。（陈守吉等译《大自然的分形几何学》，上海远东出版社出版）

分形理论有很强的解释能力，能说明大自然的许多形态发生和自组织过程，分形自相似原理和分形迭代生成原理对于人们更好地认识世界起到了推动作用。分形图形生成技术也对传统艺术造成了不小的冲击。但不能把一种科学理论任意夸大、玄学化。分形理论与所有其他科 学理论一样，决不是万能的。分形理论已走过轰轰烈烈的革命式发展时期，已进入平稳发展 过程。注意到其限度，不断创新，由分形引出的新科学才有生命力。

§1.2 分形纪事

文艺复兴时期著名艺术家、科学家丢勒(Albert Durer，1471-1528)基于正五边形向外无穷 复制，生成了一个分形体。

1827 年英国植物学家布朗(R.Brown,1773-1858)用显微镜发现微细颗粒在液体中作无规行走 ，此现象被称为布朗运动。后来科学家对布朗运动进行了多方面的研究，维纳(N.Wiener， 1894-1964)等人在此基础上创立随机过程理论。进入 80 年代，人们以分形的眼光看待布朗运动，并与“列维飞行”(Levy flight)相联系，找到了确定论与随机论的内在联系。

学过微积分的人都知道，函数的可微(即可求导数)性与连续性有内在联系。两者的关系是可微的函数必定连续，但连续的函数未必可微。一个简单的例子就是函数 $y=|x|$ 在 $x=0$ 处连续，但不可微。这个函数只有这么一个特别的点，除此以外其他点都可微。有的函数在有限个点处是不可微的，也有更特别的函数，它们几乎处处不可微。

1860 年，瑞士一个名气不算大的数学家塞莱里埃(C.Cellerer,1818-1889)在课堂上向皮克太特(R.Pictet)等学生讲解:“连续函数必定可微”的流行观念是错误的，并给出了一个类似维尔斯特拉斯(K.T.W.Weierstrass,1815-1897)函数的反例。黎曼(G.F.B.Riemann， 1826-1 866)的学生曼海姆(J.H.Manheim)等人回忆说，大约在 1861 年，黎曼在讲座中提到了类似的例子，但未发表。

不过，1970 年有人证明，塞莱里埃函数和黎曼函数不同于维尔斯特拉斯函数，它们不是处处不可微的，在某些点上它们是有导数的。

1872 年 7 月 18 日,维尔斯特拉斯向柏林科学院报告了分析学中的一个反例--一个处处连续、但处处不可微的三角函数级数，即著名的维尔斯特拉斯函数。不过此函数直到 1875 年才由杜布瓦-雷蒙(E.du Bois-Reymond)正式发表出来。

据考察,在维尔斯特拉斯之前,已有不少数学家知道存在所谓的“维尔斯特拉斯函数”，但都耻于发表它!因为它破坏了分析学的完美性。大约在 1834 年波尔查诺(B.Bolzano，1781-18 48)构造过类似的函数，但他可能并不知道它有那样“可怕的”性质。

1883 年，康托尔(G.F.P.Cantor,1845-1918)构造了三分集，也叫康托尔非连续统(Cantor di scontinuum)。它与实直线是相对立的，当时人们觉得它几乎是病态的。如今它已成为分形几何学的最典型、最简单的模型。

1890 年，皮亚诺(G.Peano,1858-1932)提出充满空间的曲线——皮亚诺曲线。

1891 年，希尔伯特(D.Hilbert,1862-1943)在《数学年刊》(*Mathematische Annalin*)上发表短文，提出了能充满平面区域的著名的希尔伯特曲线。

1904 年，瑞典数学家柯赫(H.von Koch,1870-1924)构造出柯赫雪花曲线。

1915-1916 年，波兰数学家谢尔宾斯基(W.Sierpinski,1882-1969)构造了谢氏曲线、海绵、墓垛。谢氏地毯是平面万有曲线(plane universal curve),谢氏海绵是空间万有曲线。奥地利数学家门格尔(K.Menger)证明，任何曲线都可嵌入谢尔宾斯基地毯中。

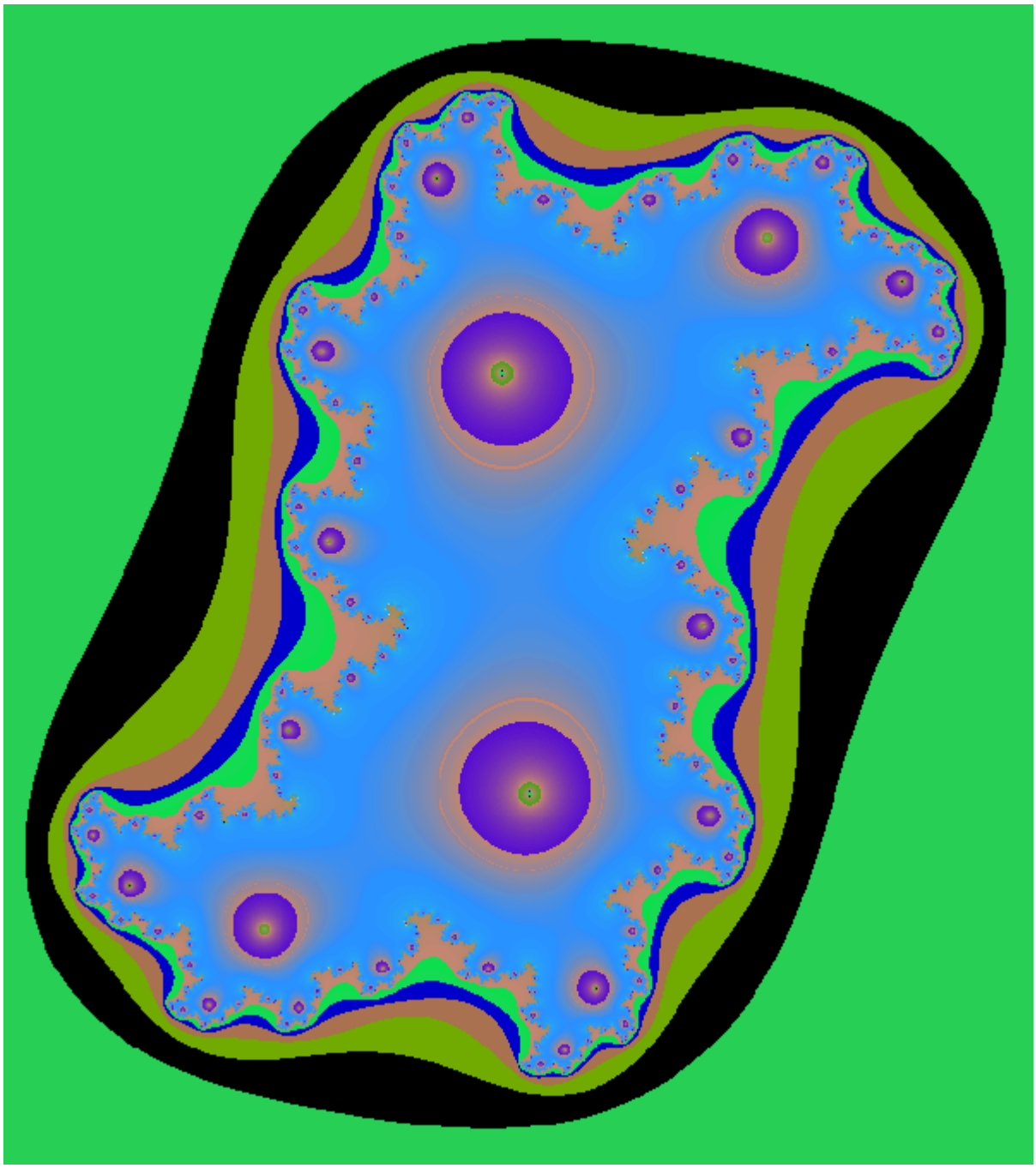
1918 年，康托尔去世。

1919 年，豪斯道夫(F.Hausdorff,1868-1942)给出维数的新定义，为维数的非整化提供了理论基础。

1918-1920 年左右，法国数学家朱丽亚(G.Julia,1893-1978)、法图(P.J.L.Fatou, 1878-1929)研究复迭代。朱丽亚于 1918 年(当时他 25 岁)在《纯粹数学与应用数学杂志》上发表了长达 199 页的杰作，一举成名。

1924 年 11 月 20 日 Mandelbrot 生于波兰。

1925 年柏林大学的克莱默(H.Cremer)组织讨论班学习朱丽亚的工作，并首次手工绘制了朱丽亚集的图象。



1926 年，洛特卡(A. J. Lotka, 1880-1949)提出洛氏定律。里查逊就“风”是否具有一定的速度发表议论。

1932 年，庞特里亚金(L. S. Pontryagin, 1908-)给出盒维数的定义。

1934 年，贝塞克维奇(A. S. Besicovich, 1891-1970)给出维数新定义。

1936 年 Mandelbrot 全家迁到巴黎。大约在 1945 年，他的叔叔芒德勃罗伊(S.Mandelbrojt,1899- 1983)向他介绍了朱丽亚的工作，但当时他并不喜欢朱丽亚那一套。可是大约在 1977 年，芒德勃罗自觉地回到了朱丽亚的论文里汲取营养。

40 年代末齐夫(G.K.Zipf，1902-1950)总结出不同语言中词频分布的幂律关系。

1952 年，芒德勃罗获博士学位。

1954 年，波兰数学家斯坦因豪斯(H.Steinhaus,1887-1972)讨论“长度”悖论，引起芒德勃罗注意，芒氏在 1967 年的“海岸线”文章中引用过此文。

1961，1963，1965 年芒德勃罗开始研究棉花价格，帕累托(V.Pareto,1848-1923)收入分布。

1967 年，芒德勃罗在《IEEE 信息理论学报》上发表论文《具有 *1/f* 谱的某些噪声，直流与白噪声之间的一座桥梁》。

1967 年芒德勃罗在《科学》上发表题为《英国海岸线有多长?统计自相似性与分数维数》的著名论文。

1968 年美国生物学家林德梅叶(A.Lindenmayer,1925-1989)提出研究植物形态与生长的“L 系统”方法。1990 年普鲁辛凯维奇(P.Prusinkiewicz)与林氏出版《植物的算法美》(*The Algorithmic Beauty of Plants*)一书。史密斯(A.R.Smith)等人 80 年代将 L 系统引

入计算机图形学，L 系统从此广为人知。现在，L 系统是生成分形图形的最典型方法之一。

1975 年，芒德勃罗创造分形(fractal)一词，以法文出版专著《分形对象》；沃斯(R.F.Voss ,1948-)用分形的思想研究音乐中的 *1/f* 噪声问题。沃斯在计算机上制作出“分形山脉”(被芒氏引作 1977 年专著的封底)。

1977 年，芒德勃罗出版英文版专著《分形：形、机遇与维数》，它是 1975 年法文版《分形对象》的增补本。

1977 年 9 月在英国塞尔福特(Salford)举行颗粒粒度分析会议，分形思想引入粒度分析。

1981 年，美国洛斯阿拉莫斯(Los Alamos)国立实验室成立非线性研究中心(CNLS)，以后世界 各国相继成立许多非线性科学中心。

1981 年维腾(T.A.Witten)和桑德(L.M.Sander)提出著名的 DLA 分形生长模型。

1982 年，芒德勃罗出版《分形：形、机遇与维数》一书的增补版，改名《大自然的分形几何学》。

80 年代初，弗尔聂(A.Fournier)、富塞尔(D.Fussell)、卡本特(L.Carpenter)将分形图形推 向好莱坞影视业，主要影片有《星际旅行之二：可罕之怒》*(Star Trek Ⅱ: The Wrath of Khan)*、《最后的星球斗士》(*The Last Starfighter*)。

1982 年，道阿弟(A.Douady)和哈伯德(J.H.Hubbard)等证明芒德勃罗集是单连通的。

1983 年格拉斯伯格（P.Grassberger)和普洛克西娅(I.Procaccia, http://chemphys.weizmann.ac.il/~cfprocac/ home.html）提出了从实验数据序列求分维的算法，现在通称为G-P算法。

1984 年《数学信使》(*The Mathematical Intelligencer*)杂志和德国的《GEO》杂志 刊登布来梅大学动力系统研究小组的分形艺术图片。

1985 年，芒德勃罗荣获巴纳杰出科学贡献奖章(Barnard Medal for Meritorious Service t o Science)。1985 年 5 月，芒氏受邀请去布来梅大学为分形艺术图形展览揭幕。

1985 年法尔柯内(K.J.Falconer)的专著《分形集的几何学》(*The Geometry of Fractal Sets*)出版。

1985 年昂伯格(D.K.Umberger)和法默(J.D.Farmer)提出胖分形(fat fractal)概念。胖分形是指具有分形边界且勒贝格(H.L.Lebesgue,1875-1941)测度不为零的集合。胖分形的勒贝格测度为非零有限值，维数为整数而且与所在的欧氏空间维数相等。分维已经不是描述胖分形的敏感参数，需要引入胖分形指数来刻画它。

80 年代中期美国洛斯阿拉莫斯非线性科学中心将非线性科学要研究的问题归纳为三个方面： 1)孤子和拟序结构； 2)混沌和分形； 3)斑图(patterns)的形成。

1986 年，芒德勃罗荣获富兰克林奖章。

1986 年北京大学成立非线性科学中心，挂靠在力学系。

1986 年培特根(H.-O.Peitgen,1945-)和里希特(P.H.Richter)出版《分形之美: 复动力系 统图象》画册，书中包括 88 幅全彩色分形图形，分形图形艺术正式诞生，此书 1987 年荣获“ 杰出技术交流奖”(Distinguished Technical Communication Prize)。

1986 年迪万内(R.L.Devaney,1948-)的专著《混沌动力系统导论》(*Introduction to Ch aotic Dynamical System*)出版，该书以很大篇幅讲述与分形有关的复解析动力学。

1985-1988 年，巴恩斯利(M.F.Barnsley,1946-)等人研究迭代函数系统(IFS)，试图解决图 形生成的逆问题--对已知图象找分形压缩算法，创建分形图形公司，分形技术开始推向市场，1988 年出版专著《处处得分形》(*Fractal Everywhere*)。

1987 年芒德勃罗荣获亚历山大·洪堡奖(Alexander von Humboldt Prize)，

1988 年荣获斯坦因迈兹奖章(Steinmetz Medal)。

1988 年费德(J.Feder)著《分形》一书出版。

1988 年，纽约时报记者格莱克(J.Gleick,1954- ，http://www.around.com)著畅销书《混沌: 开创新科学》(*Chaos:Making*

a New Science)出版，该书先后被译成近 20 种文字，书中收有多幅彩色分形图片。

1989 年芒德勃罗荣获哈维(Harvey)奖。

1989 年 7 月在成都四川大学召开“第一届全国分形理论及应用学术讨论会”。1991 年 11 月在 武汉华中理工大学召开第二届会议。1993 年 10 月在合肥中国科技大学召开第三届会议。

1990 年英国成立了一家利用混沌/分形理论生产并出售计算机艺术品的商店“Strange Attra ctions”。

1990 年李后强(1962-)、程光钺著《分形与分维》由四川教育出版社出版。

1991 年英国创办国际学术性刊物《混沌、孤子和分形》(*Chaos, Soliton and Fractals*)。

1991 年芒德勃罗荣获内华达奖章(Nevada Medal)。

1991 年底中国国家攀登计划“非线性科学”项目(“八五”期间 1991-1995)启动，到 1995 年 五年总资助金额 498 万，首席科学家为谷超豪(1926-)教授，挂靠单位为北京大学非线性科学中心。“八五”期间在该项目资助下共发表论文 1111 篇，被《科学引文索引》(SCI)收录 384 篇。

1992 年崔锦泰(C.K.Chui)著《小波导论》(*An Introduction to Wavelets*)在美国出版。小波(wavelet)分析与分形联系日益紧密。

1993年新加坡创办国际学术性刊物《分形》(*Fractals*)。

1993年李后强、汪富泉(1955-)著《分形理论及其在分子科学中的应用》由科学出版社出版 。

1993年李后强等主编《分形理论的哲学发轫》由四川大学出版社出版。

1993年芒德勃罗荣获沃尔夫物理学奖(Wolf Prize in Physics)。

1994年11月17日芒德勃罗荣获本田奖(Honda Prize)。

1995年美国佐治亚理工学院著名学者、“混沌传教士”福特(Joseph Ford)不幸去世。

1995-1996年中国科协“青年科学家论坛”两次举行非线性科学研讨会。

1995年王东生、曹磊著《混沌、分形及其应用》由中国科学技术大学出版社出版。

1996年北京大学非线性科学中心创办英文杂志《非线性科学与数值模拟通讯》(*Communic ations in Nonlinear Science & Numerical Simulation*)，在Internet上发行，由陈耀松(1928-)任主编。此杂志现已被美国《工程索引》(EI)检索。

1996年4月中央工艺美术学院、北京市科协等主办“ 96北京国际计算机艺术展”，在入选的300余幅作品中有近20幅作品直接采用了分形方法。

1996年8月FRACTINT 19.5在Internet上发行。

目前，混沌、分形、小波、时空离散系统、斑图、自组织系统仍然是非线性科学研究的重点，而分形与所有其他方面都有联系。

§1.3 分形概念

1.3.1 分形的定义

在欧氏几何中，有一些我们所熟悉的基本元素，如点、直线、圆等。在某种意义上，只有当它们结合起来构成了复杂物体后才具有实际的意义。但在分形中，它们的最基本元素却无法直接观察到，应该说分形首先是一种几何语言，它是由算法和数学程序集而不是由什么原始形态来描述的，这些算法借助于计算机而被转换成一些几何形态。一旦人们掌握了分形语言，那么就能象一位建筑师采用根据传统几何语言所制作的蓝图来描绘一栋房屋那样，准确而简洁地描绘出一朵云彩的形状。在讨论分形的定义以前，我们先看分形都具有什么性质。

1.3.1.1 分形应具有的经典性质

从数学上说，分形几何是一门几何学，它研究的对象是欧氏空间的一类结构比较复杂的子集。从分类的严密性角度，我们似乎应该首先给分形下一个明确的定义，由此来判断一个给定的图形是不是分形。但是，实际经验告诉我们，这样做太过于简单化了。事实上，分形几何的创始人 Mandelbrot 本人一开始也想这么做，并给过两个定义，但经过理论和应用的检验，人们发现这么简单的定义根本无法包括丰富的分形内容，因而被人们慢慢淡忘了。为了搞清什么是分形，我们先来分析一下

分形应具有的经典性质。

通过前面的一些例子分析可以看到，从几何学上看，分形是实空间或复空间上一些复杂的点的集合，它们构成一个紧子集，并且具有下面经典的几何性质:

分形集都具有任意小尺度下的比例细节，即具有无限精细结构;

分形集无法用传统几何语言来描述，它不是某些简单方程的解集，也不是满足某些条件的点的轨迹;

分形集具有某种自相似形式，包括近似和统计上的自相似;

分形集一般可以用简单的方法定义和产生，如迭代;

按某种维数定义，分形集的分形维数大于相应的拓扑维数。

针对不同分形图形，有时它可能只具有上面大部分性质，而不满足某个性质，但我们一般仍然把它归入分形。实际上，自然界和科学实验中涉及的分形绝大部分都是近似的，因为当尺度小到无法分辨时，分形性质也自然消失了，所以，严格的分形只存在于理论研究中。

1.3.1.2 分形定义

分形繁杂多样，势必会碰到如何区分和比较的问题，比如云彩千奇百怪，谁也说不清楚它的形状到底是什么，但是谁又都知道什么是云，而且还能分清什么是乌云，什么是浮云。显然，这些背后肯定存在某种规则性的东西。实际上，无任分形的起源和构造方法如何，所有分形都应具有某种重要的特征，我们可以通过这个特征来测定其不平整度、复

杂度或卷积度，而且这个特征的微小变化会引起形状的急剧变化。Mandelbrot 一开始时认为这个特征就是分数维数，他认为分数维是分形理论的最基本的特征，并给出分形的定义为：一个集合，如果其 Hausdoff 维数严格大于其拓扑维数，则称该集合为分形集。现在看来，这个定义显然不是很合理，因为它把一些明显的分形集排除了。此后，人们还提出过一些其它定义，但都有或多或少的缺陷。事实上，“分形”的定义有点象生物中“生命”的定义一样，“生命”没有严格明确的定义，但可以列出一系列生命体所具有的特征，比如对环境的适应能力、生命能力、运动能力以及繁殖能力等等。大多数生物都有上述特征，但也有少数生物对其中某些特征有例外。同样对于“分形”，似乎最好也是把它看成具有某些特征的集合，而不必刻意追求难以精确的定义。

综合起来，我们可以给分形下这么一个定义：分形集是一类不能用经典几何方法描述的“不规则”集合，它们基本满足分形的经典性质，但是在自相似的程度上可以有很大差别。分形几何就是研究所谓“简单”空间上这样一类“复杂”子集的一门新兴数学分支。

1.3.1.3 分形分类

我们可以根据分形集合生成算法的特征对分形进行分类，一般分为线性分形和非线性分形两大类。这两类分形，都有无穷的算法表述，因而也都包含无穷的分形图形。线性分形是最基本的一种，从数学上说，就是实现这些分形的算法中仅含有一次项，如Koch曲线、Peano曲线及迭代函数系统(IFS)，这些迭代变换使直线保持平直，仅改变其长度、

位置和方向。非线性分形则内容丰富得多，变化也多，如Julia集。

我们也可以根据分形产生算法中随机性加入的影响，把分形分为**确定性分形**和**随机性分形**两类。对确定性分形，其算法的规则是确定的，在算法中也没有随机性的加入，或者虽然有随机性的加入，但并不影响分形图形的形状，即算法的多次重复仍然产生同一个分形图，如IFS；对随机性分形，虽然其产生的规则也是一定的，但随机因素的影响，可以使每次生成过程产生的分形虽然可以具有一样的复杂度，但形态都会有所不同，它不具有可重现性，如Brown运动。

1.3.1.4 分形与欧氏几何的关系

分形已是当今最激动人心的研究领域之一，Mandelbrot 在他的书中这样写道："科学家发现，他们以前必须称之为粒状、流体状、中间状、丘疹状、麻窝状、树枝状、海草状、奇异状、斋乱状、弯曲状、波形状、束状、折皱状等不少形状从今以后能以严格的和强有力的定量方法加以处理，对此他们将会惊喜不已。""数学家们发现，那些至今为止认为异常的集合在某种意义下说应该是规则的，被认为病态的结构应该自然而然地从非常具体的问题中演化出来，对于大自然的研究应该有助于解决一些老问题和产生如此之多的新问题，对此他们也会惊喜不已。"

分形的发展很大程度上依赖于计算机科学技术的进步，这也对纯数学的传统观念提出了挑战。计算机技术不仅使分形领域的一些新发现成为可能，同时因其图形直观的表现方式也极大地激发了科学家和人们的兴趣与认识，推动了分形理论的发展。

我们可以对分形几何与欧氏几何作一个简单的比较:

	欧氏几何	分形几何
历史	2000 多年	最近 20 几年
对象	人造物体	适于自然形态
尺度	可用特定比例和尺度度量	没有特定的比例和尺度，具有无限细节性
方法	公式、基本元素	递归、迭代算法

1.3.2 作为认知方法的分形

分形概念早已不限于具体的数理学科，成了好似“力”、“能量”、“原子”、“量子”一 类的概念，具有世界观和方法论意义。当人们“蓦然回首”，发现无处不分形之时，这一功能上的拓展就已基本完成，剩下的主要任务只是揭示出具体的特征和类型。用库恩 (T.S. Kuhn, 1922-1996)的语言说,“范式”(paradigm)已发生转变，现在正进入常规发展阶段。

认知是以一定的理论、概念为前提的，这是当代心理学、认识论研究的一个基本结论。这也 从一个侧面反映出，人是“运用符号创造世界”的动物，人的概念世界就是人所抵达的自然 世界的极限。概念世界的缺陷决定了人所认识的“自然世界”的清晰程度。从人类进步的角度看，概念、语言、理论、模型是人类理解世界的凭据和藩篱。语言以

概念为要素，概念的 展开是理论（理论的浓缩是概念），概念的图象架构是模型，因而概念更显基本。不借助概 念无以认识世界，而概念不过是一定历史阶段的产物，即使在当时看来精致完美，从长远看 来也必定不恰当甚至错误，于是概念的局限影响到“世界图景”的正确描绘；概念的狭隘和偏激，塑造了扭曲的自然之象。回顾科学史，这一切脉络清晰。上世纪末，用以理解原子或 亚原子的“合适”概念是牛顿力学概念及建立其上的太阳系行星规则运动模型，因而，早期 的原子模型总是逃脱不了太阳系结构的影子。科学的发展最终突破了原来的牛顿力学框架， 在新的原子概念中，原子核不像太阳，电子也不像行星。

分形概念的创制、传播、被认可并非偶然。它是伴随非线性复杂性研究的时代潮流涌现出来的。传统的“整形”几何学对于几千年的人类文明作出了不可磨灭的伟大贡献，但是在 20 世纪下半叶，它（们）已不足以揭示复杂的动力学过程和大自然的纷繁组织机制了，这时分形概念应运而生，显示了其无比的方便灵活性，甚至给人一种印象：分形概念早就该出现， 因为它太顺理成章，太符合于自然的本性了。至于何为“自然的本性”，恐怕一时无人说得清楚，人类精心创制的每一个科学概念当初不都认为符合于自然的本性吗？

新的科学概念一经被广泛接受，就同时产生方法论意义，其根据在于“相似性外推”。 相似外推是人类自童年就一再采用的最基本的认知方法。分形概念不但有相似外推、进而认 识新事物的可能性，而且为描述这种认识原则本身提供了一种“元语言”，从而具有特别的 方法论重要性。有了分形概念，人们发现世界是分形地存在、分形地演化

的（五台山南山寺 刻有“分形变化”的字句！），世界是自相似的统一体，层层嵌套，无止无休！然而，人们不会傻到相信世界是完全自相似的。分形概念、理论提供了正确认识复杂自相似结构的方法 、手段。分形概念自然引导人们思考等级层次(hierarchies)之间的“相似度”问题，哲学上的“*A=A*之含义”的问题也有了深入考察的可能了。弗雷格(G.Frege,1848-1925)率先从语言哲学角度追问“*A=A*”的含义，实际上触及了认知方法和认知的本质。 在认识新事物时，总是在忽略某些特征的前提下，把它还原为公式 *A=A*。哲学上的说法是：世上没有两个完全一样的东西；也没有两个完全不同的东西。量子力学里有“泡利(W.Pauli,1900-1958)不相容原理”；唯物辩证法中有“世界的物质统一性”这样一条原理。 其实，物质统一性原理也是自然科学原理。物质本身是有结构的，由物质的统一性可以逻辑 地推出“结构的统一”或“形式的统一”，并不能一概斥之为唯心主义。

1.3.3 作为解释工具的分形

如果说“认知方法”强调温故知新，则“解释工具”强调回顾反思，力图藉此对以往的知识 重新进行分类整理，纳入大脑的特定记忆单元，改变神经元的连接方式，绘制新的世界图景 。同一种现象或历史事实，可以有不同的描述方式，构成不同的解释系统。作为解释工具的概念，愈是简洁、自然，愈有吸引力，在与传统概念竞争中愈容易取胜。分形概念的出笼， 淘汰了许多过去复杂冗长的描述方式。同样的现象和过程一经用分形去刻画，就显得格外清晰、明了。与此同时，人们没有理

由再认为背后的机制像过去理解的那样；人们理所当然地 设想出简明的分形迭代机制。这里不想作笼统的说明，只举两个例子，作者相信用分形概念思考，可获得新的认识。解释不是最终目的，认识新事物更显重要。

例一：中国封建制的组织特征是什么？

这是一个非常老的问题，清末时的一大批有识之士就已思考它了。在今天，如果采用分形概 念，这个问题很好回答。封建制的本质特征是“家”或“家族”，英文“fueda1” 本来就有家族的义项。在中国，封建制中的“家”因素极为突出，构成了社会的最基本“元胞”， 是自相似之分形的“生成子”。由这个“家”生成元，仅仅通过尺度变换，便可非常逼真地 再现出现实的封建社会图景。在封建社会里，一切社会现象都可以从“家”得到程度不同的解释，封建制也就是家长制；家庭成员之间的关系就反映了社会机构之间的关系。

《大学》中讲“格物、致知、诚意、正心、修身、齐家、治国、平天下”，关键在“修身— —齐家”一环。中国封建社会家庭里，没有民主可言，父亲把持一切，等级森严，“三纲五常”宣扬的就是这一套东西。父亲怎样管教儿子，皇帝也就怎样管教臣宰，县令也就怎样管教百姓；兄如何训斥弟，上级也就如何训斥下级。家庭靠血缘关系维持，社会便靠“裙带关系”组构。一人得势，鸡犬升天；一臣罹难，泱及众亲。王朝虽大，不过几个家族把持。“ 家”在形式上的放大就有了不同层次的社会建制。“家”内部的单向负责关系同样在复制品中保留下

来，甚至加强了。于是，“父母官”、“国父”的说法没有任何难理解之处。“国家”在中国人眼里也是“家”！在君王、官僚眼中，国家就是“自己的家”，想怎样就怎样 ，想拿什么就拿什么，坚决不要外人插手。

从殷周时代起，形势一直没有本质上的改观。这种近似规则分形的组织结构具有“自我稳定 ”机制，王朝可以几易其主，社会形态却不变丝毫。纵观中国漫长的封建历史，多动乱却少变革；封建思想根深蒂固，业已渗透到百姓的骨髓里，想在几年、几十年之内消除封建观念，根本不现实。反封建、同封建腐朽思想作斗争必然是中国人民注定要承担的长期、艰苦、复杂的任务。改革开放的重大阻力来自封建残渣的拖曳。

有一点值得高兴，计划生育政策将根本上有助于打碎“家”元胞的结构渊薮，使“家”的规模变小，关系趋于简单化，从而令社会仅仅因这一项努力就悄悄地发生革命。这样看来，计划生育的确是“国策”！

例二：“转换生成语法”与计算机高级语言

语言的创造性总是令人吃惊，再复杂、再精致的作品也是按某些规则构成的，本质上不存在写不出和理解不了的言语。但不可小看“迭代”过程，胡奥特(J.Huarte)于16世纪就指出，表示“智力”的词ingenio同表示“产生”、“生成”的词，具有相同的拉丁词根；智能是一种生成力。在现代社会，只要多少了解一点计算机语言的无比威力，就会认为胡奥特是个天才。倘若可以揭示出语言和言语背后的生成规

则，便可构造各种人工语言，但长期以来 语言学家过分关注语言的分类和分解，忽视了语言的“句法生成”。索绪尔(F.de Saussure,1857-1913)奠定了现代结构主义语言学，却不为生成语法所理解的深层结构留下任何地位 ，认为唯一正确的语言分析方法是切分和归类，当这些分析完成后，语言的结构必然也就揭示无遗，语言学这门科学也就全面完成了它的任务。正如乔姆斯基(A.N.Chomsky,1928-)所说：“在他的用语中，造句严格讲不是语言(langue)的事情，而毋宁说是被归至他所称的言语(parole)，并因此不属于语言学本身的范围；”“从这个观点看，句法学是微不足道的。 而且事实上在整个结构主义语言学时期，人们在句法领域中也很少进行工作。”

乔姆斯基的理论可总结为：说话人的语法必须是包含有限规则的系统，它能生成无限多的、 相互联系的深层、表层结构；一套完整的生成语法必须包括句法组成部分、语义组成部分和音位组成部分。句法组成部分生成许多“句法描述”(syntactic description,简称 *SD*)，每个 *SD*有一个深层结构和一个表层结构。语义组成部分赋予深层结构一个语义解释，而音位组成部分赋予表层结构一个语音解释。语法规则可以通过如下步骤抽 取出来：

$$L \rightarrow AD \rightarrow G,$$

其中 *L* 表示“语言 *L* 的原始资料输入”；*AD*表示“一种习得语言的机械装置”；*G* 表示“能充分描写语言 *L* 的语法”。这里 *AD*最复杂，既要适

合一切语言，又要符合经验。首先要对各种语言的共性作出内容尽可能丰富的假设，其次要建立一套总的评价程序（也属于 *AD*）。

乔姆斯基的转换生成语法理论对语言现象作了恰到好处的简化、假设，取得令人瞩目的结果，与数理逻辑、人工智能研究有重要关系。其实，句法的生成就好比分形的迭代。比如，英语中的从句（子句）与主句是“结构相似”的，再复杂的复合句也可以通过语法分析弄清修饰关系和子句的具体结构，从而理解所表达的意义；同样，由结构简单的句子也能生成极其复杂的复合句，达到特殊的效果、功能。汉语一向不大注重语法，特别是缺少表达子句结构的关系代词、关系副词，致使汉语与西文相比不够严谨，中西文对译比较困难。汉语把复合句内部本应该有的严密修饰关系转移成了句子之间、句群之间甚至段落之间的关系。

数理逻辑中的语句演算形式系统 *L*[注意,数理逻辑中讲的“*L* 系统”与林德 梅叶发明的“L 系统”是两回事。]、结构化的计算机高级语言（如 Pascal 语言和 C 语言）都可以用分形概念去理解。*L* 系统的任何定理必然与其三条公理模式“相似”；同样， 任何通过“代入”规则生成的相似语句必然是 *L* 的一个定理。Pascal 语言的“过程” 、“函数”和“单元”与主程序结构上也颇相似。无论如何复杂的程序，都是一点一点“迭代生成”的。离散动力系统当然更是一种不断“代入”的系统。

由此看来，分形概念解释力很强，许多问题一旦用分形理论思考，似乎更加简单明确了。可以尝试用分形概念分析其他复杂事物。只是要留心，莫把它泛化和庸俗化。

§1.4 分形维数

1.4.1 从拓扑维到度量维

整数维数是整数，这还好理解，原来我们知道的整数维数是拓扑维数，只能取整数，维数表示描述一个对象所需的独立变量的个数。在直线上确定一个点需要一个坐标，在平面上确定一个点得用两个坐标，在三维空间中确定一个点得用三个坐标，等等。

除拓扑维数外，还有度量维数，它是从测量的角度定义的。原来的维数也可以从测量的角度重新理解。为什么要发展测量维数的定义?其实维数概念并不是从天上掉下来的，都有“操 作”的成分，都可以从操作的角度说明。学过数学的人都知道，积分理论从黎曼积分发展到勒贝格积分，就是因为引入了“测度”这一概念，这一举动克服了传统积分理论的许多缺陷 ，扩充了所研究的函数的范围和极限的意义。后来柯尔莫哥洛夫(A.N.Kolmogorov,1903-1987)将勒贝格测度引入概率论，又为概率论奠定了坚实的基础。

分数维数并不神秘。我们首先说明，从测量的角度看，维数是可变的。

看一个毛线团。从远处看，它是一个点：0 维的，好比在广阔的银河系外宇宙空间看地球， 地球的大小可以忽略不计。再近一些，毛线团是三维的球，好比进入太阳系后，乘航天飞机 在太空沿地球轨道飞行。再近一些，贴近其表面，它是二维的球面，甚至二维的平面，这好比我们站在旷野上环顾左右或者站在草原的小山丘上向四周眺望。再近一些，看一根毛线，它是一维的线。再细看，它是三维的柱体。再近一些，它又是二维柱面或者二维平面。

再接近，看毛线上的纤维，它又是一维的。再近则又变成三维柱了……

所以说对象的维数是可以变化的，关键是我们从什么尺度去观察它、研究它，一旦尺度确定了，对象的维数就确定了。反过来，不规定尺度，问一个对象的维数，其实很难回答。这正如问海岸线的长度一样,只有告诉用什么样的刻尺去测量，才能得到明确的结果。

作为整数的拓扑维，在拓扑变换下是不变的，所以拓扑学也叫“橡皮几何”,拓扑空间可以像橡皮一样任意拉伸,只要不发生粘连和撕断。对于分形对象，仍然可用拓扑变换来考察， 但也可以用别的更好的、更形象的办法考察。分形体有许多空洞，像冻豆腐一样，用空间充填的办法测度它，是一个好主意。

从测量的角度重新理解维数概念，就会自然地得出分数维数的概念.

例子 1:

一根线段 L，它是一维的，取单位长度 A，将它的线度(边长)扩大到原来的三倍，看看能得到几个原始对象(单位长度为 A 的线段)。显然得到三个：L→3L=3^1*L.

再看平面上的一个正方形 P，边长为 A，假设仍然将其线度(边长)扩大到原来的三倍，则得到 9 个正方形：P→9P=3^2*P.

对于三维空间上的正方体 V，边长为 A，假设仍然将其线度(边长)扩大到原来 的三倍，则得到 27 个立方体：V→27V=3^3*V.

得到的总个数可以表达为关系：M=B^d,

其中 B 指放大倍数，M 是总个数，d 相当于对象的维数。上式换一种写法，就有：

d=logM/logB,

其中指数 d 相当于维数。

例子 2:

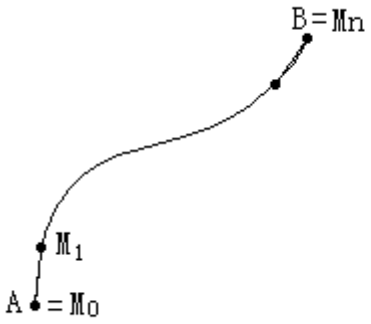
按经典欧氏几何的方法，对任一拓扑一维的光滑曲线，我们可以用折线段来逼近它，欲测出曲线 AB 的长度，可以在 AB 上取等分点 M_0=A, M_1，…，M_n=B，记 $\varepsilon_i=|M_{i-1}M_i|$ 表示相邻两等分点的直线段的长度，则有

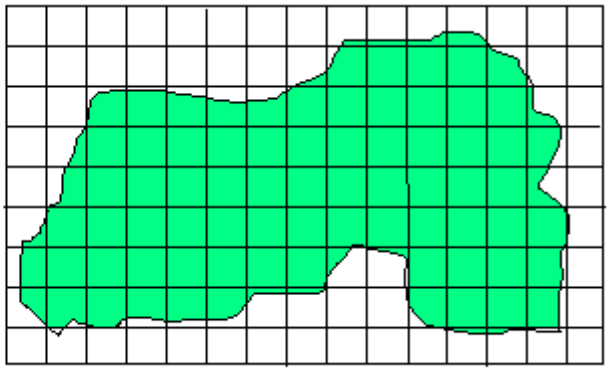
曲线 AB 的长度=lim($\varepsilon_i\to 0$) Sumε_i.

同样，对一个二维区域，如果 G 是平面上具有光滑边界的区域，构造正方形网格与二维区域 G 相交。则有

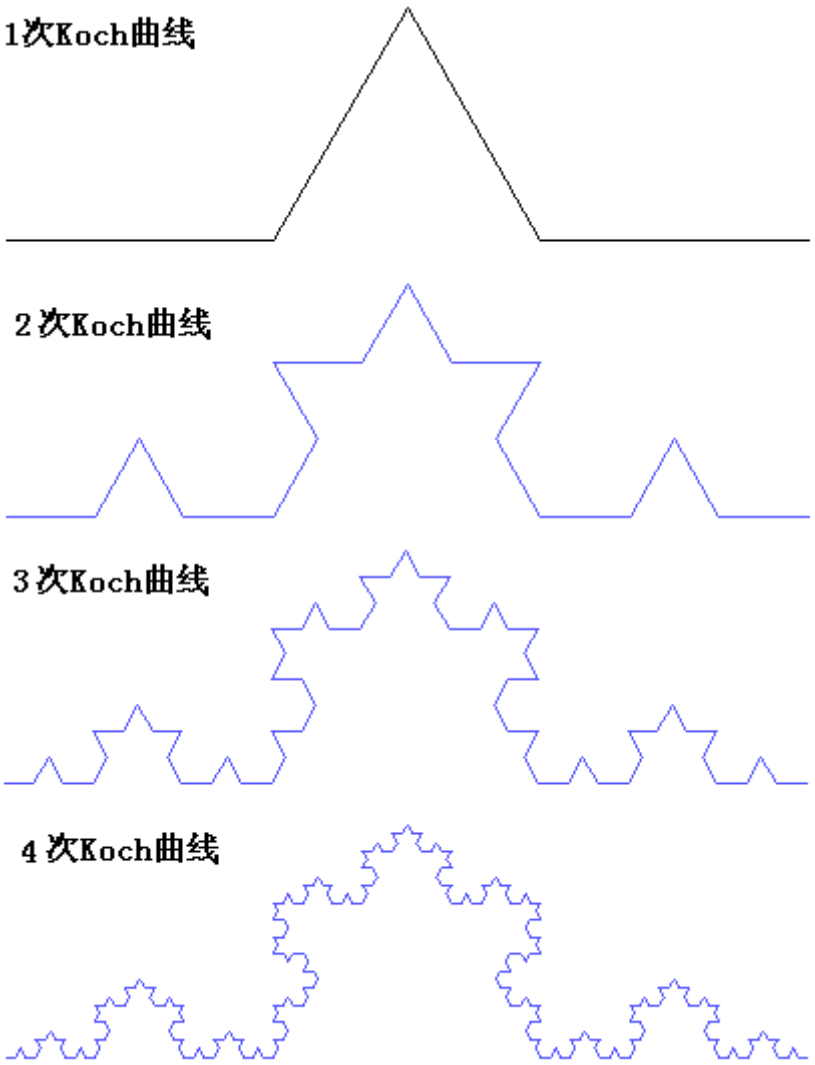
G 的面积 = $\lim(\varepsilon_i \to 0)\ \mathrm{Sum}\varepsilon_i^2$

其中 Sum 表示对所有与 G 相交的正方形求和，ε_i 表示正方形的边长。





例子 3:



现在我们用上面经典方法来求 Koch 曲线的长度。第 k 步迭代时，Koch 多边形的长度为 (4/3)^k，令 k∞→，则有 Koch 曲线的长度为∞。如果我们用二维方法度量，容易计算出它在平面上不占面积，即面积为零。可见，经典一维长度度量和二维面积度量对 Koch 曲线的大小都没有给出有效的描述，那么，有没有一个合适的度量能给 Koch 曲线这样的分形集合理的描述呢？数学家们经过研究发现，如果我们不是用整数 1、2 来度量光滑曲线和光滑二维区域的分形集大小，而是用一个合适的实数 s: 1<s<2，则对 Koch 曲线有 lim(ε i→0)Sum ε is 是一个有限数，这就是

说，这个介于 1 和 2 之间的实数 s 能更好地反映 Koch 曲线的几何特征，这个 s 更适合用来描述 Koch 曲线，这个 s 就是我们下面要讲到的分形维数。

单这样讲，大家也许还无法理解，但至少大家已经知道，通常所用的整数维已不足于用来描述分形集的复杂程度，不能很好地用来说明各种集合充满空间程度的不同，也无法很好地对比两个不同分形集的复杂程度，所以，引进分形维数就成为很自然的事。每一个分形集都对应一个以某种方式定义的分形维数，这个维数一般不是整数的，但对有些特殊的分形集，其维数也可以是整数。

在分形集维数的定义方法上，有许多数学家作出了很好的工作，出现了许多不同的定义方法，有的定义在理论上比较科学，但在实际计算上确很困难，而另一些则理论上差一些，但应用起来很方便。这儿，我们介绍三种有代表意义的分形维数：自相似维数、Hausdoff 维数和盒维数。

1.4.2 自相似维数度量

这个定义只对自相似集有意义，也只有自相似集才有自相似维数，但确最为直观，也很容易求出。设 A 是一个有界子集，可以分成 N 个相等的且与 A 相似的部分，则称 A 为自相似集。如果每部分与 A 的相似比为 r=(1/N)^(1/D)，则称 D 为 A 的自相似维数，记为 dimsA=D，即有

D=dimsA=-lnN/lnr.

对于光滑的自相似集，其自相似维数一般就等于它的拓扑维数。如一直线段A，可以n等分，每段长度为原来的 1/n，其自相似维数为dimsA=-lnn/ln(1/n)=1；对正方形B，可以分成n^2 个比例系数为 1/n的小正方形，所以其自相似维数为dimsB=-lnn^2/ln(1/n)=2.

要注意的是，对于自相似集合，分割成几部分并不是固定的，可以有不同的分割方法，但是最后求出的自相似维数肯定是一样的，不然就不是自相似集合。例如对于 Koch 曲线，可以分割成两部分，其相似比为 1/ √3,所以自相似维数为 D=-ln2/ln(1/ √3)=ln4/ln3；如果把它分割成 4 部分，其相似比就为 1/3，按定义其自相似维数为 D=ln4/ln3 是完全一样的。

另外一点，我们上面规定自相似集合的分割部分是完全相等的，这是严格自相似的情形，另一类的自相似集合是整体集合可以分割成几部分并不完全相等的与整体相似的小集合，这时的分割就不是简单的分割了，但只要适当的处理，也可以求出其自相似维数，只是计算相当困难，甚至根本无法求解，但是这类分形是确实存在的。一般来说，设某分形图形可以分割成 N_1 个相似比为 r_1，N_2 个相似比为 r_2，…，N_m 个相似比为 r_m 的与整体相似的小集合，则其自相似维数由下式决定:

$N_1/r_1^D+N_2/r_2^D+\ldots+N_m/r_m^D=1$,

特别地，$N_1=N$，m=1 时就是前面严格自相似情形。在一般情形，上式无法求解，但在一些特殊情况，其解很方便得到，例如 m=2,$r_2=r_1^2$ 时，设 $r_1=r$，$r^D=x$，则 x 为 2 次方程式 $x^2-N_1x-N_2=0$ 的解：

x=(N1+Sqrt(N1^2+4N2))/2, D=lnx/lnr.

类似的分形图有相似比不同的广义 Koch 曲线，Sierpinski 海绵，Arboresent 肺图。

相似比不同的广义 Koch 曲线

1.4.3 Hausdoff维数度量

Hausdoff 维数是从数学的测度论中推导出来的，它可以定义在任何集合上，且对于经典集合，其维数值与表示它的坐标个数相同。虽然理解它有点困难，但确是很完善的一个定义。

设 $\{U_i\}$ 为可数(或有限)个直径不超过 δ 的度量空间 (X, ρ) 上的一个子集类，A 是 X 的一个子集，若 $A\subset\{U_i\}$ ，则称子集类 $\{U_i\}$是 A 的一个 δ-覆盖。现设 A 为 d 维度量空间的一个有界子集，$s>0$，对任意 $\delta>0$，定义

$$H(s,\delta,A)=\inf\{\text{muS}\ \varepsilon^s:\ \{U_i\}\ \text{是 A 的}\ \delta\text{-覆盖}\},$$

其中 ε 表示集合 U_i 的直径，当 δ 值变小时，能覆盖 A 的 δ-覆盖类变少，因此 $H(s,\delta,A)$ 的值是不降的，于是，极限值

$$H(s,A)=\lim_{(\to\delta 0)} H(s,\delta,A)$$

存在(可以是∞)，由测度论的概念，我们就定义了度量空间的 Borel 集

类上的一个测度。

设集合 A 是 d 维是度量空间的一个 Borel 子集，对任意 s>0，称上式定义的 H(s, A)为集 A 的 s-Hausdoff 测度。

值得注意的是，对固定的 A， H(s, A)作为 s 的函数，其值域很特殊，只有两个值(0 和∞)或三个值(0、一个有限值和∞)。

在数学上可以证明：设集合 A 是 d 维是度量空间的一个有限子集，则存在唯一的一个实数 h∈[0,d]使得当 s<h 时， H(s, A)=∞，而当 s>h 时，H(s, A)=0。我们就定义这个实数 h 为集合 A 的 Hausdoff 维数，记为 dimhA=h，即 Hausdoff 维数是 s-Hausdoff 测度 H(s, A)从∞“跳跃”到 0 发生的 s 的数值。

Hausdoff 维数是一个比较合理的维数，它建立在相对比较容易处理的测度概念的基础上，在数学上可以证明它具有许多很好的性质，这儿从略。我们来看一个例子:Cantor 尘

原始图形为单位正方形，将其等分成 16 个完全相等的小正方形，并去掉 12 个使得每行每列都有且只有一个小正方形，对每个留下的小正方形施行同样的做法以至无穷，其极限集合称为 Cantor 尘。

Cantor 尘具有典型的分形性质，是一个严格的自相似分形，但其维数为整数。类似地，用前面自相似维数定义也可以求出 Cantor 尘的自相似维数也等于 1。数学上可以证明，对严格自相似集合，其自相似维数总等于 Hausdoff 维数。

1.4.4 盒维数度量

Hausdoff 维数是理论性很强而实际背景较少的维数，由于其计算要用到较为复杂的测度理论，相应的计算较为复杂而很少被人用到。在实际应用中，我们经常用到的一种维数是盒维数，它能够通过实验近似地计算，而且在一些比较规则的集合上，盒维数与 Hausdoff 维数是相同的。

直观地，从“铺砌”的角度看，对于给定的对象， 用很小的单元块 ε 充填它，最后数一数所使用的小单元数目 N。改变 ε 的大小，自然会得到不同的 N 值， ε 越小，得到的 N 显然越大 ， ε 越大，得到的 N 就越小。将测到的结果在“双对数”坐标纸上标出来， 往往会得到一条直线，此直线的斜率的绝对值就是对象的维数 d。用数学关系表达就是:

$$d=\lim(\varepsilon\to 0)\log N(\varepsilon)/\log(1/\varepsilon)$$

$$=-\lim(\varepsilon\to 0)\log N(\varepsilon)/\log\varepsilon .$$

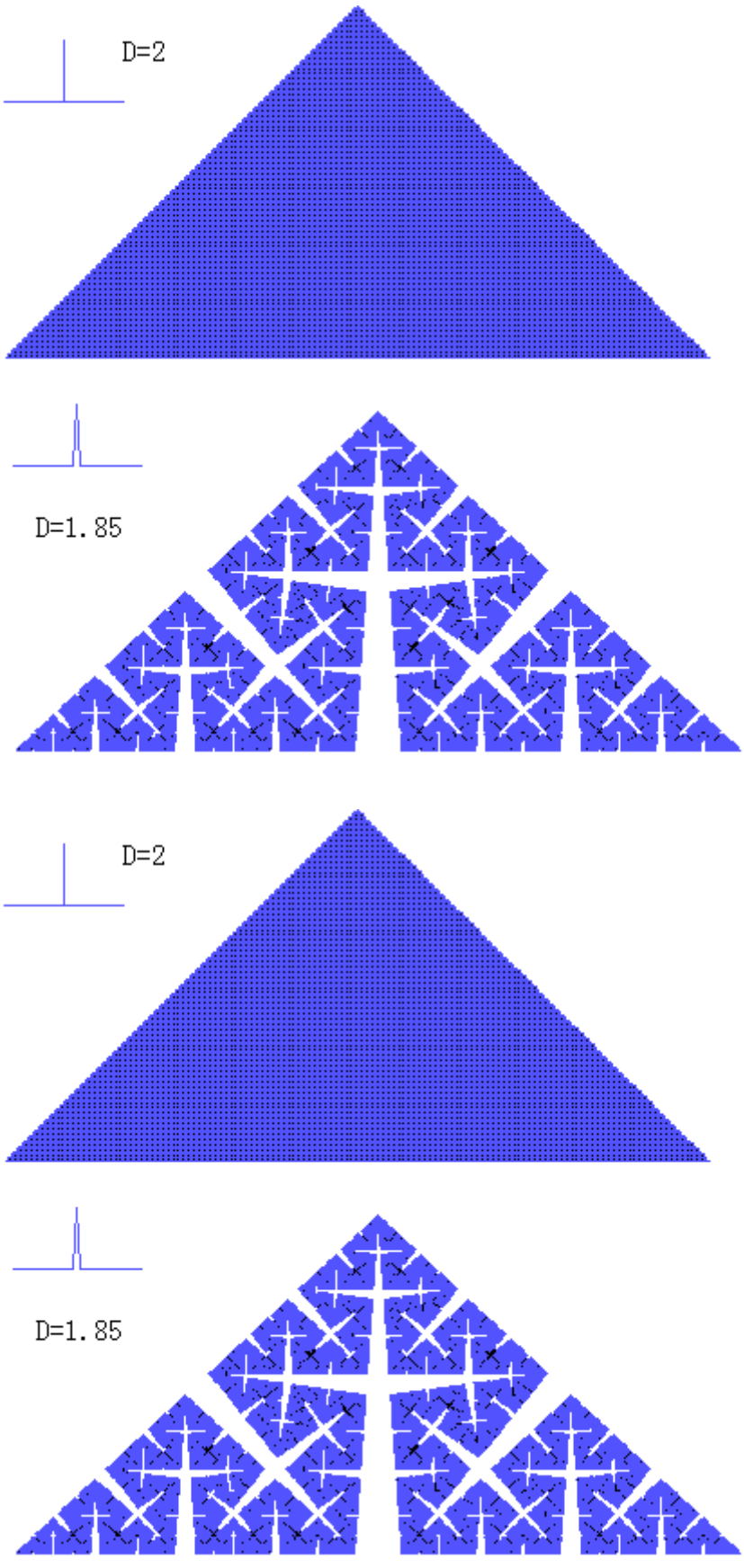
严格的定义是，设 F(X)表示度量空间 X 上全体紧子集组成的集类。A∈F(x)， (X,ρ)为度量空间，对每个 ε>0，用 N(δ,A)表示覆盖 A 的半径为 ε 的闭球的最少个数，如果 ε→0 时，b=lim{-lnN/lnε}存在，则称 b 为集合 A 的盒维数，记为 dimbA。

可以证明，盒维数有与 Hausdoff 维数很类似的性质，且可以证明对一般的紧子集 A，一定有不等式: d≥dimbA≥dimhd，其中 d 为 A 所在度量空间的维数。

在实际应用上，我们常用盒维数来作为分形集的维数，因为盒维数可以用数值实验近似求出。实验的方法是：根据实际问题的背景和尺度选定ε值的一个合适取值范围，并求出相应的N，作N－ε的重对数图，由其斜率来近似估计A的盒维数。(lnN--lnε)

值得注意的是，ε的范围与要估计的分形集的特征长度密切相关，至于ε如何取值更合适，只能依靠大家自己的摸索和经验积累，才能找出合适的标度范围。

根据前面有关维数的定义，就可以计算出各种分形图形的维数。分形图形的维数一般是小数，但其维数不应与古典维数相抵触，即用上述维数定义计算时，直线是1维的，平面是2维的。如果我们把维数看成是“复杂性指标”，则在曲线的构成中，从最简单的1维线段到最复杂的2维Peano曲线，其中间应该还有各种各样的复杂图形。以Koch曲线为例，当转角从小到大时，其分形维数也从小到大，甚至很接近于2。



关于分数维的计算，可用下面的打油诗来描述：

山重水复疑无路，

分数维数新测度。

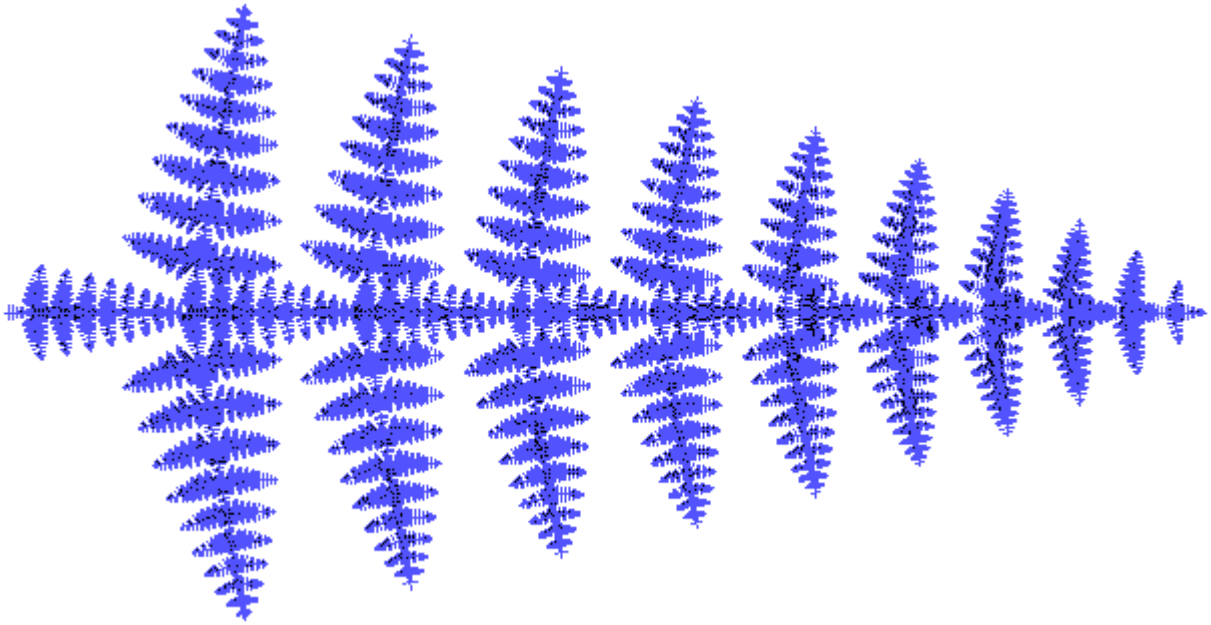
幂律关系显结构，

标度变换双对数。

§1.5 分形哲学

1.5.1 自然界中的分形现象

事实上，自然界有许多自然景物就非常象分形图形，我们可以用简单的分形程序画出一些分形，其逼真程度可以和自然界的真物照片相比，如桧树的树枝、羊凿树的叶子。



自然界由单纯的规则组成，而且这个规则涉及到自相似的所有层次，这是很自然就能想到的，这一点特性与分形非常想象，如支配羊凿树树叶的全体的规则同时也支配左右分开的树枝的一个一个小叶，而且对小

叶中的小叶也是如此。

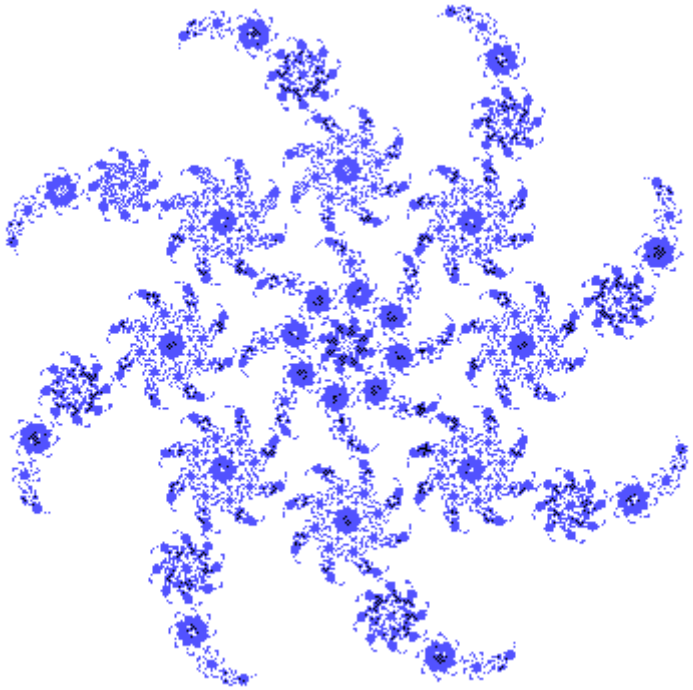
我们知道几何起源于自然界物质的抽象，我们说自然界有许多自然物体可以用分形来加以描述，如海岸线、云彩的边界。但是，应该说这些物体没有一个是真正的分形，因为用充分小的比例观测它们时，它们的分形特征就消失了。然而，在一定的比例范围内，它们表现了许多类似分形的性质，因而在这个范围内可以看成是分形。(实际上，规则几何也是理想化的产物，自然界物体中是没有真正的直线和圆的。)

早在 Mandelbrot 写书系统提出分形理论以前，他和同事 Voss 等已经在计算机上绘制了大量的逼真的月球地形、类地行星、岛屿、山脉以及类似蜗牛、水母等分形图形，这就是说，从分形开始创立时，分形就是与自然界物体密切相关的，也为人类认识许多复杂的自然界物体提供了新的工具。可以说，数学上标准的分形一开始就和自然界的现象结合在一起的。为此，Mandelbrot 猜想，自然界的许多东西都是由简单步骤的重复而产生出来的，这就使我们能够解释一些让人们困惑的事件：为什么相对少量的遗传物质可以发育成复杂的结构，如肺、大脑甚至整个机体；为什么只占人体体积的 5％的血管能布满人体的每一个部分。

单纯的东西容易反映其本性，而且也应以纯粹的形式来反映。如果我们认为分形性是自然原本生来就具有的，那么，作为同样从太古时代就有的羊齿树正好具备了充分反映自然性质的资格(据考证,羊齿树是 3 亿年前古生代石炭纪时期的主要树木)。正是因为许多基本的自然现象具有分形特征，如山脉、河流、云彩，现在有一种所谓“分形层次宇宙

论”认为宇宙就是一个分形：宇宙本身才是最能反映分形性的。这个理论的基本思想是：首先将银河系比作最基本的结构(相当于生成元、发生器)，其构成元素就是一个个星星，这些星星集中起来形成涡旋状的银河，在上一层宇宙(高宇宙)，涡旋状的银河本身又变成构成元素，从而形成更大的涡旋状银河，再进入上一层，又由这些更大的涡旋状银河作为构成元素进一步形成更加大的涡旋状银河系。

象这样重复相同规则的无限结构就表示了层次宇宙论所指的宇宙结构。如果这个理论是正确的话，宇宙本身就是一个最大的分形。



1.5.2 分形现象与生成哲学

世界是运动、变化和发展的。我们故意做点曲解，认为这句话说了三个层次上的演化问题。所谓“运动”是指宏观物体的位置移动，这个

观念主要来自经典力学，它谈的主要是量的改变，基本不涉及物体的质的改变。“变化”则牵涉到物体成分的改变，有化学过程参与，所以有质的变化。“发展”则更高一层，它涉及系统在新的层次上重新组织、自我更新、自我 否定，是真正意义上的演化，这里既有量变也有质变。研究“发展”要用到能够进行“逾层分析”的科学理论，如统计力学、系统论、自组织理论、非线性科学等等。

《道德经》中早就讲：“道生一，一生二，二生三，三生万物。”这表达的是一种朴素的宇宙生成论。最近几年中国社会科学院金吾伦(1937-)教授则提出精致的现代“生成论”，其基本观点是宇宙间的一切都是不断生成着的，世界是动态的也是整体的。生成过程是从潜存到显现的过程，“生子”是生成的因子，它具有自主性和自组织性。

金吾伦的生成论自然哲学与分形迭代生成有不谋而合之处，实际上金吾伦一直非常重视总结 、概括非线性科学理论的哲学意义。

生成哲学中的“生子”概念类似于分形理论中的“生成元”。世界在广义的迭代机制下自我进化——相当于本节开头讲的“发展”。宇宙的对称性逐渐破缺，有序性、复杂性不断增强，以至于出现山川、草木、动物、人类及人的思维。

对于生物体,20 世纪最终确立的“基因”概念加强了对生命体生成过程的理解。事实上，“ 基因”的词根与“创世纪”、“创造”、“遗传”、“生成”、“世代”都是相同的。

世界上第一只克隆羊——多利(Dolly)，诞生于英国罗斯林研究所，右侧是其代理母亲。

生命体区别于非生命体的一个主要标志是“自我复制”，[从这种意义上讲，对待人体克隆(clone)技术应慎重，不能简单地说“可以”或者“不可以”。]基因在其中担当重任。基因负责对生命体的形态、结构、功能进行全方位的编码，其信息量想必极大，但基因存在于染色体上，染色体的个数和容量是有限的，基因所包含的信息也决不会是无限的。常识的想法是，要准确描述后代生命的性状，原则上需要无穷多信

息，这在物理科学水平是无法解释通的。现在有了分形理论，这个矛盾立即消失，简单而少量的规则是可以生成复杂结构的。生命体在自我复制过程中必然大量使用分形迭代机制。

非生命界也可作类比的考虑，迭代规则是一些简单的力学、物理、化学规律。这种想法仍然是朴素的，但也许很有用。随着科学的进步，这种自然观还必然要更新，因为它太简单，在细节上一定歪曲了自然的本来面目。不过，作为一种哲学，复杂化就失去了意义。

元胞自动机、L 系统和复杂性研究都将有助于深入探索生命复制的特点，在这方面我们不得不提到先行者康韦(J.H.Conway)，70 年代他发明了“生命”元胞自动机游戏。

§2.1 艺术的含义

2.1.1 艺术的含义

书法、绘画作为艺术是没有问题的。用竹板、手指在纸上、墙壁上写字、作画，也能作为艺术。用电烙铁在木板上烧烤作画，也能作为艺术。雕塑、民间剪纸、工艺制作，也是艺术。

电视台不止一次播放过如下节目：在地上铺上画布，当众将一女性裸体粘上颜料，拉住手臂，在画布上打转、拖动，这也叫艺术。将一段海岸铺上几十米甚至上百米塑料布，这也是一件艺术品。最近还时兴行

为艺术：将一个人的动作描述下来，这一系列动作叫做行为艺术。当然，还有其他五花八门的艺术。

图中的“风”用 Photoshop 的涂抹工具加工而成

日本的黑田龙二还搞了一种“点墨艺术”。点墨艺术品的制法是：取一长方形池盆，注入 8 厘米深清水，用滴管向池水表面随机地但又有所控制地滴入五颜六色的油性彩墨，彩墨在水的表面扩散形成美丽的花纹，然后迅速取一张吸水性很强的宣纸覆于水表面，待彩墨恰到好处地吸着于纸面时，捞起宣纸、凉干，艺术品就制成了。文人墨客再在这种有精美图案的纸上书写词句或者作画，别有一番品味。据黑田龙二讲，点墨艺术的特点是，无论怎样操作， 每次都得到意想不到的图案。分形图形作品至少可与点墨画相媲美，而点墨的确被称为是一种艺术。

在普通人的眼里，到了 20 世纪 90 年代，似乎什么都可以称作艺术，如果有哪位公然敢于说某某不是艺术，通常会被认为缺少某种情调和细

胞而遭受嘲讽。用文化人的观点看，当今的艺术是艺术家的创作过程和创作结果以及作品的展播方式，一个东西、一种行为和一种现象是不是艺术，关键看当事人的态度。什么是艺术家呢?几乎说不出什么实打实的标准。首先要自己感觉像是一位艺术家，其次至少有那么一小部分公众觉得你像是一位艺术家。

2.1.2 否定计算机艺术的观点

那么，用计算机在屏幕上作图，再打印输出，如此作出来的东西能否叫做艺术品？这似乎没有什么问题，但是，还有另外一种很强的声音：什么都可以是艺术，唯独计算机作的东西不能称为艺术。这种认识有什么理由呢?总结起来看，这些理由包括:

1) 计算机是机械，它缺乏人性。

2) 计算机太灵活，它可以任意生成一大堆无意义的东西，通过计算机作出的东西缺少内涵和意义深度。

3) 计算机文件可以任意复制，计算机作出的东西不具有唯一性，因而成不了艺术品。

4) 机器创作代替了人脑的创作，作品的内容并不真正反映操作者的思想。

单纯看每一条，好像都很在理。计算机的确是机械，想驾驭计算机并非容易事，以微机为例 ，至少要先学 DOS，然后要会使用一种作图软

件，也许还要掌握高级语言。如果是自己家里组装一台兼容微机的话，还要能应付时不时出现的小故障，机器染了病毒，还要学会清理等等。

现在的计算机配备了许多现成的软件，能够自动生成许多图案，甚至随机地生成一些漂亮的画面，这种东西如果都叫艺术的话，艺术也的确没什么值得尊重的了。

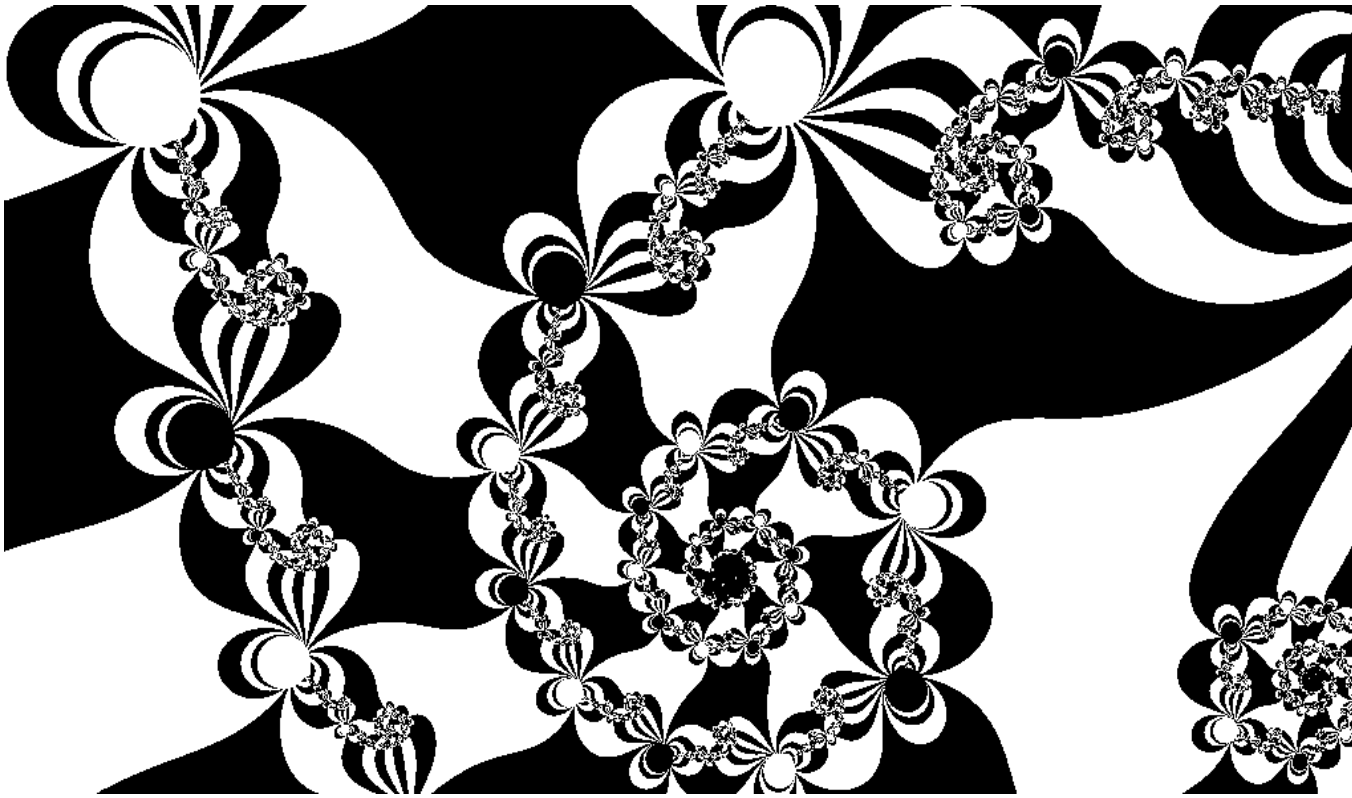
无论如何美妙的图形，一旦存成计算机文件，其信息便可分毫不差地保存在磁盘上，并可以丝毫不差地任意复制，拿到异地，存放十年、五百年也能保证不发生任何信息损失。如此看来，一件好的东西，谁都可以任意拥有，什么原件、真品，都消失得无影无踪。从商业角度看，一件投入很大的作品，可能立即变得一文不值。

严锋先生曾在 1997 年第一期《读书》上发表一篇有趣的文章《数码复制时代的知识分子命运》，对数码时代的“无限复制”表示了担忧。他问：“如果卢佛宫的名画能像自来水那样， 龙头一开，就哗哗地(免费!)流进你家，你对艺术的态度会发生变化吗?”文章最后说，在机械复制时代萎缩的东西是艺术作品的韵味，而数码复制时代萎缩的是价值，人从来都追求价值和无限，但是至少有一种无限会摧毁价值，那就是无限的复制。通过Netscape或者IE联入Internet并用过 Yahoo!搜索或AltaVista引擎的朋友多半会有同感，但这种想法过于悲观，信息时代只能改造艺术而不会扼杀、消灭艺术。而且，信息时代将比以往任何时代更能激发人的创造性，艺术世界也会更加辉煌、灿烂。

复制，这是生命得以存续的本能。诋毁复制，不但否定了生命体，也否定了人类文明。印刷 、复写、复印、照相、录音、录像、拷盘等等，哪一个离开了复制?印刷术有利于读书人， 而不利于收藏家，同样，电子图书有利于读书人，而不利于书商。信息时代、网络时代艺术品走到千家万户，感到高兴的是艺术爱好者和艺术家，前者希望看到更多更好的作品，后者希望有更多的人了解、欣赏自己的作品，这还有什么疑问吗?沮丧的可能是少数收藏家和艺术贩子，说实在的他们中不少人只是口头上关心艺术，盯住的只是其中的金钱罢了。我们为什么时时担心丢失了原著，而热衷于考据、版本和校勘?没有它们就不行吗？发达的信息社会将使传统的“知识产权”概念发生深刻的变化,现在就需要研究这种变化。

借助于软件和计算机网络，滥竽充数也时有发生，一个人本来没有什么艺术修养和技法，也能作出精致、美妙的图形，以至于真正的艺术与伪艺术难以分辨。

否定计算机艺术，还能讲理由，但是我们认为所有这些都没有真正动摇计算机艺术的合法性。所有的反驳出其他一些，也都是可以反驳的。“点墨”能称为艺术，“扔进”计算机中一个数学公式，让机器捣腾一阵，得到一幅图形，难道这不能称为艺术?



2.1.3 对否定观点的反驳

反对计算机艺术的论调在整体上是站不住脚的，它回避了艺术形式发展变化的客观进程。

回顾一下艺术史，便可以发现，艺术的范围不断在扩大。“人诗意地栖居在大地上”，人的生存本身便是艺术。这是一个信息化的时代，80 年代个人机风靡全球，90 年代计算机网络布满地球。按照尼葛洛庞帝 (N.Negroponte,1943- ,)的说法，人类越来越“数字化”地生存，这种生存必有自己独特的艺术体现。

如今五大洲各肤色各民族的百姓只有一种符号是共同的——0、1、2、3、4、5、6、7、8、9 这些数字，虽然读音不同(这些数字其实由印度人首创，但冠以“阿拉伯数字”的名字)。这种一致性是一个奇迹，是

人类共同体、人类大家庭联系在一起的一个重要纽带。各种文字对译时，以阿拉伯数字书写的内容照抄即可。中俄边贸集市中买卖双方根本不需要懂得对方语言就可做成各种交易，他们只需在手上比划一下数字。

当计算机在一般意义上成为人脑的延伸，计算机就被同化为人脑的一部分。计算机所采用的 0、1 二进制数字将对世界上的一切进行编码，“万物皆数”的古训真的成为了现实。人类的活动都将打上计算机数字化的烙印，人类将用 0 和 1 对一切事物和过程进行编码，艺术活动当然也不例外。著名艺术家康定斯基(W.Kandinsky,1866-1944)曾说:“一切艺术的最后抽象表现是数字。”这并不是科学幻想，我们正一天一天向这种世界迈进。

如果这些多少还带有猜测成分的话，我们还是直接面对现实，看看当代计算机的作用范围，特别地，看看它对于绘画艺术介入到了什么程度。

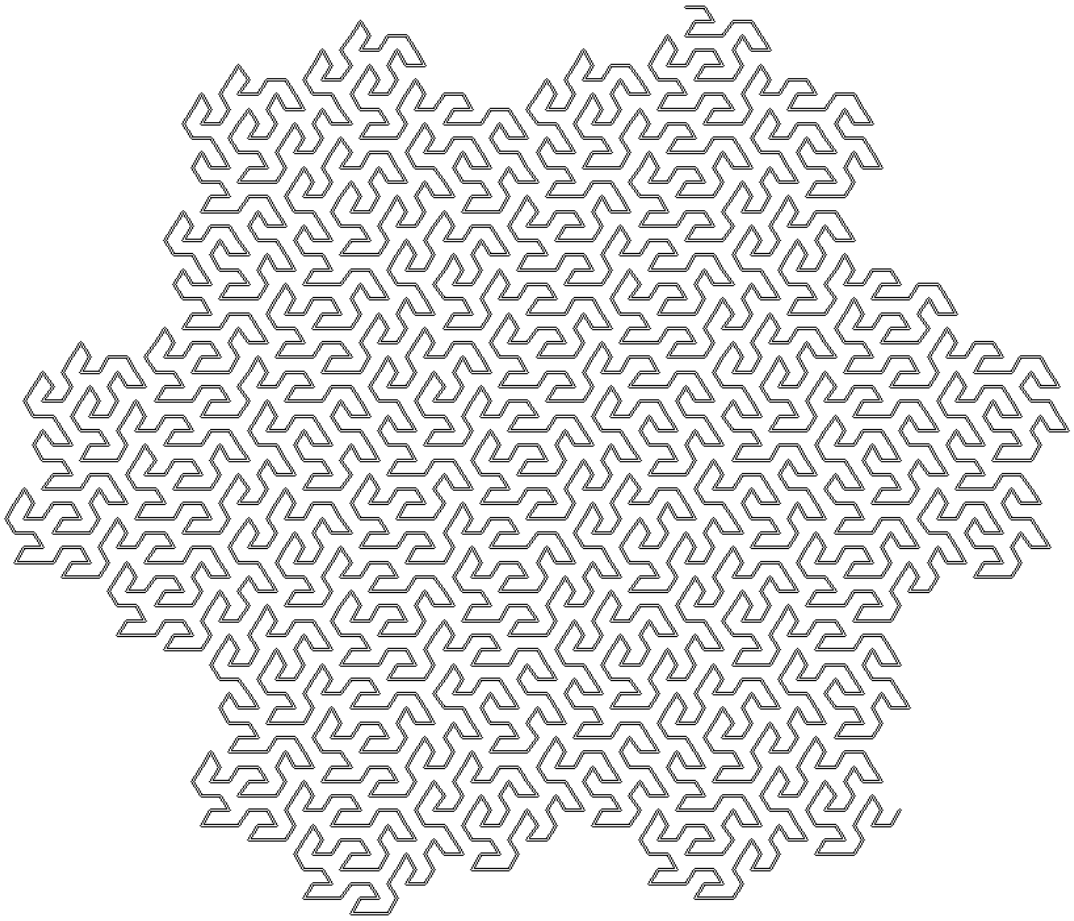
否定计算机艺术不但再现了文化保守主义的立场，也充满了对计算机图形学的种种误解。

人以工具进行艺术创作，毛笔、铅笔、钢笔、油画笔、刷子、纸团、小木片等等，都可以作为绘画的工具，原则上它们没有等级贵贱之分，不能说油画笔比毛笔优越，也不能说小木片不及铅笔，等等。基于这种考虑,计算机也是一种“平常”的工具,姑且认作它是一种平凡的工具,用它的确能够画画、能够设计图案，计算机原则上并不比毛笔优越，也不比毛笔差劲。从工具的角度看，计算机绘画只是所采用的工具不同而

已。我们没有理由认为一种采用 了不同工具而作出的绘画作品因此就与众不同，不能称为艺术品。

计算机是机器，它并不会像人一样思考，借助一些软件，操作者用计算机能够生成一大堆有趣的图形。至于这些图形是否都是艺术作品，可以区别对待。其中，好的可称为艺术品，不好的则不能称作艺术品。一旦将计算机视为等同于普通画笔的绘画工具，问题就比较好办了，运用工具能力的不同，最终会导致绘画结果的不同，笼统讲也就出现了好艺术与坏艺术。只要积累了一定经验，哪些是机器自动生成的，哪些是创作者有意创作的，还是可以分辨的 。采用固定的套路，制作雷同的东西，当然称不上艺术了。我们知道即使最普通的计算机绘图软件，都有一套“工具箱”，其中包括“画笔”、“放大镜”、“刷子”、“喷嘴”、“ 橡皮擦”、“剪刀”之类部件。 怎样合理、艺术地运用这些工具进行造型、色彩和构图设计，完全是开放的，作者可以自由设计，一旦不合人意，可以方便地进行修改，不浪费画布 或者画纸。从教学角度和节省资源、环境保护的角度看，计算机绘画是值得提倡的。

维护好自己的微机实际上也不是很难的，计算机工业发展迅速，质量越来越可靠，价格越来越便宜，买原装机是大趋势，即使组装机稳定性也有保证。操作系统也在进化，用户根本不 必知道机器内部实际在做什么，只需外在地指挥机器如何运作即可。计算机创作平台功能不 断增强，以往任何单一的绘画工具都无法与之相比。



用 L 系统方法生成的分形图案

传统艺术是具有“稀缺性”属性的，但是，我们深入思考一下，稀缺性一定是一件艺术品的 内在属性吗?回答是否定的。

艺术是美好的，美好的东西从道德的角度看，是应当尽可能与同类分享的。稀缺性只是一定历史阶段社会建构出来的价值规范，有充分的理由反驳艺术品一定是稀缺的、甚至独一无二的。难道说某种美好的东西，某种有艺术品味的作品，仅仅因为它有许多一模一样的复本， 它就不成为艺术品了吗?的确，从商业的角度看，拥有一幅与别人一模一样的作品，没有什么值得夸耀的，也不能使自己发大财，难道艺术之为艺术就是因为它能让你赚钱吗?这样做 难道不亵渎了艺术的纯洁性吗?

退一步讲，在现阶段，为照顾传统艺术，为了保持艺术的稀缺性，也是有办法的。办法有两个：第一，从作者处着眼，作者主动消除复本，即打印出作品后立即永久性地删除文件，并且不记录作品的创作过程。第二，从法律的角度着眼，通过知识产权保护，限制作品的复制，禁止以他人作品从事商业行为。

但从哲学的观点看，从艺术的本真意义看，一贯维护艺术作品的唯一性是不足取的。

计算机艺术是一个范围很广的概念，几乎所有绘画形式都能包含进来，在屏幕上能够作油画 ，也能作水彩画和国画，更能作平面设计和三维立体设计。不但如此，计算机艺术还有独特之处，计算机动画就是一种新艺术形式。计算机动画已成为像电视一样的综合艺术。多媒体技术的应用，使计算机艺术冲出视觉艺术的限制，已将听觉艺术、造形艺术和文学创作与画 面表现有机地融为一体，对几乎所有传统艺术产生了革命性的影响。近年来，由计算机动画 发展到“虚拟现实”(virtual reality，简称 VR)，艺术的天地顿时开阔了许多，非但如此，这种艺术本身也提出了一些深刻的哲学问题，加深了自中世纪以来有关“实在性”的哲学讨论。实际上，波普尔(K.R.Popper,1902-1994)讲的“世界 3 ”就是广义的虚拟现实，而我们人类与自然世界的接触过程中，总是以广义的虚拟现实作为 中介的，从这种意义上看，虚拟现实并不神秘，现有的科学技术以及文化所构造的一切观念 系统都是虚拟现实。

国际 SIGGRAPH 会议发行的录像带，从一个侧面展示了计算机艺术不断进取、五彩缤纷的局面 。有几万人参加的学术会议盛况空前，与会者多数是年轻人,这充分预示了计算机艺术朝气蓬勃的明天。当初“SIG”只是代表一个“特殊兴趣小组”，但现在由这一题目召集的计算机图形学会议差不多已是世界上最大的国际学术会议了。1996 年 10 月在浙江大学召开了中国的首届计算机图形大会(ChinaGraph),以后每两年一届，中国也有了自己展示计算机艺术的大舞台。

计算机艺术不但扩展了艺术的天地，还一定程度上改变了艺术的存在方式和传播方式。计算 机节省了人的重复性劳动，代替了一般性的平庸创作，使创作者有更多的时间发挥人的主体 性，充分用自己的大脑创造出更新更美的艺术作品。计算机艺术可以通过数字编码存贮，可以在国际互联网(Internet)上自由访问，可以让全人类共享。

总之，计算机使艺术更综合，使艺术更艺术，使艺术更具人性。计算机艺术已经如雨后春笋般涌来，看看计算机艺术展览吧，这是时代的要求，我们几乎没有选择。如果说有艺术的话 ，这才是真正的艺术。当然，前提是用计算机制作的作品有“合格的”内容，此外计算机艺术并不否认其他艺术形式。

§2.2 分形作为艺术

2.2.1 什么叫分形图形艺术

分形理论隶属于非线性科学，而非线性科学基本上属于数学和基础自然科学，主要由数学家、物理学家、力学家、化学家、地学家、天文学家、计算机科学家、工程师在做这方面的研究，此外，也有经济学家、社会学家和哲学家关心分形理论。

从原本意义上看，分形研究并不担负审美的使命，它与艺术的关系并不比一般科学与艺术的关系更具特殊性。

但是，分形几何是大自然的几何，是混沌的几何，是复杂性的几何，分形从提出那天起，它就紧紧地与艺术联系在一起。Mandelbrot 的奠基性著作充满大量分形美术插图，分形理论的实践者中从来没有人怀疑过分形图形的艺术价值。并且，这种确认是直接的而非反思的，原因在于分形图形给人的感官刺激过分强烈。科学家直觉上认定，如果这些美妙图形还称不上艺术的话，不知天下还有什么可以称作艺术。更为有趣的是，这种艺术不模仿任何自然对象， 它与现实主义无缘，但它又是一种新型的现实主义流派,因为分形图形作品包含严格确定的数学内容,这些数学绝对“真实”，在任何时代对任何人都一样。我们可以称它代表了一种 “虚拟现实主义” (virtual realism)艺术，或者叫“数学现实主义”(mathematical realism)艺术。

分形不仅仅是一个新概念，也体现一种世界观和方法论，它对于一切涉及组织结构和形态发生的领域均有启示意义，影响到艺术理论与实践是十分自然的。

分形图形艺术是计算机图形艺术的一种。计算机图形(绘画)艺术一般分作两大流派:

计算机绘画艺术 = 波普(popular)派 + 数学公式派

波普派主要通过画笔、鼠标、扫描仪等工具在屏幕上进行创作，不直接采用数学公式，虽然 计算机在后台必定一定程度上也在利用数学公式进行复杂的运算。这一派在广告界和大众层面非常流行。

数学公式派则有意识地运用数学公式进行造型、色彩和构图设计，他们像画家能够感觉到空间纵深、颜色冷暖和关系紧张一样，能够看出数学公式的内在结构以及这种结构配上色彩后所表现的热烈、庄严和静穆。

严格说来，这两者并无高低之分，两者都将长期存在。并且，更为重要的是，这种区分只是为了说明方便，实际中这种划分也有一定的任意性，两者在现实中的边界日益模糊。走相结合的道路，至少不排斥对方，才是正确的选择。

用数学公式进行创作，对作者的数理基础有一定的要求，除了懂得绘画中讲的透视关系外， 还要知道射影几何、矩阵变换、计算机图形学等方面的知识。随着软件平台的进步，对基础层次数学知识的要求会逐步降低，但不会降到不需要的程度。在中国，在世界，计算机艺术做得好的大师们，无一不精通数学。

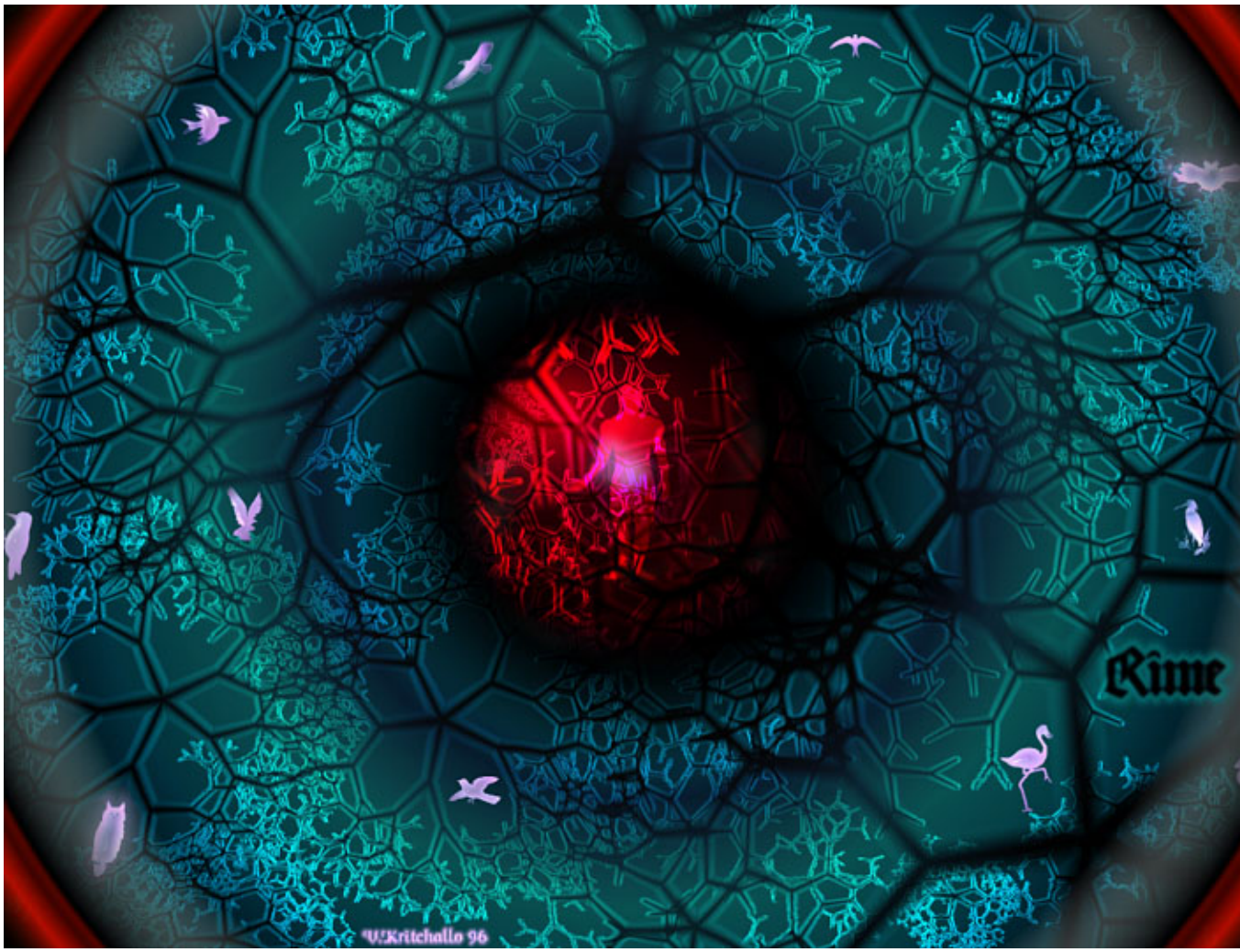
分形艺术主要包括分形音乐和分形美术两个方面。沃斯和克拉克(J.Clarke)对音乐中 *1/f* 噪声的研究已经揭示了音乐中的自相似奥妙，

并为用迭代方法、用 MIDI 系统创作电脑音 乐指明了方向。这里说的是后者——分形美术，也叫分形图形艺术，它是计算机艺术中很活跃的一支。

分形图形艺术主要是视觉艺术(当然也可以配上分形音乐)，在以上粗略的二分法中它属于数学公式派。值得庆幸的是即使用非常简单的数学公式也能生成精美的分形图形。分形的层次性和自相似性决定，分形理论用于指导装饰艺术是毫无问题的，实际上还不限于此。

定义：根据非线性科学原理，通过计算机数值计算，生成某种同时具有审美情趣和科学内涵的图形、动画，并以某种方式向观众演示、播放、展览， 这样的一门艺术叫做分形图形艺术。

这个定义稍稍窄了一些，比如根据分形原理，不用计算机，单纯通过手工复制，用手工完成计算机的迭代操作(也可以用其他办法)，也能创造出各种丰富多彩的分形图形。这种做法很有趣，读者不妨试试。看来，计算机并不是必需的，但分形思想则是必不可少的。荷兰的克里查罗(V.V.Kritchallo)用一种“与分形无关”的方法生成过一幅十分典型的分形作品《霜》(*Rime*)。



2.2.2 分形艺术的特点

分形图形不仅仅想成为艺术，以合法的身份进入艺术殿堂，还想改造现有的艺术，推动艺术的发展。所说的“改造”是以注入科学精神和现代科学知识为主要内容的。历史上科学与艺术的每一次碰撞、结合，都促进了艺术的繁荣，分形艺术以至计算机艺术肩负着在新时代促进艺术繁荣的伟大使命。

李雁在《科技日报》上发表的“科学、艺术与艺术拓扑”一文中指出：“现代美术无论是作为一种‘空间艺术’还是作为一种‘视觉艺

术’，在形态问题上，它对空间和人类视觉系统的认识都远远落后于现代科学。”现时代，科学与艺术分道扬镳，艺术落后于科学，科学也落后于艺术。艺术落后于科学是指艺术界没有把握科学的进步，一定程度上背离了理性精神；说科学落后于艺术是指科学家为利益驱使，丧失了艺术家的气质，牺牲了艺术地生存的权利，逐步异化为奴隶和机器。在20世纪行将结束之时，科学家与艺术家有相互学习的必要。

从理论上看，真、善、美的划分和指定也有一定的缺陷。一旦作出这种划分，通常人们认为科学只求“真”，宗教、伦理学只求“善”，艺术只求“美”。其实这是巨大的、不可饶恕的误解和简单化。

科学求“真”的同时，也求“善”，也求“美”。对人类有危害的科学，真正的科学家拒绝研究它，并积极组织起来反对滥用科学。科学家有探索真理的自由，但真理并不只是某种“ 符合”，科学家对社会承担责任。科学家探索自然、社会和人生，追求简单性，追求规律之美，在美与丑之间、在简单与复杂之间，科学无条件选择美和简单。伦理也不是只讲善，而不讲真和美。脱离了真，大谈善，只能是伪善。

对于艺术，“真”和“善”与“美”一样反映作品的深度和作者的旨趣，因而也是艺术好与坏的重要指标，只是这种指标不能作庸俗化、简单化的理解。艺术并不讳言模仿自然，无论是布莱克(W.Blake,1757-1827)、塞尚(P.Cezanne,1839-1906)、凡·高(V.van Gogh, 1853-1890)还是毕加索(P.Picasso,1881-1973)、柯里(J.S.Curry,1897-1946)，就艺术家本人而言，他们在自己的作品中当

时所看到的东西毫无疑问与客观事物十分酷似。几十年后二流艺术家和大众才一点一点理解到他们的作品的真实性。讲究三维画法几何透视关系的自然主义、现实主义艺术被认为追求了“真实”，但这并不代表一切真实，他们所理解的真实也 不过是人类观念的某种投射，坐标是人强加于自然的，视点更是个人主观选定的。立体派、 抽象表现主义，不但扭曲了对象，而且扭曲了坐标轴，在同一幅画面中不断变换视点，创作者认为这才是真实，这才反映了对象的全部信息，而传统的透视关系无疑是单调的、更加片 面的。

艺术也讲善，艺术家选择什么题材、反映社会的什么方面是大有回旋余地的。艺术家可以宁愿表现贫苦百姓的世俗生活，也可以专心表现帝王、公子哥、贵妇、小姐的奢侈靡烂情趣， 这体现了艺术家的伦理价值判断，所以历史上有“人民艺术家”、“无产阶级革命艺术家”之称。这些称呼并不仅仅是历史的笑料，它们的确存在过，并将默默地存在下去。

特别是当一种风格走过头时，真、善、美这三种力量的结合，有助于将脱缰的野马拉回正轨 。况且在艺术百花园中历来有自然主义、超现实主义、印象主义、抽象表现主义等几大思潮，如果仅仅从美的角度出发，把美作为艺术的唯一标准，就无法对纷繁复杂的艺术流派进行合理定位。用真、善、美的有机结合才能对重大艺术转型作出理性的评价。

阿恩海姆(R.Arnheim)说：“在印象主义盛行的同时，还存在着一种致力于挽回失去的客观 世界的另一种艺术思潮，这种世界思潮的代表人物就是鲍尔·克利(P.Klee,1879-1940)。”

真、善、美本身并不是艺术，三者以及处于张力状态的两极之间某种不可言说的“度” 才是艺术。艺术不是对称或者非对称、艺术不是真或者假、艺术不是具体或者抽象等等，艺术是两极之间的和谐。艺术是真假的合谐，是善恶的和谐，是美丑的和谐，一方衬托另一方 ，除掉一方，另一方也就不复存在。

分形方法能够表现各种和谐，分形图形艺术的兴起有助于现代科学与现代艺术的完美结合。 分形是最讲究图形的，而图形有助于形象思维，是表达事物的最好工具。

大型分形图形是有感召力的，能给人强烈的感官刺激，给人留下美好的回忆。分形之美让人思索柏拉图(Plato,前 427-前 347)宇宙的和谐和现实宇宙的壮阔，以及它们两者在大和小尺度上的无限性。

分形图形能够满足艺术作品的所有要求，能够使观众产生审美愉悦。

分形图形艺术的特点是:

第一，有科学内涵，作品有内在的数学结构;

第二，一般采用计算机数值计算;

第三，画面一般具有多重自相似结构;

第四，有后现代主义的风味，一般不强调作品的稀缺性，美感是其第一考虑。

现代绘画艺术百花齐放、光怪陆离，很难用一种模式概括，分形充其量只能概括其中的一小 部分，然而这一小部分却是重要的，随着计算机图形技术的发展，这一小部分会快速成长为 艺术百花园中引人注目的奇葩。分形艺术也必将给艺术观念带来变革，同时产生经济效益和文化效益。

分形图形艺术是伴随分形科学到来的，到 1996 年其历史不足 15 年，但发展迅速，影响巨大。

从 1984 年开始，德国布来梅(Bremen)大学动力系统计算机图形室的培特根等制作出第一批、 第二批和第三批优美的分形图片，在两位参议员的支持下他们成功地举办了一个展览。后来图片先后在英国和美国展出，引起轰动，最终出版了影响甚大的《分形之美》一书，以无可置疑的艺术美向所有专业与非专业人员展示了复解析动力系统的奇妙，该书 1987 年荣获“杰出技术交流奖”。一时间芒德勃罗集和朱丽亚集占据了校园的机时。如今大苹果机上的 Photoshop 图形处理软件专门设置了朱丽亚集和芒德勃罗集生成程序，任何非专业人员随便输入几个参数也能快速得到一幅妙不可言的分形图形。

顺便一提，西方学者常就“科学、宗教、艺术”三者来谈“真、善、美”。比如，著名科学史家萨顿(G.Sarton,1884-1956)在评论洪堡(A.von Humboldt,1769-1859)所著《宇宙》第二卷时说:

我们的科学家在某些方面才智出众，但在另一些方面却十分愚笨可笑；我们的艺术家非常聪敏，但又颇为愚昧无知。真、善、美存在，人们都能看到，但能够明白这些不过是同一秘密 的不同方面的人为何如此之少啊?……科学是生活的理智，艺术是生活的欢乐，而宗教则是 生活的和谐。缺少其他方面，任何一方都不完全。只有在这一三角关系的基础上去理解生活 ，我们才可以指望揭破生活的秘密。”

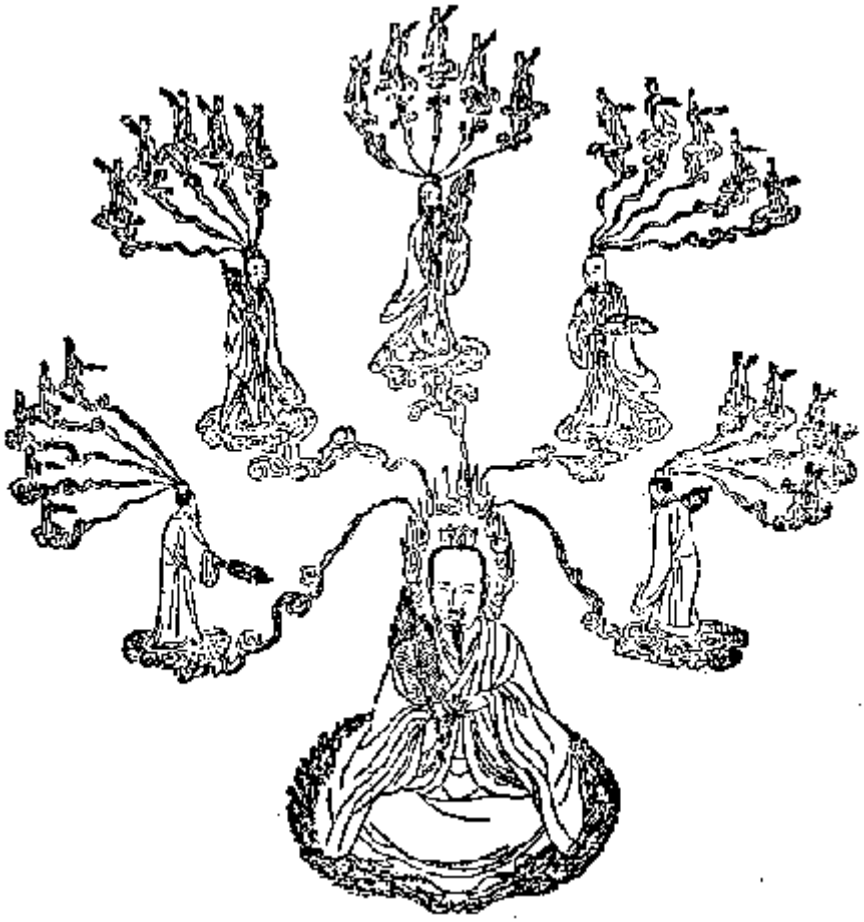
如果不算离题太远的话，我愿意提及1929年二徐关于塞尚、马蒂斯(H.Matisse,1869-1954) 的激烈争论。[徐悲鸿1929年4月22日先发表了《惑》一文,后又著《惑之不解(一)》《惑之 不解(二)》，徐志摩著《我也“惑”》，针锋相对。]这二徐一个是徐悲鸿(1895-1953)，一个是徐志摩(1896-1931)，都是顶尖的人物。前者贬塞尚，后者极力反对。抛开对错问题， 两人的讨论以及后来李毅士、杨清磬的参与，广泛涉及了真、善、美之间的关系。他们的讨论读起来仍觉亲切，现在似乎很难见到这样的对话了。特别是徐志摩的文章极其优美，容不得你不信，虽然你知道有些论述也是可反驳的。现摘录李毅士《我不“惑”》的几段话:

我想悲鸿先生的态度，是真正艺术家的态度，换一句话说，是主观的态度。志摩先生的言论 ，是评论家的口气，把主观抛开了讲话。所以他们双方的话，讲不拢来。……艺术和道德确 有深切的关系。……一个艺术家不是至诚的耶稣教徒，他决不能有好的圣母画出来，这是我 相信的。一个鄙陋的画家，任凭他技巧如何的

精妙，决不会有流芳百世的艺术品产生，这也 是我深信的。……艺术家假使没有利害的关系，那他所表现的必定是真实的，也必定是他本 性的，所以也是善的。

§2.3 分形艺术在中国

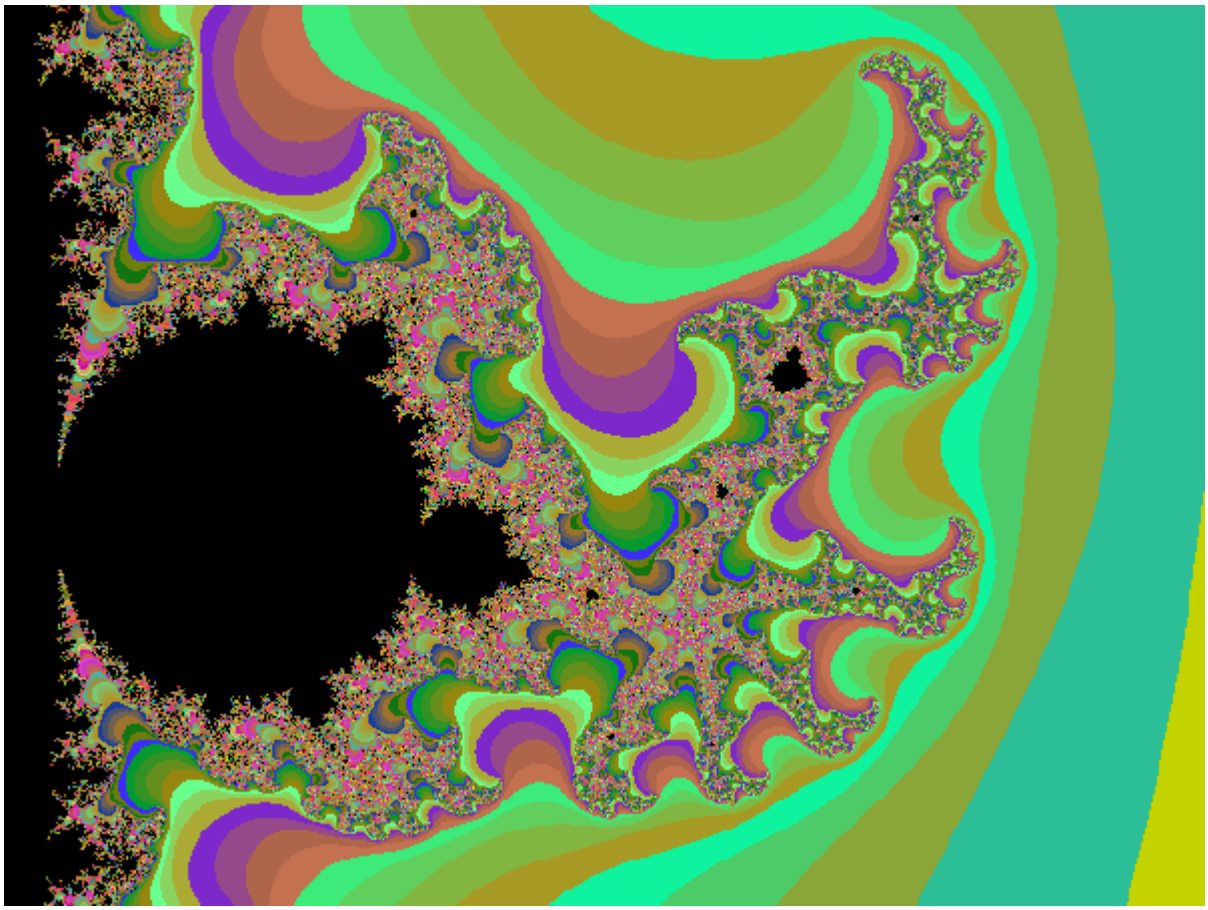
科学意义上的分形概念和分形理论是由外国人提出来的，但中国文化和古代文献中并不缺少分形观念。或者换种说法，东方人的思维方式更是分形的。中国套箱、俄罗斯娃娃套远近闻名，佛教中的无穷嵌套思想更是随处可见。从南北朝，特别是宋代开始，中国传统儒学借鉴了佛学思想，分形观念亦本土化。在中国大地上伴随佛教活动出现的原始分形艺术，中国人早已习以为常。例如长沙马王堆出土的汉墓纺织品图片上面精美的图案有敦煌“飞天”的韵味，就好似采用了分形方法制作出来的。中国佛教、道教中亦有相当艺术作品有分形味道，如元代《萨伽寺喜金刚铸像》、千手佛、化身五五图等。



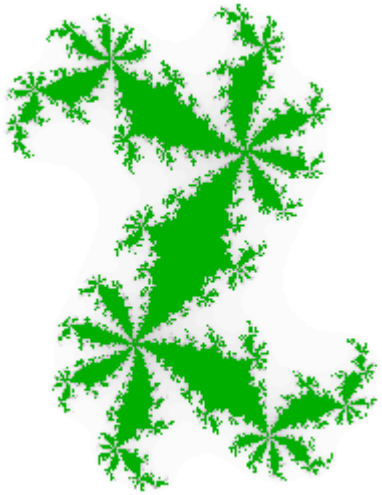
《性命圭旨》中的“化身五五图”。芒德勃罗集的局部图(见图 2.4)正好也有五五分形过程。这些图形都表达了自相似性或者“理一分殊”，“理”指一般性规律，“殊”指蕴藏着、表达着规律的具体形态。

但是，本来有着分形文化基础的国度，对当今分形艺术的接受却惊人地迟缓，其原因的确应当研究一番。分形思维是讲究生成关系并力求层次贯通的整体性思维，但仍然能分出“知性 ”的和“思辨”的两种类型，中国传统文化所具有的只是思辨的、体悟的整体性思维，而当今非线性科学揭示出来的整体性思维却是知性的。在价值层面两者无法直接对比，可是后者能够重新发现前者的意义(现在正是如此)，而前者不

能简单地、直接发展到后者，并且一定程度上拒绝接受后者。在文化对比中，这种现象是否是普遍的?抑或是作者主观臆造的?



芒德勃罗集局部的一个“五五分形”过程



朱丽亚集的形状，请与上图对照。

在国内，《二十一世纪》(香港)、《自然杂志》(上海)、《科技导报》(北京)和《科学中国人》(北京)等杂志都先后在封面上刊登过分形/混沌图形，还有一些图书也开始用分形图作封面设计,例如:《一个女医生的日记》、《伪科学曝光》、《 自然之数》、《分形与分维》、《浑沌学纵横论》、《混沌学》、《量子混沌运动》、《非线性人口学导论》、《混沌、分形及其应用》、《自组织的自然观》，还有上海科技教育出版社的一套“非线性科学”丛书、山东教育出版社的一套“新视野”丛书，国外的书籍就更不用说了。但还是如北方工业大学的齐东旭教授所讲，在国内，许多人多少年来都试图将分形图形作为艺术品推向市场，却从未坚持下来，也未见成功。

曾有人向广告公司推荐过分形图形，但反应平淡。出乎我们的意料，他们竟回答:“太抽象，我们理解不了，其他观众也不会接受的”。作者坚信这只是暂时现象，也未必代表艺术界的普遍看法。分形图形艺术在中国总有出头之日。分析中国艺术界不接受分形图形，猜测大致有三种原因:

1) 我们做的图形还不够美，还不能在第一印象上打动那些占据装饰艺术界主流的感觉良好的艺人们。

2) 在中国，科学与艺术相结合有相当长的道路要走。科学教育与艺术教育是脱节的、单向度发展的。科学界不大关心艺术，甚至认为一些人关心艺术是不务正业。艺术界一般不懂科学，更不懂非线性科学中的分形理论，需要向他们普及灌输分形知识，以求有更多共同语言 。

3) 宣传不够，没有形成规模效应，特别是没有办影响较大的展览。此外，新闻媒体的“追星”导向、广告制作的粗俗化、社会普遍的急功近利所表现出来的短视行为，也不利于分形图形艺术的出世。

1995 年《科技日报》曾举办“科学与艺术”画展，邀请了许多知名科学家与画家畅谈。不过很奇怪的是，展品中无一幅分形作品，也没有人提请艺术界关注非线性科学给艺术带来的冲击。

李政道教授近年来多次在重要场合宣传科学与艺术在绘画中结合，1996 年 5 月 27-31 日中国高等科学技术中心举办国际“复杂性与简单性”学术研讨会，李政道在致开幕词时展示了吴冠中(1919-)先生为大会所作的一幅充满分形味道的中国画。

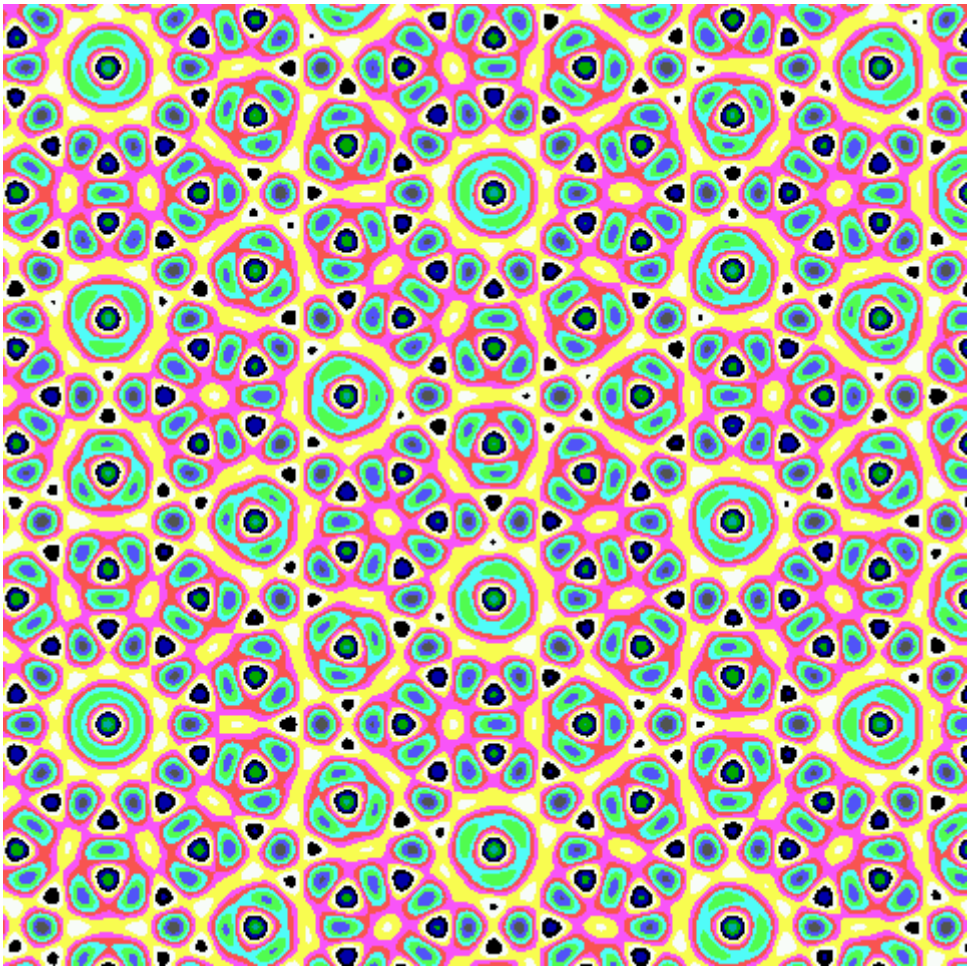
1996年5月中央工艺美术学院与北京工业设计促进会等举办“96北京国际计算机艺术”展览会，其中虽有近 20 幅作品直接采用了分形迭代技术，却无一获奖。一位评委认为，艺术界还没有跟上时代步伐，获奖的几幅作品在创意和制作技术上均无大的突破，图片既无美感更无数学深度。

§2.4 分形艺术的生成方法

2.4.1 分形图形的生成方法

艺术与科学一样都是讲求创新的，分形艺术与传统艺术相比无疑有较大的创新，同时分形艺术本身也能不断创新。我们有无穷多种函数，有无穷多种着色方案，还有若干种生成方法，组合起来简直无穷无尽，再加上每个人的创造，分形艺术广阔天地、任人驰骋。总括起来看,分形图形生成手法主要有五类:

1) 实数相空间上的非线性映射、非线性微分方程求解、保守系统准规则斑图。



准规则斑图实例

2) 复域上各式广义的 Julia 集和 Mandelbrot 集“等势面着色”方法，球面、双曲面对称图形的动力学生成。

3) 迭代函数系统(IFS)、分形插值和小波(wavelets)变换方法。

4) 林德梅叶形式语言方法。

5) 扩散置限凝聚(DLA)模型、细胞自动机(cellular automation,简称CA)模型和自组织临界性(self-organized criticality,简称SOC)方法等。

2.4.2 分形图形的输出与展示方法

分形图形艺术是一门高科技的特殊艺术，分形艺术作品的展示问题一直令人头痛，屏幕显示比较好解决，打印输出则比较麻烦。一般分形图形以图形文件形式存贮，格式主要为TIF、GIF、JPG、BMP等。其中用GIF和JPG存贮能节省大量磁盘空间，但JPG存贮是有损压缩。这些图形可以做得很小(如300×300象素)，或者做成标准VGA大小(640×480象素)，颜色有两色、16色等。但是这样的图形还不具有很强的美学感召力，要达到彩色印刷的水平，图形必须做得足够大(2000×2000象素以上)，色彩应当256色以上。输出分形图形虽然有许多方法，但都不甚理想。输出分形图形的方法主要有:

1) 针式打印(黑白或彩色)或者黑白喷墨，这是最简陋的老办法，质量也不佳。

2) 屏幕拍照，然后冲印、放大，对于大图形无效。

3) 黑白激光打印，对于彩色或者大幅面图形无效。

4) 彩色喷墨打印(如Canon,HP,Epson机等)，幅面小，色彩还原不准，用高档设备效果尚可 。

5) 热腊、热升华，质量较好(600dpi)，但成本太高，幅面小。

6) 静电印画，幅面大，精度高，成本也高。

7) NovaJet 或者 HP DesignJet 喷绘，质量较好(360dpi 以上)，幅面大(90cm×无限长)。

8)“天工”或者“御笔”等大型彩色喷绘，幅面大，精度低(30dpi 以下)，成本高。

9) 用苹果机或者 PC 机分色(一般分 CMYK 四色,有时需出专色)胶片，按传统办法彩色印刷，精度很高(胶片可达 3000dpi)。

能够制作出展览级较大图形的方法大概只有 6)、7)和 9)。

§2.5 分形艺术的发展前景

2.5.1 分形图形的发展前景

分形艺术有着广阔的发展前景，也能产生巨大的经济效益。例如:

1) 书籍装帧、杂志封面设计。

2) 广告业，作为素材制作新颖的广告画面。

3) 各种装饰艺术，如大型壁画、扑克牌、挂历、马赛克瓷砖画、居室装饰画等。

4) 纺织工业，如文化衫图案、装饰布料设计、刺绣花样设计、真丝方巾印花、时装设计等 。

目前第一方面已经开始，但经济效益不大，如果后三类中任一类发展起来，都将产生丰厚的利润。其中实施起来最容易、风险最小的可能

是向纺织行业、建筑行业推出分形艺术。但纺织印染不同于一般的彩色纸张印刷,分色、制版方面还有一些技术问题需要研究。分形图形以“比特”形式存在，但目前情况下，只有把“比特”转化为“原子”，才能为世人所接受。也许将来不需要这种转化。

无疑，分形图形艺术在中国还远未得到艺术界和公众的广泛承认。怎样促进分形艺术的发展呢?弗兰克(H.W.Franke)说得好: **“艺术与交流很有关系，也就是艺术家及其观众之间的交流。最佳的情况是使这种交流形成一个闭循环：艺术家向观众展示他的作品，激起观众的反应，他把这些反应看成是艺术效果，使自己更加了解观众，并把它作为今后艺术创作的借鉴** 。”1996 年 7 月中旬在北京航空航天大学举行的中国工程图学学会第四届代表大会决定成立“分形”和“计算机艺术”两个新的分会，对加强分形图形艺术界同行的交流将起到重要作用。

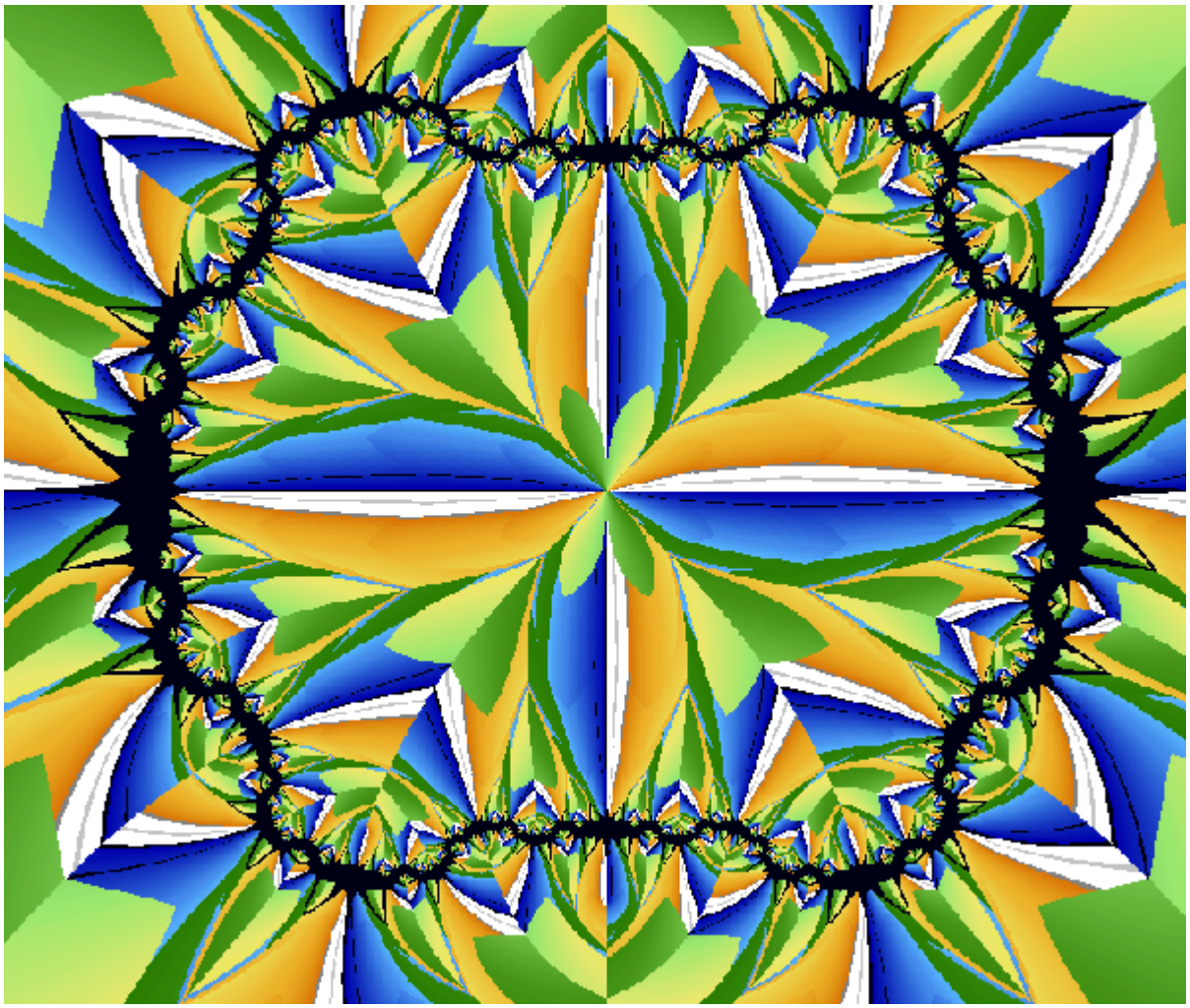
2.5.2 超大图形与装饰艺术

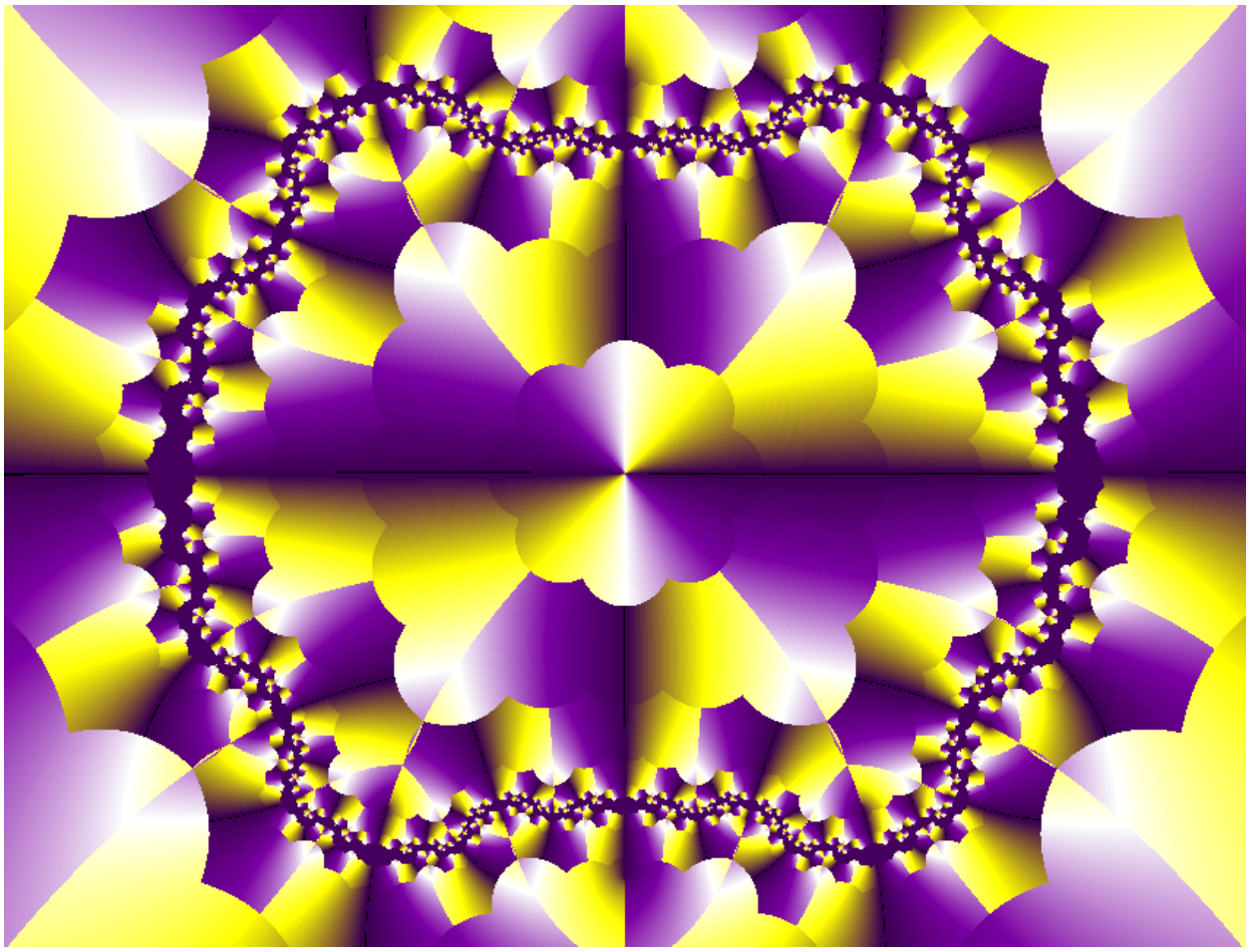
具有强烈审美效果的分形图形除了本身的内容外，图形大小也是关键因素之一。分形本身就是讲“尺度”的，分形艺术作品必须足够大，能够让观众肉眼观察到几个层次(对数尺度，一般说来需跨越 3 个数量级)。这涉及到一些计算机图形学和 DOS 文件操作的技术，目前已经基本解决了超大规模分形图形的计算、存贮和信息压缩问题。

实现分形艺术图形产业化，要完成“基础科学→应用科学→商品市场”两个转换，其中第二个转换比较难。目前中国纺织工业不甚景气，

已显示出生产过剩的迹象，将分形艺术图形应用于纺织印染也许可以给纺织工业带来一点生气。

目前国际上比较流行的分形图形创作软件有 FRACTINT。但用此软件做大图形仍然比较麻烦，最大能做 2048×2048 象素的图形，而且对显示卡及内存有特殊要求。对于大型壁画、大幅面挂历，这种精度还是不够的。





2.5.3 分形艺术与新几何学

分形图片具有无可争议的美学感召力，特别是对于从事分形研究的科学家来说。因为首先是科学家在研究中发现了美，这种美好比牛顿发现第二定律 F=ma 时、爱因斯坦提出质能方程 E=mc^2 时、杨振宁提出规范场理论时所体验到的科学美。科学不但是真实的，科学也是美的。但科学美并不是人人都能欣赏，欣赏科学之美，要有一定的科学素养。

欣赏分形之美当然也要求具有一定的科学文化知识，但相对而言，分形美是通俗易懂的。分形就在我们身边，我们身体中的血液循环管道系统、肺脏气管分岔过程、大脑皮层、消化道 小肠绒毛等等都是分形，

参天大树、连绵的山脉、奔涌的河水、漂浮的云朵等等，也都是分形。人们对这些东西太熟悉了，当然熟悉不等于真正理解。分形的确贴近人们的生活，因而 由分形而来的分形艺术也并不遥远，普通人也能体验分形之美。

说得深入些，分形之美是一种几何学之美，而几何学与艺术的关系源远流长，每一种艺术、 每个艺术流派都无法回避几何学。问题不在于是否接受几何学，是否受几何学影响，而在于接受哪一种几何学，主动或者被动吸收哪一种几何学给出的空间观念。

我们从初中就开始学习几何学，那是平面几何。那时候我们在平面上研究“点”、“ 角”、“直线”、“三角形”、“圆”、“平行四边形”等等。我们清楚地记得第一堂几何课上教师就讲到“几何点”是无大小的，“几何线”是无宽度的，两条直线只交于一个点， 后来又学到了三角形内角和为 180 度，三角形的全等与相似，圆与线的相切、相割，圆内接多边形，等等。到了高中，我们又学立体几何，从平面王国(Flatland)进入到空间王国(Spaceland)，开始在三维空间中考虑图形之间的关系，知道了异面直线，面与面的平行、垂直，双垂线定理，等等。

中学毕业后，人人都知道“点”是零维的，“线”是一维的，“面”是二维的，“体”是三维的。其实，几何学还有许多种，如解析几何、射影几何、非欧几何、黎曼几何等等，几何学研究的维数也可以超过三

维，比如研究“四维”以至“N维”，搞数学的人可能常听说“五维空间上的一个球”之类的话。

对于一般人来说，这似乎有些不可思议，但更不可思议的却是：维数不一定总取整数，也可以取分数，比如1.23维，2.48维等等。这里说的“分形几何学”常具有分数维数，这种几何学某种意义上更接近于大自然的本来面目，所以“分形几何学”常被称为“大自然的几何学 ”，有时也被称为“混沌几何学”，用它来说明混沌运动和复杂性现象特别有效。

在人类认识史上，伴随由二维到三维的转换，是林耐(C.von Linnaeaus,1707-1778)体系到达尔文(C.R.Darwin,1809-1882)体系的转换，是托勒密(C.Ptolemy,约85-165)地心说到哥白尼(N.Copernicus,1473-1543)日心说的转换。几何学影响着人们思考什么、看到什么以及对于事件价值的权衡。

在艺术领域公认有两次最大的创新，一次是**文艺复兴**，另一次是本世纪初兴起的**现代艺术**。 两次大的变革都与几何学的变革有关。前者与三维透视几何有关，后者与N维几何和非欧几何有关。

谢瑞尔(R.R.Shearer)写了一篇极有趣的文章，指出每一时代的主流绘画艺术背后都隐藏着一种深层数学结构——几何学，绘画艺术流派转换有着与波普尔-柯恩-库恩(Popper- Cohen-Kuhn)所谓的“科学革命”相类似的结构关系。在达芬奇(L.da Vinci,1452-1519)那里是讲求透视关系的射影几何学；在毕加索和埃舍尔(M.C.Escher, 1902-1970)那里

是非欧几何学；在后现代主义、纯粹主义那里也许是现在说的分形几何学，虽然艺术家们本身也许并未意识到。

套用阿伯特的用词，现在有这样一个序列转换：

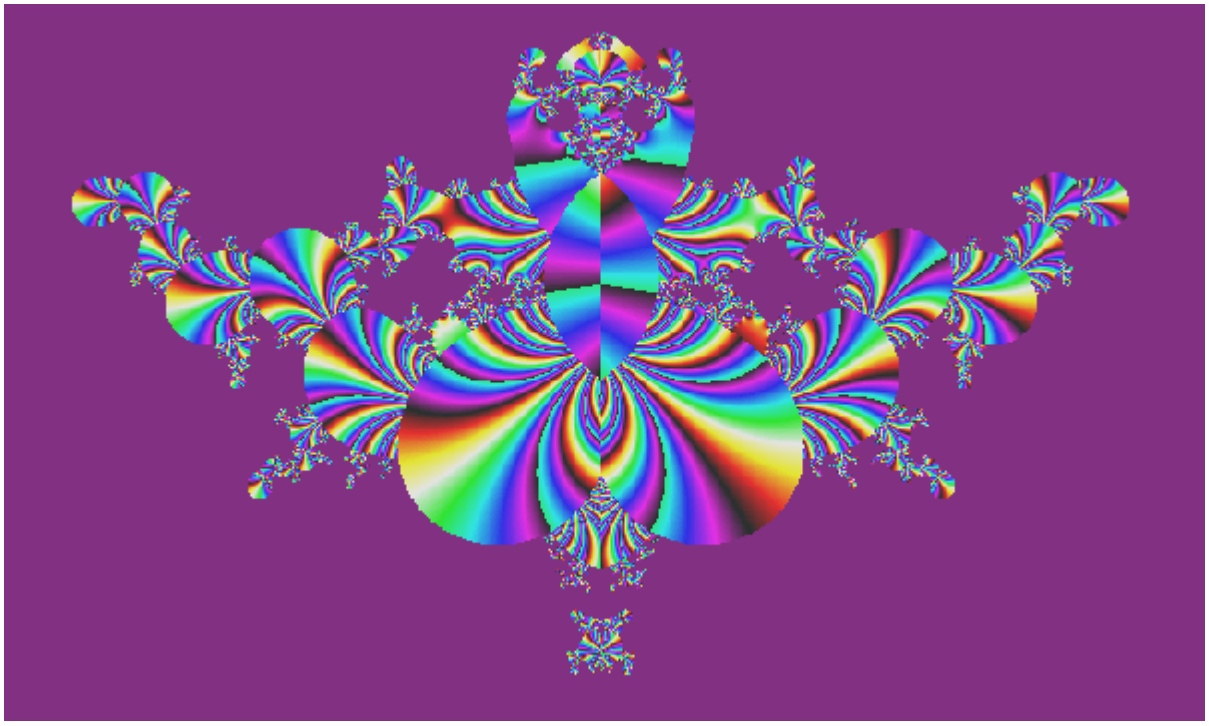
平面国(Flatland)→空间国(Spaceland)→分形国(Fractaland)。

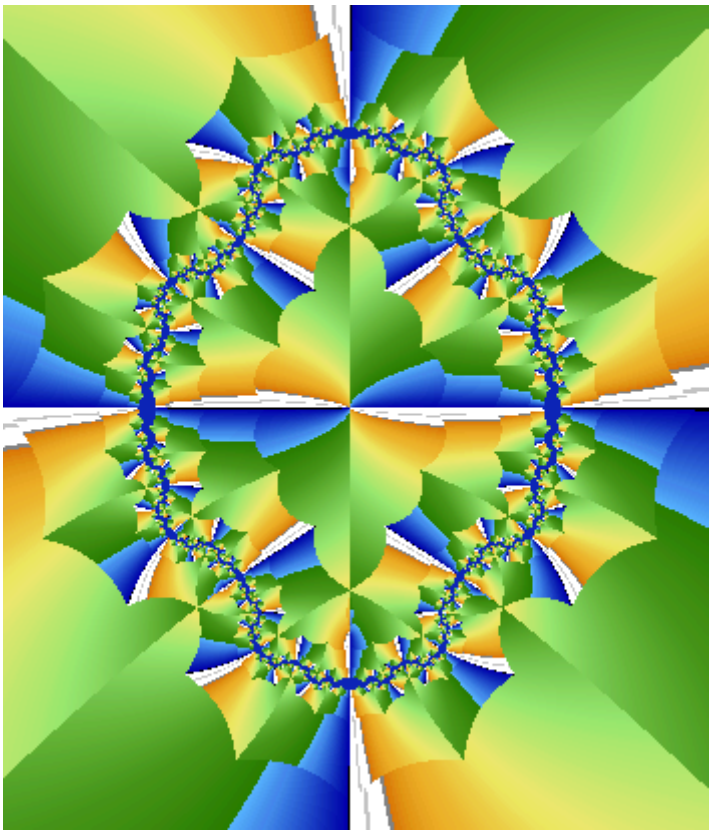
每一次转换背后都是一种几何学的转换，都代表一次革命。分形是非整数维数的对象，它不受整数维数的限制，可在分数维数空间自由飘荡。

分形几何作为一门新几何学注入我们的文化，必将引起语言、隐喻的转换，观念、方法论的 转换。从柏拉图式的经典几何到分形几何的范式转换，人们感受到了从规则到不规则、从有序到无序、从线性到非线性、从简单性到复杂性、从简单秩序到复杂秩序、从简单对称到复 杂对称、从静观到生成、从单一层面到复合层面等等思想走向。

在科学界，芒德勃罗集成了新科学的图标(icon)，这个图标既是有机的又是几何的，既是抽象的又是具体的。分形观念摒弃了传统意义的二分法，分形几何的特性总是两种极端性状的折衷、调和。从分形对象中既可以看到秩序，又可以看到混沌，并且秩序与混沌是有机地组织起来的，两者缺一不可。多组对立因素的张力和组织方式正是艺术美的体现。

分形作为一种几何影响艺术，其实刚刚开始，前景如何，人们拭目以待。





§3.1 计算机坐标

3.1.1 计算机不只会计算

计算机只认识 0 和 1 两个数，只会作加法，但它善于做大量的逻辑判断和重复性数值计算。无限重复某种操作将使系统产生“突现性”，即发生质的飞跃。1997 年 5 月 IBM 的 “深蓝”计算机甚至以惊人的战绩击败国际象棋世界最高手卡斯帕罗夫(G.Kasparov)。如今 的计算机，只有一少部分还用来做传统意义上的计算工作，相当多的计算机在做数据存贮、 查询和事务管理方面的工作。

一般用户没有必要详细了解计算机的工作原理，只需大致知道某几个部分是做什么的就可以 。比如，在 1996 年，流行的 IBM 系列微机包括主机箱、显示器、键盘、打印机、鼠标(轨迹球)、音箱等，主机箱内主要有主机板、中央微处理单元(CPU)、软盘驱动器、硬盘驱动器、光盘驱动器、内存条、显示卡、多功能卡(有的已集成在主板上)、网卡、调制解调器、声卡、解压卡。

微机的部件可简单分作输入设备(键盘等)、输出设备(显示器、打印机)和存贮设备(磁盘、 内存、光盘等)。

内存与磁盘的区别在于，前者只是临时存贮信息，关机后信息消失，后者永久性存贮信息。 内存存取信息速度较快，磁盘存取信息相对慢些。软盘存取信息比硬盘和光盘都慢。

一个英文字符(或者英文标点符号、数字)存贮时占 1 个字节，即 8 比特。汉字是双字节字符，每个汉字占两个字节，即 16 比特。这样，一张 1.2M 的软盘，原则上能存贮 60 多万个汉字。

用户一般不直接控制计算机硬件，而是通过操作系统(operating system)这个媒介间接指挥。开机时，系统要引导(boot)操作系统，即把操作系统的主控部分调入内存，形成一个外壳 ，或环境，以备其他软件的运行。也就是说，其他任何程序都是在操作系统的基础上运作的 ，它们必然依赖于操作系统。举例说，用计算机打字、编制表格、开发应用程序等，实际上已经使用了某种软件，但此种软件不是最基本的，它们要靠机器所配的操作系统才能运行。

3.1.2 操作系统与文件

IBM 微机用户使用的操作系统主要有: 1)DOS(磁盘操作系统)，这是多年来传下来的，已形成一定的惯性; 2)Windows 3.X,现在已无人用; 3)Windows 95/98/2000。有趣的是，这三个平台即使在一台机器上也常常是并存的，一个用户一会儿用 DOS，一会儿用 Windows 3.X，一会儿用 Windows 95。但是,无论用哪种操作系统，DOS 都是基础，知道一点 DOS 命令不会有害处。

DOS 是一种最常见的操作系统，它是英文“Disk Operating System”的简称。目前几乎所有微机都使用 DOS,但不要产生一种误解: 以为 DOS 是唯一的操作系统。事实上它并不是计算机的唯一操作系统，也不是微机的唯一操作系统。比如 UNIX、OS/2、Windows 都是操作系统。目前苹果公司的 MacOS 7.6 已面世，它具有完善的访问 Internet 的功能，内建 OpenDoc 支持，最新多媒体界面，增强的 DOS 和 Windows 文件兼容性等。

DOS 是一个不断升级变化的软件,在升级过程中曾吸收了 UNIX 系统的许多优点。其实 DOS 也不直接指挥硬件，它是通过固化的 BIOS(基本输入输出系统)来管理机器的。刚买来的微机自带的直接与操作系统打交道的软件是 BIOS，BIOS 装在主板上。于是有如下基本关系:

应用程序→DOS: 磁盘操作系统→BIOS: 基本输入输出系统→硬件

微软(Microsoft)公司掌握着 DOS 的命脉。1981 年 10 月，IBM 把微软为 Intel 8086/8088 微处理器研制的 MS-DOS 改用于 PC 机,推出第一个 DOS 版本 DOS1.0。后来一直有 PC-DOS 和 MS-DOS 两大族类(二者兼容)。到 1993 年 MS-DOS 已有 6.2 版本，1994 年已有 6.22 版。

几种重要版本的 DOS 推出时间及性能说明如下:

○PC-DOS 1.0，1981 年 10 月，支持单面软盘，为 PC 机第 1 个操作系统。

○PC-DOS 2.01(=MS-DOS 2.0)，1983 年 3 月，支持带硬盘的 PC/XT 机。

○DOS 3.0，1984 年，支持 80286CPU 的 PC/AT 机，1.2MB 软盘。

○DOS 3.2，1986 年，支持 3.5 英寸 720KB 软盘。

○DOS 3.3，1987 年，支持 3.5 英寸 1.44MB 软盘，32MB 硬盘。

○DOS 3.31，1988 年，首次开始支持大硬盘分区。

○DOS 4.0，1988 年，支持 2GB 硬盘分区，支持 EMS 4.0 内存管理。

○MS-DOS 5.0，1991 年 7 月，DOS 发展史上最重大的一次革新。支持 2.88MB、3.5 英寸软盘 ，全面的扩展、扩充内存管理，外壳(DOSSHELL)，

全屏幕编辑(Edit)，QBasic,失误保护(Un delete 和 Unformat 命令)，Help 等等。

○DOS 6.2，1993 年，文件同名覆盖提示，硬盘翻倍，在线帮助，多功能备份，磁盘整理(ScanDisk 命令)，反病毒。

皮亚诺曲线(前四步)围成的区域

虽说 DOS 版本越高，功能越强(到 1995 年 MS-DOS 已有 6.22 版，PC　DOS 已有 7.0 版)。但这并不一定意味着版本越高越好。DOS 版本应与应用程序相配合。应用软件在某一版本的 DOS 下运行得最好，那么对此应用程序而言此 DOS 版本就是最好的。注意，一味想用高版本的 DOS 是不足取的，一些老软件在高版本 DOS 下可能根本就不能运行。因此，用户最好能保留两到三个版本的 DOS，以备使用。

操作系统有 5 大管理功能: 处理器管理、存储管理、设备管理、文件管理、作业管理。一般用户接触最多的是文件管理。文件(file)指一个具有符号名的一组相关元素组成的有序序列 。文件名是文件的标识，正如人名是人的标识一样。多个人叫一个名，就会引起许多麻烦， 同样，多个不同的文件取了相同的名字也会造成麻烦。

当然，不同“路径”(path)中，文件可以重名。比如，D 盘 D:\\KA 子目录下与 D:\\KB 子目录下可以同时有名字为 MYFI.TXT 的不同或者相同的文件。

DOS 所能管理的文件称 DOS 文件，简称文件。本书中所有文件都指 DOS 文件。DOS 文件的命名要 遵守一定的规则。DOS 文件全名由两部分构成: 文件名(filename)和扩展名。其中文件名长 度为 1-8 个单字节(byte)字符，也就是说用英文或数字最多可有 8 个符号，用汉字最多可用 4 个(一个汉字占两个字节)。注意，Windows 95 放松了对文件名长度的限制。文件名可用字符 有 26 个英文字母(大小写不作区分)、0-9 十个数字、一些特殊符号(如%,!,@等，但不能用控 制符和输入输出定向符号|等)、汉字和一些汉字部首。扩展名最大长度为 3 个字节。扩展名 可有可无。比如 README.DOC、COMMAND.COM、CONFIG.SYS 等都是合法的 DOS 文件名。

应当说明的是扩展名有特定的含义。通常，看文件的扩展名，就能猜出此文件的类型。DOS 文件的扩展名遵循的约定如下:

○*.COM——内存映象文件，可在 DOS 命令行执行;

○*.EXE——浮动二进制代码文件，可执行，并可重定位;

○*.SYS——系统配置文件;

○*.OBJ——目标程序文件;

○*.BAT——批处理文件;

○*.BAK——备份文件;

○*.TXT——文本文件;

○*.DOC——资料文件;

○*.DAT——数据文件;

○*.OVL——程序覆盖文件;

○*.HTM——超文本文件;

○*.C——C 语言源程序文件;

○*.CPP——C++语言源程序文件;

○*.ASM——汇编语言源程序文件;

○*.BAS——BASIC 语言源程序文件;

○*.FOR——FORTRAN 语言源程序文件;

○*.PAS——PASCAL 语言源程序文件;

○*.COB——COBOL 语言源程序文件。

初学者记住这些约定是很有好处的。上面用到的“*”，是一种通配符，可代替一个或多个字符，最多可代表 8 个字节的字符(8 个 ASCII 或 4 个汉字)。图形文件扩展名的约定见后面。

另一个通配符是“?”，它只能代替一个字节的字符。比如MYAB.EXE、QUIZ.EXE、MAJUNG.EX E、VPIC.EXE 等都可用*.EXE 代表; AS.WPS 和 AS.BAK 都可用 AS.*代表; ACS.WPS 和 AMD.WPS 可用 A??.WPS 代表。通配符用处很多，应当牢记，但尽量少用*.*形式的通配符，它可代表任何 DOS 文件。

DOS 文件有一定的逻辑结构和物理结构。逻辑结构上分: 1)记录式; 2)流式。物理结构上分三种: 1)连续结构; 2)链接结构; 3)索引结构。

逻辑结构常常是文件呈现在用户面前的表观形式。背后的物理结构用于实现表观的逻辑结构 。同样的逻辑结构，背后的物理结构可以是多样化的。

一张准备好的磁盘，即格式化后的磁盘，在其上分出了 4 个区域:

1)引导区(boot sector): 为保留区，用户不能访问。

2)文件分配表区: 文件分配表(file allocation table,简称 FAT)是 DOS 建立的一张记录正在 使用的、可用的以及损坏的磁盘扇区情况表。每张盘上有两个一模一样的 FAT，如果一个损坏，可以通过工具软件(如 Norton Utilities)用备份表修复。

3)目录区: 存有文件目录表(file directory table,简称 FDT)，存贮文件名字、扩展名、属 性、时间、日期、首簇位置、大小等等。

4)数据区: 用来存放文本文件、数据文件。此区域由 FAT 管理分配和释放。

前三个区与整个盘相比所占比例较小，但很重要，它们是计算机病毒重点攻击的对象。初学者一般可不必了解前三个区的细节。

在 DOS 命令行上可执行的 DOS 文件有三种: *.COM、*.EXE 和*.BAT。

DOS 文件都有自己的特定属性(attribute)，可用属性参数有三种: H(隐式文件参数)、S(系统文件参数)、A(档案文件参数)。它们组合起来使得 DOS 文件可有 16 种属性。

N=C^0_4+C^1_4+C^2_4+C^3_4+C^4_4=1+4+6+4+1=16

DOS 引导失败有许多种原因。如果用 A 驱引导首先要检查 A 驱是否有引导盘，以及驱动器门是 否关上了。如果硬盘引导失败，可用 NORTON 软件的 NDD 检查一遍。附带说明一下，一张可引导盘必须有三个文件: 两个隐式文件 IO.SYS 和 MSDOS.SYS，一个命令解释文件 COMMAND.COM。但有此三个文件未必一定能引导。若想把磁盘格式化成可引导的磁盘，在操作上有特殊要求 ，可用 FORMAT 命令、SYS 命令和 DISKCOPY 命令。

运行一些程序可能随时要重新读 COMMAND.COM，若机器是用软盘引导的，系统每次总是向 A 驱 动器找 COMMAND，如果找不到则报错，所以引导盘要经常放回 A 驱动器。如果用硬盘(只能是 C 盘)引导，则十分方便，每次需要 COMMAND.COM 时，向 C 盘根目录区寻找就是了。

3.1.3 计算机屏幕坐标

自然界本身没有坐标，人类发明了坐标。坐标定义了一种参考系，在此参考系下，空间的对称性降低了，出现了上、下、左、右之分，出

现东、南、西、北之分，当然还有人们最熟悉的笛卡尔(R.Descartes,1596-1650)坐标系下的四象限之分。

有了坐标系，图形上的点就有了确定的位置。对于平面而言，确定平面上的一个点需要两个参数,或者叫两个坐标,可以用(x,y)或者(ρ，θ)表示。

从初中起，人们就熟悉了平面直角坐标，也叫平面笛卡尔坐标，实际上这是人们使用最多的一种坐标。它的特点是：1)有两个坐标轴；2)坐标轴是相互垂直的。到了高中，又学了立体几何和立体解析几何，知道了三维笛卡尔坐标，这时的坐标系是立体的，坐标轴仍然是相互垂直的。

然而，坐标系未必都是这种模样，高中还学了极坐标，大学还讲了柱坐标和球坐标，实际上还有许多种可能的坐标。只要给出一种对应关系，就能定义一种坐标系，甚至不必要求是一一对应。

了解不同坐标系之间的变换关系，是计算机图形学的必备知识。

几何学上我们可以轻松地谈论三维几何体、四维几何体以及N维几何体，但在计算机中，至今人们只能在二维平面上做事情，所有图形不论是多少维的，都要投影到平面上。至于怎样投影，就需要研究了。对于比二维多一维的三维对象，已有了一整套投影方案，但对于更高维的对象，还没有十分通用的投影办法。

对于用PASCAL语言或者C语言作图而言，屏幕坐标采用平面直角坐标，但坐标原点不在屏幕中心，而在屏幕左上角。由左上角向右为横

轴正向，由左上角向下为纵轴正向。这样规定主要是为了屏幕滚动方便，因为屏幕总是向上翻。

如果用户不习惯这种坐标，也可以方便地转换成通常的直角坐标，即纵轴以向上为正向，横轴不变。只需把纵坐标反射一下，即可相互转换。

比如，假设计算出来的坐标值为(a,b)，为了得到与通常坐标一样的显示结果(即纵轴向上为坐标增加)，则可以用(a,const-b)在屏幕上描点，其中 const 是一个常数，可以取 300，也可以取 200，你可以任意给定一个常数，这要看显示器和显示卡的类型，以及你本人想将此点大致描在屏幕的哪个位置上。

§3.2 色彩与图文件格式

3.2.1 孟塞尔标色体系及其他

在计算机上作图不仅仅要作黑白图形，也要作带有颜色的彩色图形，因此总要谈到色彩，实际上色彩问题很复杂。色彩是美术讨论的中心问题之一，它涉及心理学、物理学、生理学、 美学和艺术理论。色彩学与透视学、艺术解剖学一起构成了美术的三大基础理论。

色彩学是一专门的学问，美术系的学生一般要把它当作一门重要课程来学习，并在实践中逐步摸索掌握色彩鉴别、对比、调和的规律。目前，色彩构成与平面构成和立体构成一起成为视觉艺术的基础理论课程。在色彩学上作出重大贡献的人物有牛顿(I.Newton，1642-1727) 、

亥姆霍兹(H.L.F.von Helmholtz,1821-1894)、孟塞尔(A.H.Munsell,1858-1918)、黑林(E.Hering,1834-1918)、奥斯特瓦尔德(W.Ostwarld,1853-1932)等。值得一提的是亥姆霍兹，他在大量生理光学实验的基础上，于1856-1866年出版《生理光学》一书，发展了杨(T.Young,1773-1829)于1807年提出的色视觉理论，创立了著名的“杨亥三色理论”。 他认为，对颜色作出不同反应的视网膜内有三种不同的神经纤维。

关于标色体系，在美术界和在科学界仍然有不同的认识，这导致在标色体系上有许多系统， 最有名的有三种：1)孟塞尔体系；2)奥斯特瓦尔德体系；3)电视、计算机色彩体系。本节简单介绍前两种，只需了解几个基本概念，它们对于计算机艺术关系不大。

孟塞尔体系在美术界很有影响，此体系最早由孟塞尔于1929年提出，1943年又经修正，并得到美国光学学会(Optical Society of American)认可，成为国家标准，对美术界和工业界影响巨大。

孟塞尔体系又称HVC体系，因为此体系将色彩用三个要素：“色相”(hue)、“明度”(value)和“彩度”(chroma)表示。

色相(H)是指红、橙、黄、绿、青、蓝、紫等不同颜色。

明度(V)是指色彩的明暗程度。对于物体来说也称亮度、深浅程度，对于光源来说也称光度 。比如同样是“红”颜色，还有深浅之分，可以分出不同的明暗层次。对于“灰白”色，可以分出由“黑”到“白”若干等级。

彩度(C)是指色彩的纯净程度，也叫纯度、饱和度。从物理光学看，波长单一的光，彩度值高，波长混杂的光，彩度值较低。

三者的关系是：不同色相(H)的色彩相加，明度(V)提高，但彩度(C)降低。孟塞尔体系常用一个圆柱表示：圆柱的高由下至上表示明度(V)增加；圆柱的圆周表示色相(H)，沿圆周循环；圆柱的半径由内至外表示彩度(C)增加，至圆周处彩度最高。孟塞尔体系的好处是对色彩的划分十分详细，与人的感觉基本一致，缺点是现实世界中的色彩并不能填满 HVC 圆柱体。

另一种使用很广的色彩体系是由诺贝尔化学奖获得者、德国物理化学家奥斯特瓦尔德发明的。他以荷林的生理四原色：黄(yellow)、蓝(ultrablue)、红(red)、绿(seagreen)为基础，将四色放在等分的圆周上，相对的颜色互补。然后再在两两之间增加橙、蓝绿、紫、黄绿四色。这样一共有 8 个基本色相，然后每一个再一分为三，一共得出 24 色相。从圆环上看，相对的两色总是互补的。

在奥氏体系中，用复圆锥表示各种色彩组分，上圆锥的上尖用 W 表示，代表“纯白”、下圆锥的下尖用 B 表示，代表“纯黑”，WB 连线是复圆锥的轴线，此轴线为无彩轴线，彩度最小 。由此轴向外，色彩纯度增加，圆周边缘彩度最大，用 C 表示。复圆锥由下至上，明度增加。复圆锥的每一部位都用 WBC 三个值表示，并规定 W+B+C=100。再加上色相(H)，奥氏体系有四个参数，实际上只有三个是独立的。

著名的 Photoshop 软件用四种方式表示色彩，从事计算机艺术创作有必要了解这几种体系(模式)：

1)HSB 体系。类似上面提到的孟塞尔体系。这里 H 指色相(hue)，S 指饱和度(saturation),B 指亮度(brightness)。

2)RGB 体系。常用于加法混色。

3)CMYK 体系。常用于减法混色。

4)CIE 体系。CIE(国际照明委员会，法文全称为“Commission International d′Eclairage ”)体系是 1931 年建立的一种色彩测量国际标准，1976 年修正为 CIE L*a*b*。此体系用三个参数，一个是亮度 L(luminance)，另两个是颜色分量，分别为 a,代表从绿(green)到红(red) ，另一个是 b，代表从蓝(blue)到黄(yellow)。在电视工业中，为了配色方便，CIE 制定了 XYZ 计色制。

3.2.2 色彩与RGB值

尽管关于颜色有各种物理的理论和心理的理论、主观的理论和客观的理论，但在计算机中颜色是用 RGB 值刻划的，显然具有客观性。这里讲的与色彩有关的颜色问题与美术课的讲法略有不同，这是首先应当注意的。实际上用计算机作图，只需记住 RGB 三个纯客观的值，其他概念都可由此导出。

在计算机中用三维数组或者向量表示任何一种颜色，设三维向量为(R，G，B)，R 代表 red(红)，G 代表 green(绿)，B 代表 blue(蓝)。这里的红、绿、蓝叫做三原色，用它们的组合可以表示任何一种颜色。

R、G 和 B 的取值范围都是 0 到 255，各有 256 种取值。我们通常说的“纯红”，用 RGB 值表示就是(255，0，0)；“纯绿”用 RGB 表示为(0，255，0)；“纯蓝”用 RGB 表示就是(0,0,255)。

通常说的“黄”色用 RGB 表示则为(255，255，0)；“青”色用 RGB 表示则为(0，255，255)； “粉红”色用 RGB 表示则为(255，0，255)。

任意取三个值，比如(10，221，34)，也一定对应一种颜色，我们可以猜测此种颜色的大致模样。我们发现第一个分量 R 的值较小，第三个分量 B 的值也较小，而第二个分量 G 的值较大 ，所以此颜色以 G(即“绿”色)为主，但比纯绿要淡一些。

那么一共有多少种可能呢，即计算机可以表示多少种颜色呢?这是一个排列问题，由中学数学知识就可知道，共有

N =256×256×256=16,777,216

种颜色。但这并不意味着通常的屏幕显示可以直接显示这么多种颜色，标准 VGA 只有 16 种颜色，通常的高分辨率显示也不过只有 256 种颜色。也就是说绝大部分颜色是不能直接使用的，但对于通常的作图而言，16 色以及 256 色已足够了，对于专业图象处理才用到 16M 色和 64M 色。

混合出彩色的方式有两种：一种是彩色光线的混合法，用相加混色法；另一种是彩色颜料的混合，用相减混色法。相加混色规律为：

红光+绿光=黄光，

红光+蓝光=紫光(品红光)，

绿光+蓝光=青光，

红光+绿光+蓝光=白光。

进一步还可以混出各种色光。在彩色印刷、彩色胶片、彩色绘画中用的是相减混色法。相减混色法是利用颜料、染料的吸色性质来实现的，采用从白光中减去基色光的方法。相减混色规律为：

黄色+紫色=白光-蓝光-绿光=红色，

黄色+青色=白光-蓝光-红光=绿色，

紫色+青色=白光-绿光-红光=蓝色，

黄色+青色+紫色=白光-蓝光-红光-绿光=黑色。

在使用相减混色法时，三基色选的是黄、紫、青，但具体运用时还要加上一个“黑”，因为实际中“黄”加“青”加“紫”并不是纯的“黑”，于是有 CMYK 四色片，详见下节。

实际上 RGB 值可以“连续”变化，比如在 Windows 环境下的“画笔”程序，可以任意调制出 16 种颜色，每一种颜色的 RGB 值可以随便定义。调制 RGB 值的画面一般具有下表的式样，通常通过移动鼠标得到具体的 RGB 值。

三原色	变化范围(0←————————→255)	数值
红(Red)	●○○○○○○○○...○	0
绿(Green)	○...○○○○○○○○●	255
蓝(Blue)	○...○○○○○○○○●	255

3.2.3 CMYK分色片

一幅彩色图形可以含有各种颜色分量，输出时当然可以将所有的颜色都在一张片子上表现出来，如幻灯片、照相底片(反转了一下，与原片正好相反)，又如彩色纸样、彩色印成品。

但是，对于印刷环节而言，多种颜色分量是分次印刷出来的，CMYK分色片就是与一种印刷工艺相联系的制版胶片。

也就是说，通常的彩色画册，是通过分色，由原稿得到四张不同的胶片，由这四张片子再制作出四张印刷版，分四次印在同一张纸上最后才得到的。这四张胶片分别代表C、M、Y、K 四种颜色分量，C指cyan(青)，M指magenta(洋红)，Y指yellow(黄)，K指black(黑，为了与 “蓝”区别开，简写时不写作B而特意写作K)。实际上这四张胶片只分别反映不同颜色在不同位置的密度，仍然是黑白片。印刷的制版与分色的出片必须严格对应起来，才能正确还原出原图形的色彩。

许多软件都能对图形进行 CMYK 分色，如常见的 Photoshop 和 PhotoStyler。

有时为了得到特殊效果，除了 CMYK 四色片外，还要出专色片，比如印金色或银色。高精度的印刷品有时要出七色片，印刷七次。

丝网印刷有别于通常的彩色印刷，简单出 CMYK 片是不行的。

3.2.4 图形文件的格式

在计算机中文字和图形都是以文件的形式存贮的，文件存贮都遵循事先规定好的格式。写文件时要严格按照格式写，读文件时也要严格按照格式读，否则会出现混乱，无法正常读出内容。

文件格式与计算机所使用的操作系统也有一定关系，PC 机上 DOS 环境下就无法正常读取苹果机上的文件，反之亦然(不过，新版的苹果机操作系统可以兼容 PC 机文件格式，可以正常读出 PC 机上的文件)。

原则上文件存贮方式用户可以任意规定，但是这样做有个大问题，别人并不知道你是如何存贮的，打开文件时一定出错。为此长期以来形成了几种“标准”的文件格式，大家共同遵守使用规则，以这种方式存贮的文件大家可以共享。PC 机上未排版的纯文本文件(通常以*.TXT 为扩展名)，不含有其他特殊格式，用户读取十分方便。实际上它并不是没有格式，而是格式极其标准。磁盘上文件的格式与所存贮信息之间的关系，好比形式与内容的关系。同样的内容(信息)，可以以不同的形式

(格式)存贮，有的存贮方式节省磁盘空间，有的存贮方式不节省或只节省一点点磁盘空间。

使用 WPS 软件以 N 命令录入的文件是纯文本文件，以 D 命令录入的文件则有本软件特定的格式，其他软件读取这种文件时会见到一些乱码，但问题还不大，你能够知道文件的内容。但是如果用 Word 软件录入并以 Word 的 DOC 格式存贮的文件，其他软件一般无法识别。

以上说的都是存贮文字的文件，存贮计算机图形还要麻烦一些。计算机图形一般都比较大，特别是彩色图形，它们将占去大量磁盘空间。所以，对于图形文件要设法对信息进行压缩，然后再存贮。

现在较流行的图形文件格式,与苹果机有关的有MAC格式(MacPaint图形格式)和 IMG 文件格式 ;与 PC 机有关的主要 TIF 格式(TIFF 文件格式)、BMP 格式、GIF 格式、PCX 格式、JPG 格式(JPEG 文件格式)、DCS 格式、EPS 格式、PCD 格式、PCT 格式、PSD 格式、RLE 格式、SCT 格式、TGA 格式等等。

人们最常用的图形格式目前看只有四种: BMP 格式、GIF 格式、TIF 格式和 JPG 格式。其中 BMP 格式与 TIF 格式一般是未压缩的，其他几种都经过了信息压缩,JPG 格式文件还经过了有损压缩(即压缩过程中有信息损失)。现将这几种文件格式作简单介绍。

3.2.4.1 BMP 格式

Windows 下的“画笔”软件使用的图形格式主要是 BMP 格式，在 Windows 3.1 下常用 16 种颜色，在 Windows95 下常用 256 色。BMP 格式

信息以点位形式存贮，信息没有任何压缩，因而不经济，占有大量磁盘空间。由于这种图形格式比较简单，所以也经常使用。特别是在 Windows 环境下用屏幕截取方式获得图形时，常采用 BMP 格式将截下来的图形存起来。

BMP 格式的缺点是不经济，一般不能用此格式存大图形以及色彩丰富的图形。此外 Windows 的“画笔”软件读取大型高分辨率 BMP 格式文件时速度极慢(Windows 95 也一样)。

3.2.4.2 GIF 格式

这是最受欢迎、使用最普遍的一种图形格式，它的全称是 Graphics Interchange Format，即“图形交换格式”。GIF 格式是 CompuServe 公司首创的一种高效的图形格式标准，它极其复杂，也很令人感兴趣。

GIF 格式受欢迎的主要原因是：第一，信息压缩效率高；第二，与具体软件和硬件无关；第三，能有效处理 256 色图象；第四，已有相当多的用户，世界上有无数漂亮的图象以 GIF 格式存贮。也许还有另一条原因，网络上的浏览器所支持的少数几种图形格式中就包括 GIF 格式。如 Internet 浏览器 Netscape 支持的两种图形格式一是 GIF 格式，另一种是 JPG 格式。

GIF 格式有 GIF 87 和 GIF 89 两个差不多的标准，都采用 LZW 压缩算法。其信息压缩过程是极复杂的，一般用户不必了解细节。

GIF 格式的缺点是：1)不能存贮真彩色图象和灰度图象；2)标准是可扩充的，这对未来发展既是好事也是坏事，因为有可能造成混乱。

3.2.4.3 TIF 格式

这是目前图形处理行业使用最多的一种格式。虽然 TIF 格式不进行信息压缩(Photoshop 也提供一种压缩的 TIF 格式，但不通用)，但 TIF 格式文件可以存贮高精度 24 位、32 位真彩色的大型图形，便于印前处理和最后供照排机输出制版胶片。

TIF 格式的文件适用于几乎所有系统，便于在 PC 机和 Macintosh 机之间移植。TIF 格式的缺点是大量占用磁盘空间，以 TIF 格式存贮的文件不便于随身携带，也许不久的将来，大容量的小软盘流行起来能够解决这一问题。在目前情况下，如果坚持以 TIF 格式存贮大型彩色图形，最好用 LHA(或者 ARJ，PKZIP 等)软件另外压缩一下，再存入软盘，等拷贝到远程计算机中后，再展开(解压缩)成 TIF 文件。注意，对于 GIF 文件，用 LHA 压缩，意义不大，因为 GIF 已经压缩过了，很难再挤出“油水”。

3.2.4.4 JPG 格式

这是一种采用 JPEG 算法的有损压缩格式，近年来随着 VCD 的普及而迅速流行起来。JPG 格式文件的信息损失程度是可以人为调节的，因而问题并不像想象的那么严重。实际上一幅图象有些信息损失，是可以接受的。

由于 JEPG 算法可以大幅度压缩信息并以 16M 色存贮图象，JPG 格式倍受欢迎。JPG 文件便于传输，网络浏览器也支持此格式。不久就会流行起来的“数字化照相技术”也将采用 JPEG 压缩算法。近期内 VCD 不会过时，LD、VCD 和 DVD 还要共处一段时间，将来可能逐步过渡到 DVD。

§3.3 图形初始化

本节以 Turbo PASCAL 6.0 为例说说做分形图时要经常考虑的“图形接口”问题。PASCAL 提供了多种*.BGI 驱动程序，通过初始化后，即可在图形界面下调用有关“函数”、 “过程”，对屏幕、内存、文件等进行若干操作。(欲了解详情请参考《Turbo PASCAL 库函数参考指南》一类书,也可以启动 Turbo PASCAL 的“集成开发环境”(IDE),打开“帮助”(Help)选单学习、查询。)

程序如果要使用图形方式，程序开头应当在 Uses 语句中声明(Uses Graph;)。启动图形方式用“InitGraph 过程”，关闭图形方式用“CloseGraph 过程”。绘完图，一定不要忘记关闭图形方式。

在文本方式下输出字符是方便的，但在图形方式下，则必须用专门的语句向屏幕指定的位置输出文字。常用如下几个过程:

SetBkColor(颜色号码); {设置背景颜色}

SetColor(颜色号码); {设置前景图画颜色}

SetTextStyle(字体，方向，字号);{设置字模}

OutTextXY(横坐标，纵坐标，欲输出的字符串);{输出文字}

上面的第三个过程中说的字体，存放在*.CHR 文件中，此类文件与*.BGI 文件的路径应当告诉 程序，否则程序找不到，不能加载。

绘图中最重要的一个语句当属描点语句了，在 PASCAL 中需调用 PutPixel 过程: PutPixel(横坐标，纵坐标，颜色值); 坐标值在计算

机中当然都得用“整数”来表示，所以如果这些值原来不是整数，在描点之前 需要把它们变成整数(用取整函数 Round 即可)。

为提高效率，要经常使用块读、块写过程，特别是将内存的信息写到文件中去时。这两 个过程是:

```
BlockRead(FromF, buf,SizeOf(buf),NumRead);

BlockWrite(ToF,buf,NumRead, NumWritten);
```

在读写一个文件之前必须打开这个文件，所以这两个过程通常与下述语句连用:

```
Assign(f, filename); {将外部文件名赋给一个文件变量}

ReWrite(f, 1); {创建并打开文件}

BlockWrite(f, win.width, sizeof(win.width));

BlockWrite(f, win.height, sizeof(win.height));

BlockWrite(f, bps, SizeOf(bps));

BlockWrite(f, pal, SizeOf(pal));

…

size:=ImageSize(0,0,299,119);

GetMem(p,size);{p 是指针变量}

BlockRead(f,p^,size);

PutImage(0,0,p^,copyput);
```

```
FreeMem(p,size);
```

在图形方式下想在屏幕输出程序计算出来的变量的值，首先应当将变量的值转化成字符串， 然后移动指针到指定的位置，再用 OutText 输出符号串。

```
Uses graph,crt;
Var
pmin:real;{定义实数变量}
ppx:string;{定义字符串变量}
…
SetTextStyle(1,0,4);
Str(pmin,ppx);{将实数转换成字符串}
OutTextXY(10,380,′pmin=′);{在(10, 380)处输出“pmin=”字样}
MoveTo(80,380);{将指针移到(80,380)处}
OutText(ppx);{写 ppx 所代表的字符串}
```

最后两句也可以用 OutTextXY(80,380,ppx)写。

用熟了 PASCAL 的图形接口语句，再转到 C 或者 C++都相当容易，只是在 PASCAL 中可以随便用大小写，而 C 语言严格区分大小写(InitGraph 与 initgraph 不同)，不能随便用。

例如，为了程序运行的一致性，我们选用最为基础的系统环境 Turbo C 2.0，大家可以根据各种 C 语言之间的差别移植到其它 C 语言的环境中去。

首先我们将图形初始化定义、原点的设定和终止设定 3 个专用函数，以及 turtle graphics 用的 4 个外部变量和 5 个函数作成 include 专用图形库头文件 glib.h:

```
/*glib.h 专用图形库 */
#include <graphics.h>
#include <conio.h>
#include <dos.h>
#include <math.h>
#include <process.h>
#include <stdio.h>
#define PI M_PI
#define END 1995.6
```

```
void ginit(void)   /* 图形初始化函数 */
{
   int driver=DETECT,mode,err;
   initgraph(&driver,&mode,"");
   if(grOk!=(err=graphresult()))
   {
     printf("Error:%s\n",grapherrormsg(err));
     exit(1);
   }
}

void gend(void)    /* 图形关闭函数 */
{
    sound(3500);
    delay(400);
    nosound();
    getch();
    cleardevice();
```

```
    closegraph();
}

void set0(int x0,int y0) /* 原点位置函数 */
{
    setviewport(x0,y0,639,399,0);
}

double LPX=0.0,LPY=0.0;   /* 当前位置的 XY 坐标*/
double ANGLE=0.0,RADIAN=0.0; /*当前角的角度和弧度*/

void setlp(double lpx,double lpy) /*当前位置设置函数*/
{
  LPX=lpx;LPY=lpy;
}

void setangle(double angle)  /* 当前角度设置函数 */
{
   ANGLE=angle;RADIAN=ANGLE*M_PI/180;
```

```
}

void turn(double angle)  /* 当前角度旋转函数 */
{
  ANGLE+=angle;
  ANGLE=ANGLE-(int)ANGLE+(int)ANGLE%360;
  RADIAN=ANGLE*M_PI/180;
}

void warp(double length)  /* 指定距离的弯曲函数 */
{
    LPX+=length*cos(RADIAN);
    LPY-=length*sin(RADIAN);
}

void move(double length)  /* 指定长度直线的描画函数  */
{
    double x,y;
    x=LPX+length*cos(RADIAN);
```

```
        y=LPY-length*sin(RADIAN);
        line(LPX,LPY,x,y);
        LPX=x;LPY=y;
    }
```

本课程所有的程序使用了专用图形库 glib.h 定义的外部变量和函数、系统中的 BGI 函数(即 Turbo C 的 graphics.h 中定义的函数)和数学函数(在 math.h 中定义)等编制完成。

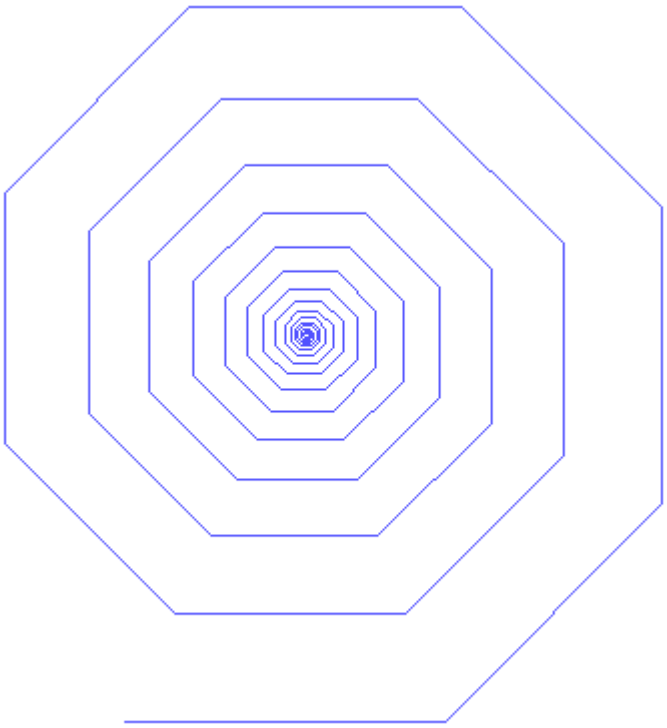
§3.4 函数递归分形图形

函数的递归是生成分形图形的最常用手法，所以，我们一开始就来熟悉这个方法。一般地，由递归所描画的图形都具有某种形式的自相似性质，但不一定是严格意义下的自相似图形，我们一般称由递归所生成的图形为“递归图形”，而“分形图形”一般指那种具有严格的自相似性质的图形。

下面我们先举两个递归图形的例子，一个是涡旋曲线，不具有自相似性质，另一个就是 Koch 曲线，具有自相似性质，是分形。

3.4.1 涡旋曲线

设现在位置为A，角度为 0 度，以直线长度 160 个象素代入递归函数 spiral中，描画直线并旋转 45 度，缩小直线长度为原来的scale倍，重复上述过程，直到直线长度小于 1 个象素时递归结束. 源程序



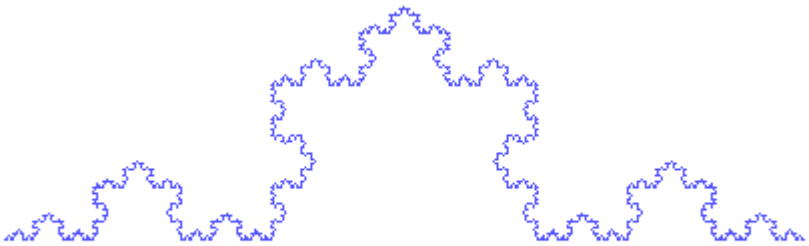
涡旋曲线

值得注意的是，在使用递归的程序中必须具有从某处退出的条件，否则会陷入死循环，永久地计算和描画下去。至于退出的方法，常用的有两种，一种就是前面用的选择一个表示长度的参数，当其值比某边界值小时则退出。另一种常用的方法是对递归次数进行计数，当达到某值时则退出。

3.4.2: Koch曲线

从当前点A开始，角度为 0，直线长度为 400 个象素点，Koch曲线的递归过程是，每次对每条边，将线段三等分，描画完第一个三分之一段后旋转 60 度，接着描画三分之一长度，再旋转-120 度，描画三分之一长度，再旋转 60 度，描画最后三分之一线段长度。我们假定递归的次

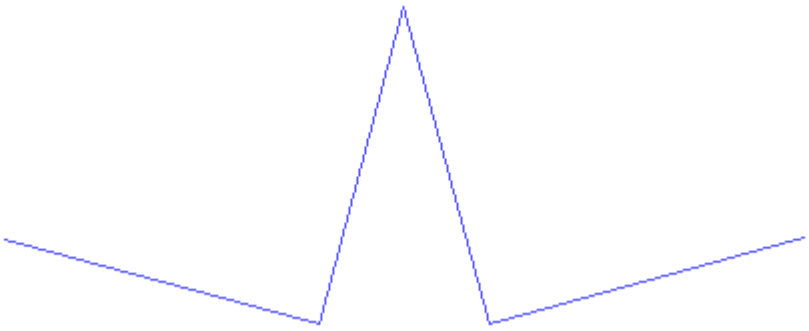
数又N决定.源程序



§3.5 生成元分形图形

3.5.1 生成元每段线段长度相同

由递归产生的分形图形中有很大一部分是由生成元，或说发生器自由地描画出来的，这时，如果生成元的每段线段的长度相同，我们可以用一个数组 gene 来指定生成元，例如数组 gene[ ]={-15.0,90.0,-150.0,90.0,END}对应的生成元如下图所示，数组的每个值表示了旋转角的大小，而从一次旋转到下次旋转之间的距离是一定的，符号常量 END 由图形库文件 glib.h 定义，它表示终止指定生成元(实际上它可以是任意值)。



数组 gene[ ]={-15.0,90.0,-150.0,90.0,END}对应的生成元

这儿要加以说明的是，数组 gene[ ]的第一个值可以取除 END 以外的任何值，其意义只是生成元第一段线段如何画，但是，在计算机屏幕上描画生成元和分形图形时，为了使开始点和终止点的 y 坐标具有相同的值，第一个值要取合适的值，上面生成元对应数组的第一个角度取 -15.0 就能做到这点。

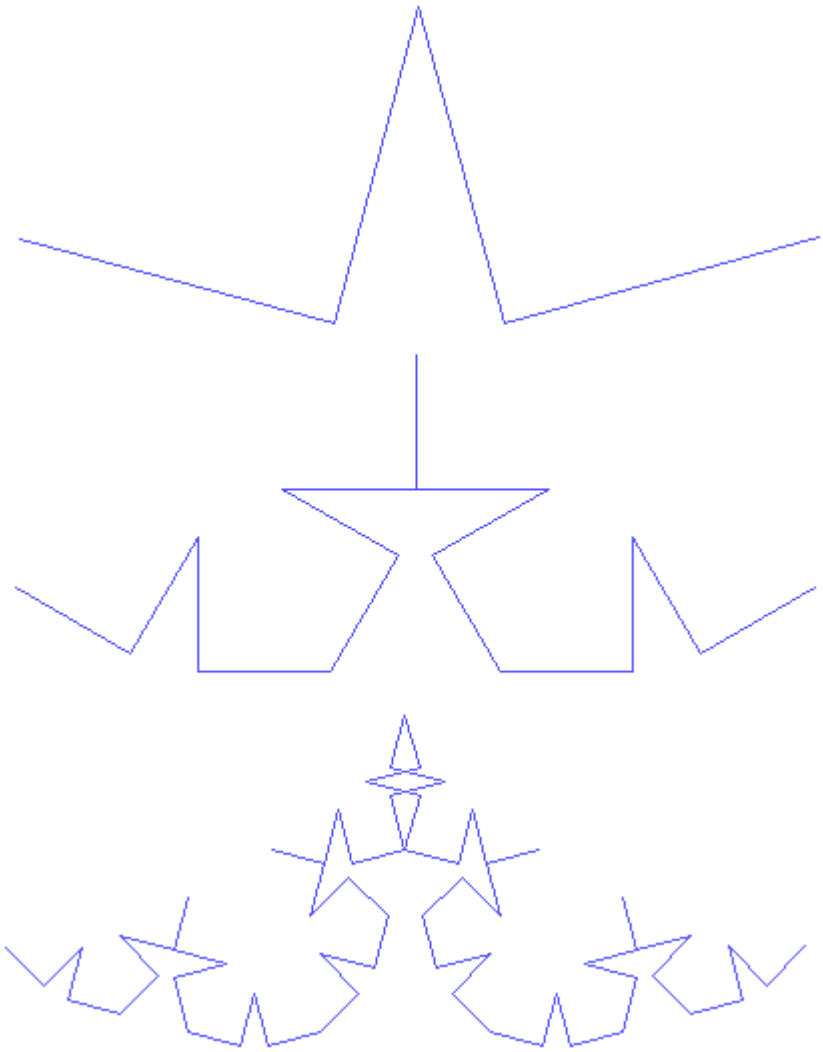
为了为生成元产生的分形图形做一个通用程序，我们还要用到几个外部变量:

N----递归次数，也是所画分形图形的次数，对递归进行计数的参数 n 达到该值时终止递归，n 为递归函数 fractal()的一个参数;

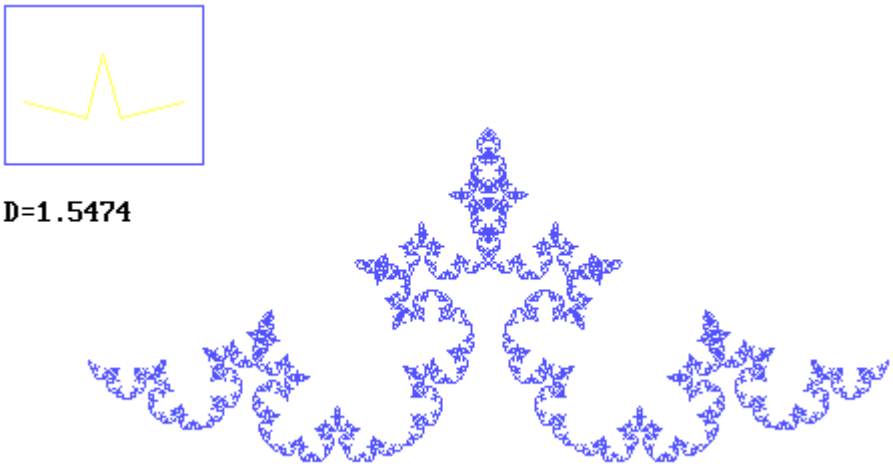
(sx,sy)----为开始描画点的 X-Y 坐标，终止点坐标为(sx+leng0,sy)，即开始点向右移动 leng0 个象素点。一旦确定了 sx,sy,leng0，则生成元无任什么形状，其开始描画点和终止点的位置均不变，要注意的是，这时相应于生成元的形状，要有使得旋转间的移动距离能自由伸缩的结构;

a----函数 fractal 的另一个参数 leng 的缩小率的倒数，其值在主函数调用函数 fractal 前求得。

例如，如上图所示的生成元生成的 1、2、3 次分形图形如下图所示，其中 1 次分形即为生成元。



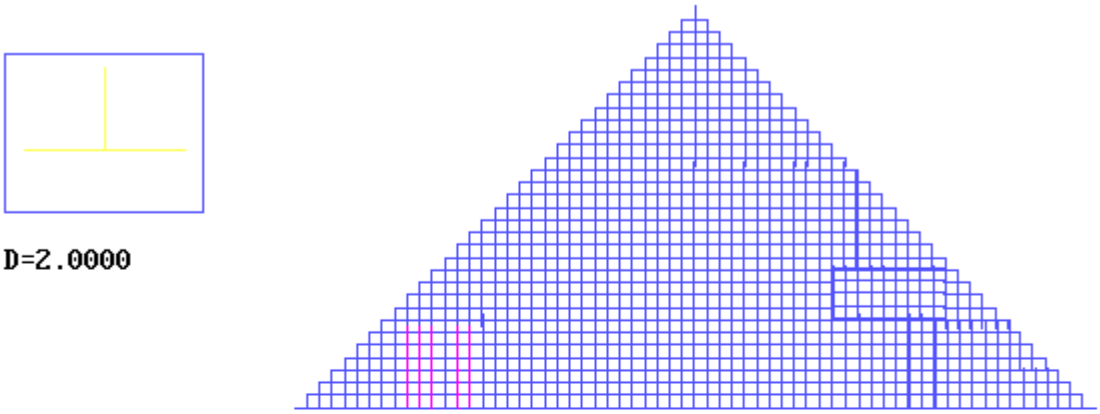
以下描画生成元生成的分形图形，我们首先在图形的左上方方框内画出生成元图形，在生成元下方显示出分形图形的维数，然后才在图形中央画出分形图形。

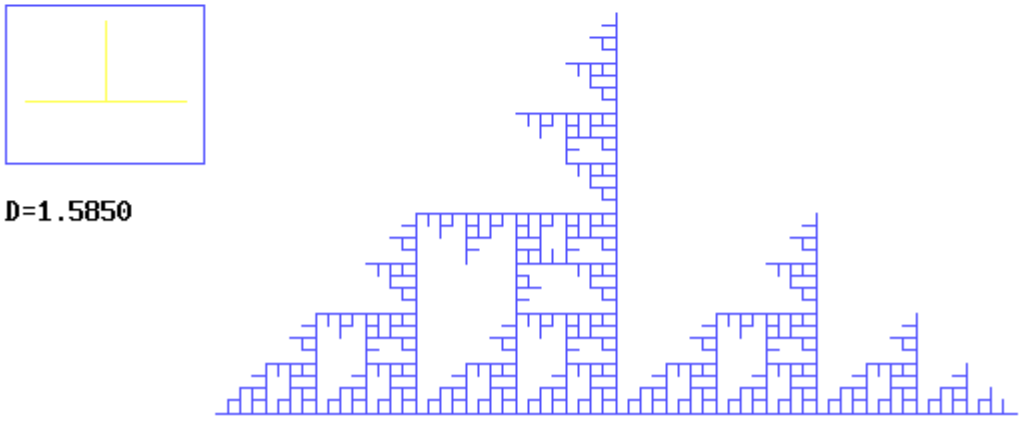


由前生成元生成的分形图形，D=1.5474

源程序

注意，当数组 gene[]的元素为 180.0 和-180.0 时，代表着返回的意思，这时是边画边返回还是停止描画跳回原来位置，需要事先加以区别，因为生成元形状对称时，一边画一边返回就会重画，从而浪费时间。一般我们规定元素值为 180.0 时边画边返回，而-180.0 时，则停止描画立即返回，其区别如下图所示。



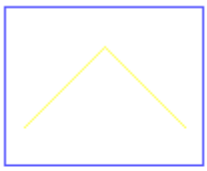


数组 gene[ ]的元素为 180.0 和-180.0 时的比较

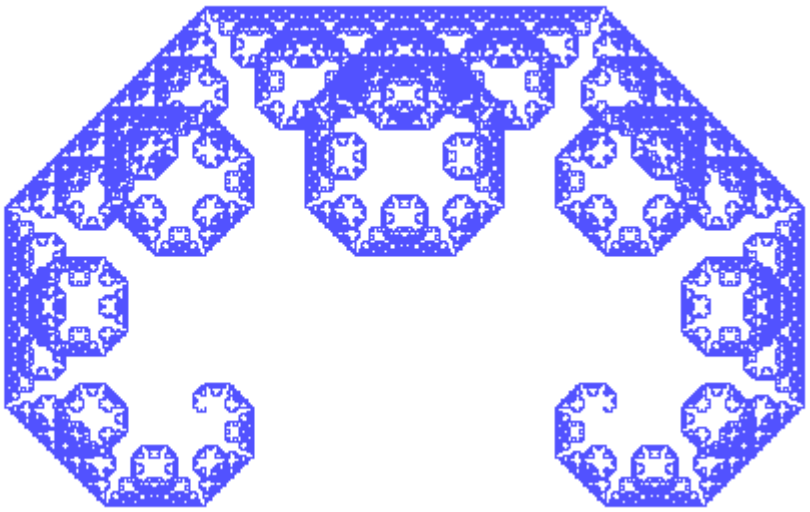
(a) gene[ ]={0.0,90.0,180.0,90.0,END}, D=2.0000;

(b) gene[ ]={0.0,90.0,-180.0,90.0,END}, D=1.5850.

生成元的的取法可以很灵活，产生的分形图形也可以千差万别，下面就是对应不同的生成元产生的分形图形。

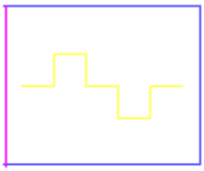


D=2.0000

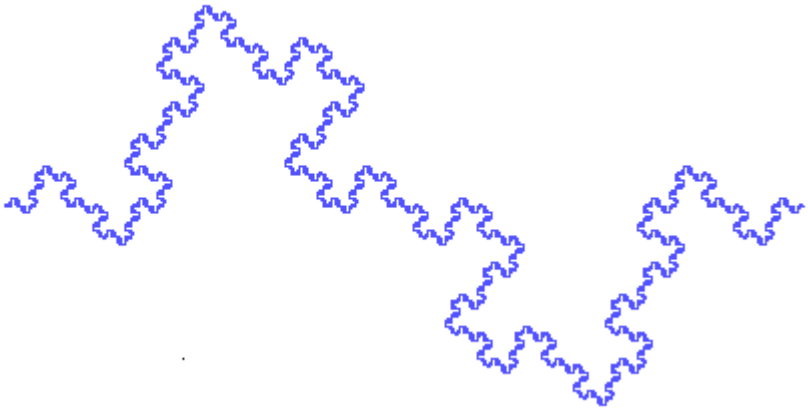


取如下数据对应的分形图形，D＝2.000

```
int N=16;                          /*递归次数*/
double sx=220.0,sy=300.0;          /*开始点坐标 */
double leng0=200.0;                /*开始点与结束点的距离  */
double gene[]={90.0,-90.0,END};
```



D=1.3652



取如下数据对应的分形图形，D＝1.3652

```
                int N=4;                        /*递归次数*/
                double sx=120.0,sy=200.0;     /*开始点坐标 */
                double leng0=400.0;           /*开始点与结束点的距
离   */
                double
gene[]={0.0,90.0,-90.0,-90.0,90.0,-90.0,90.0,90.0,-90.0,END};
```

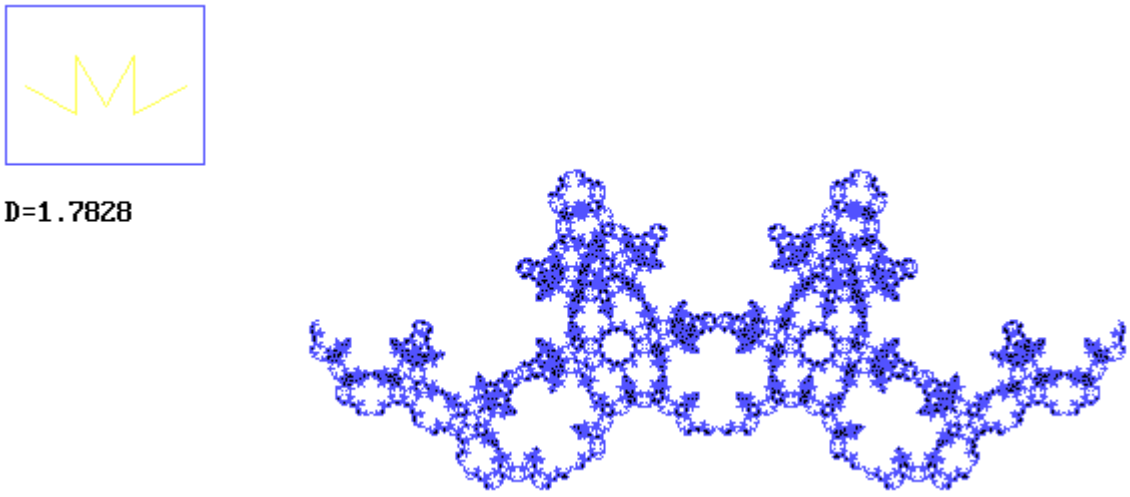


图 2.3.7　取如下数据对应的分形图形，D＝1.7828

```
                int N=5;                        /*递归次
数*/
                double sx=120.0,sy=200.0;     /*开始点坐标 */
                double leng0=400.0;           /*开始点与结束点的距
离   */
                double
```

gene[]={0.0,120.0,-150.0,120.0,-150.0,120.0,END};

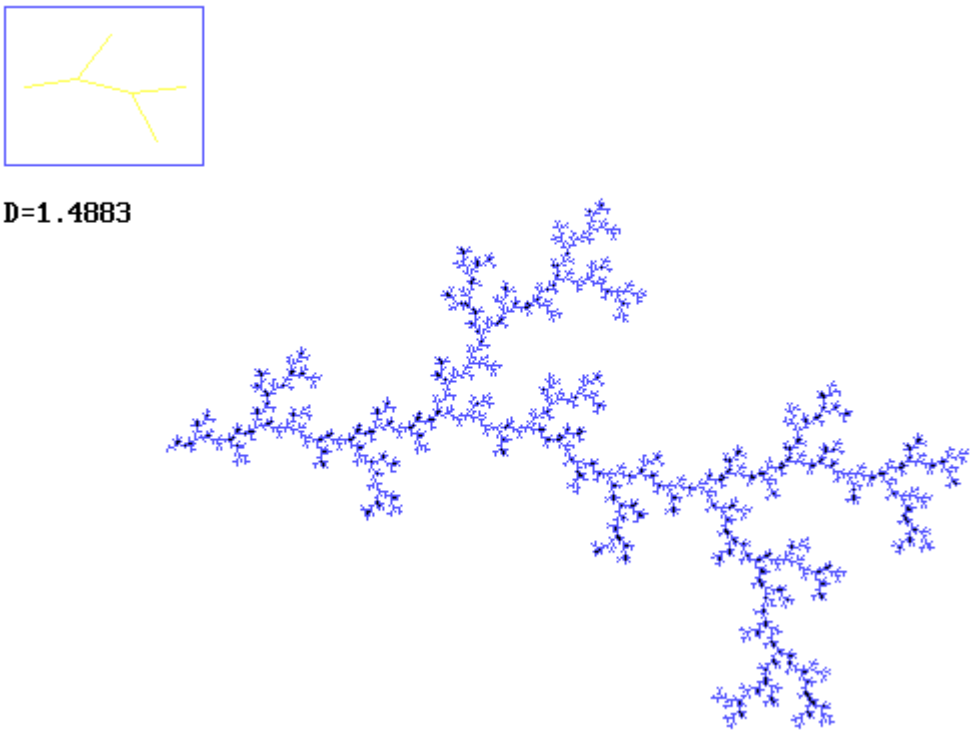
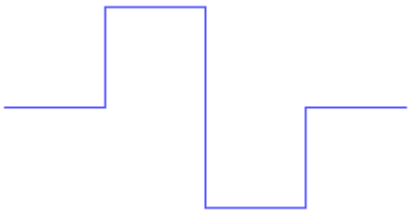


图 2.3.8　取如下数据对应的分形图形，D＝1.4883

```
int N=5;                          /*递归次数*/
double sx=120.0,sy=200.0;     /*开始点坐标 */
double leng0=400.0;           /*开始点与结束点的距离  */
double
gene[]={22.5,45.0,-180.0,112.5,-45.0,-180.0,-112.5,END};
```

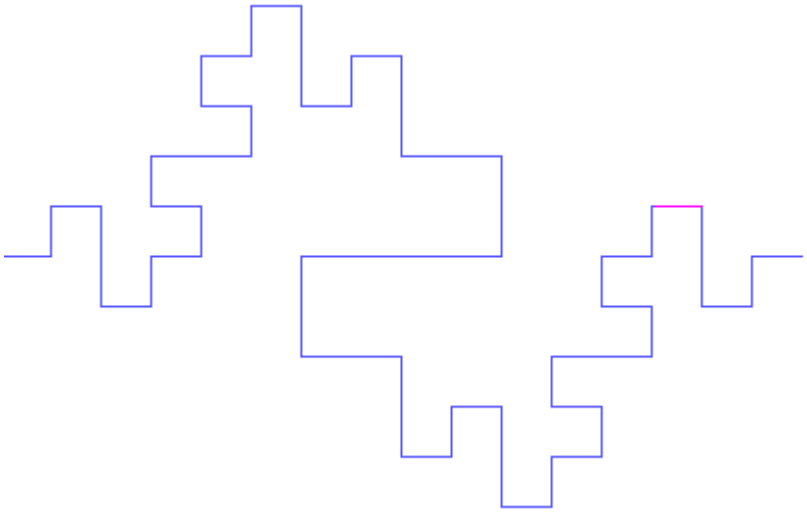
3.5.2 生成元每段线段长度不相同

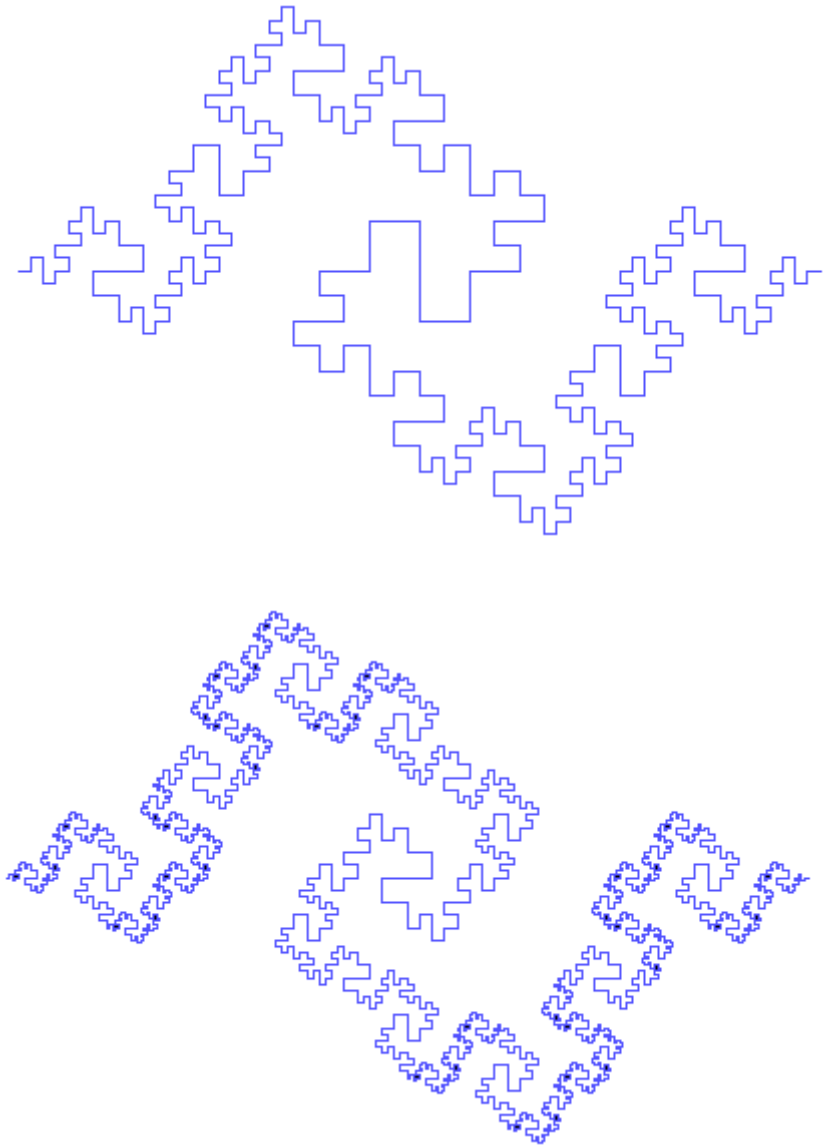
实际上，生成元的每段线段的长度可以不相同，这时，我们只需要把指定生成元的数组 gene 从一维改为二维，其中数组的第一个元素同前，指旋转角度，而第二个元素用来指定线段长度的比率，例如用数组 gene[ ][2]={{0.0,1.0},{90.0,1.0},{-90.0,1.0},{-90.0,2.0},{90.0,1.0},{90.0,1.0},{-90.0,1.0},END}指定的生成元如下图所示。



每段线段的长度不相同的生成元

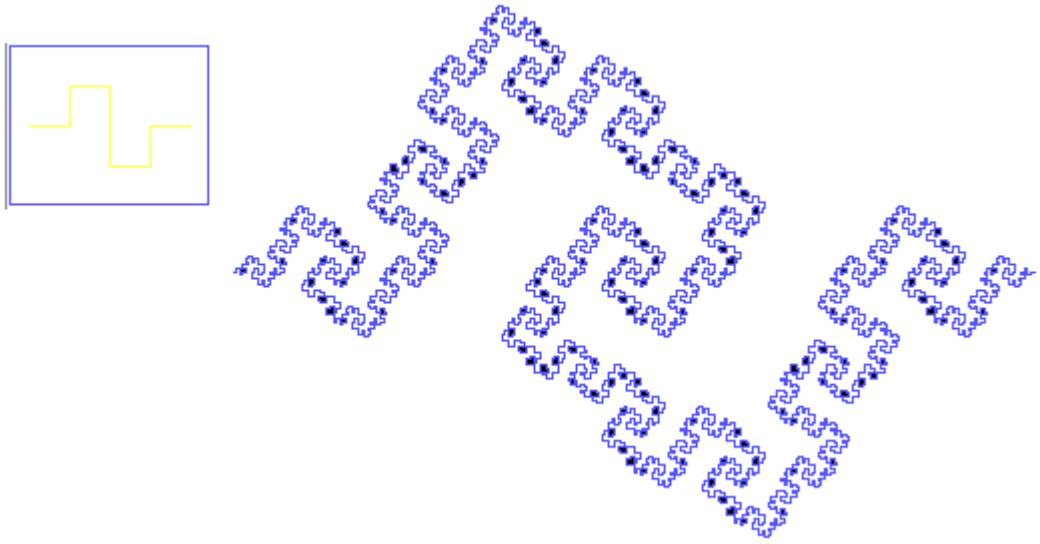
由此生成元产生的分形图形如下图，源程序。





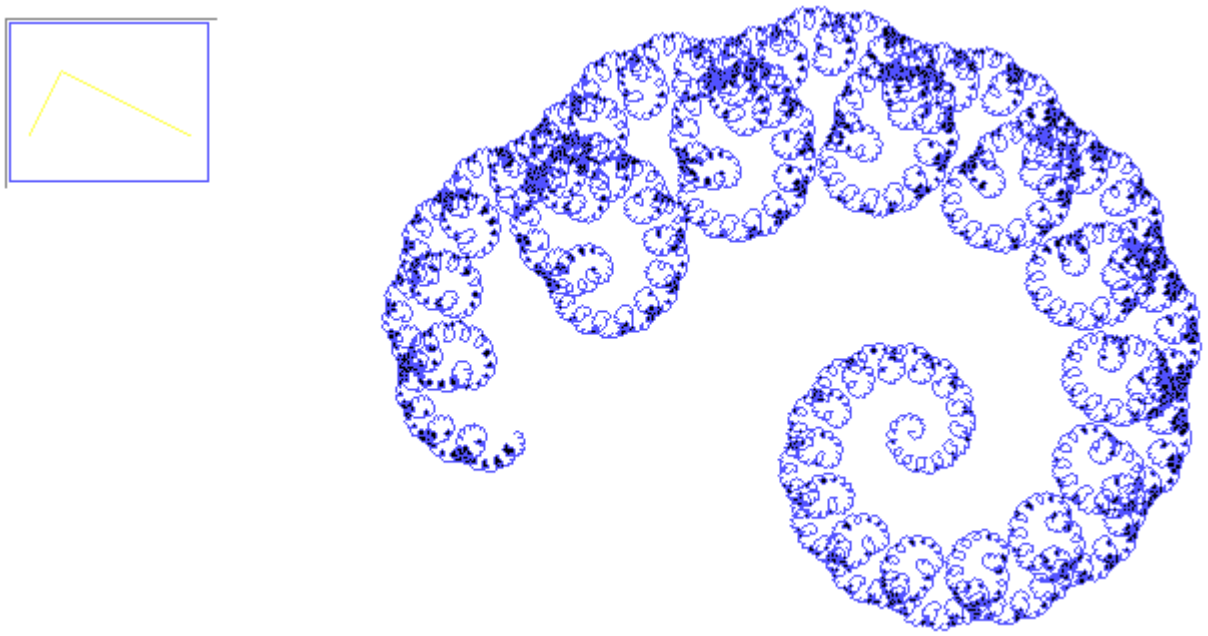
生成元产生的分形图形

看了上述程序所画的分形图形，大家很容易发现，即使大体上合格，但至少有一点不足，那就是所画的图形疏密不均，其原因就是生成元的线段长度不同而我们在程序的开头部分固定了描画分形图形的次数。为了得到疏密均匀、弯曲合适的分形图形，我们可以对终止递归的条件作修改，用leng的长度来判断是否从递归中退出，退出的理由由a赋予，leng如比a小，则执行move并终止递归。原程序.



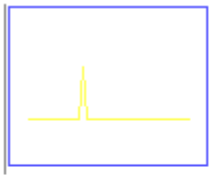
生成元产生的分形图形,终止条件由 leng 决定

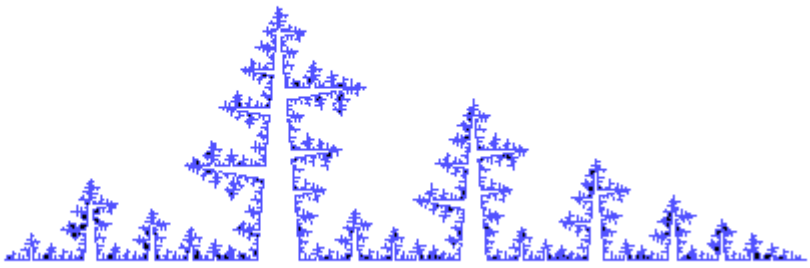
同样地，取不同的生成元可以产生差别很大的分形图形，如下列图形所示。



鹦鹉螺化石样的分形图形，其对应的数据为

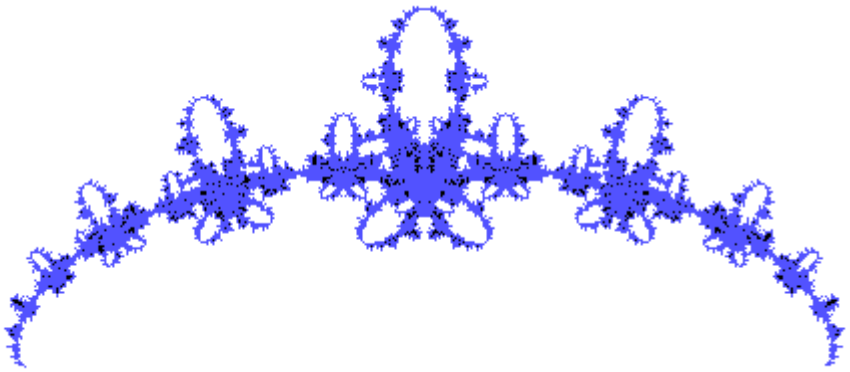
```
double sx=220.0,sy=280.0;      /*开始点坐标 */
double leng0=200.0;            /*开始点与结束点的距离
*/
double gene[][2]={{90.0,1.0},{-90.0,2.0},END};
```





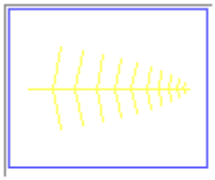
针叶树的树林分形图形，其对应的数据为

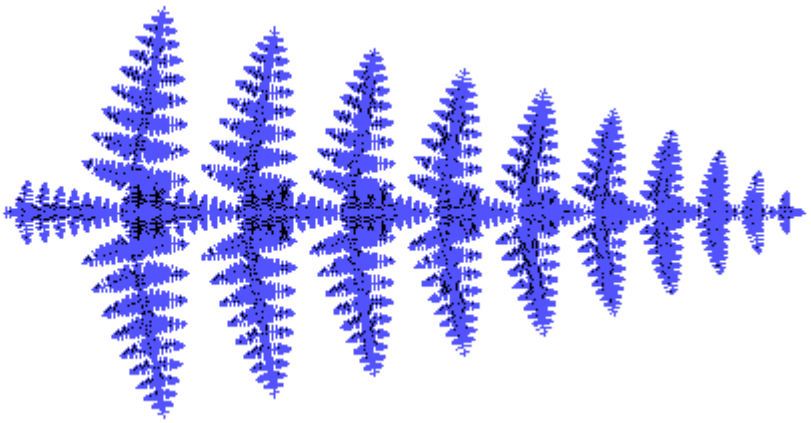
```
double sx=120.0,sy=280.0;      /*开始点坐标 */
double leng0=400.0;            /*开始点与结束点的距
离  */
double
gene[][2]={{0.0,1.0},{85.0,1.0},{-170.0,1.0},{85.0,2.0},END};
```



皇冠状分形图形，其对应的数据为

```
double  sx=120.0,sy=300.0;      /*开始点坐标 */
double leng0=400.0;             /*开始点与结束点的距离  */
double
gene[][2]={{0.0,2.0},{90.0,1.0},{150.0,1.0},{90.0,2.0},END};
```





羊凿树叶状分形图形，其对应的数据为

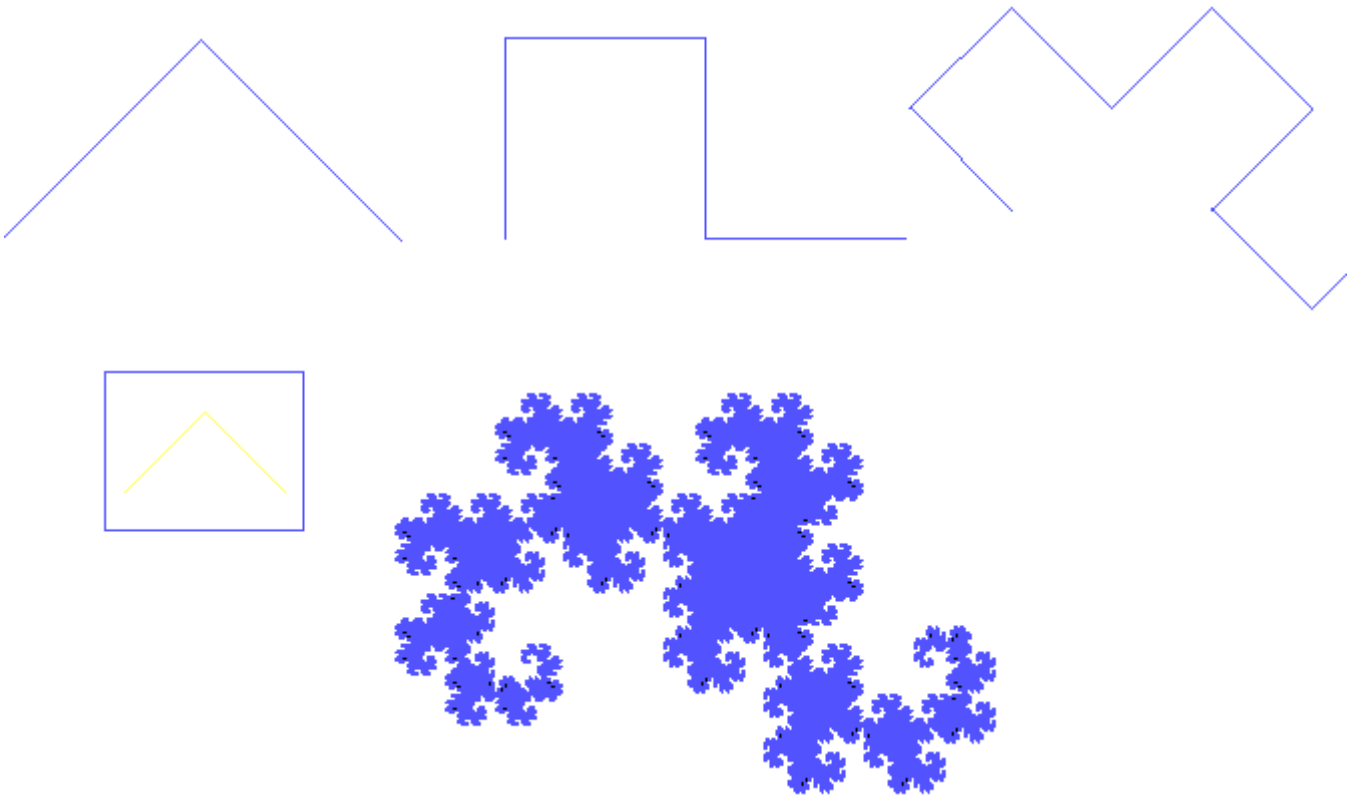
```
double  sx=120.0,sy=200.0;      /*开始点坐标 */
```

```
double leng0=400.0;                    /*开始点与结束点的
  距离  */
double  gene[][2]  =  {  {0.0,0.6},  {80.0,1.0},
  {-180.0,1.0},  {20.0,1.0},
    {-180.0,1.0},{-100.0,0.55},{80.0,0.9},{-
    180.0,0.9},{20.0,0.9}.
    {-180.0,0.9},{-100.0,0.5},{80.0,0.8},{-1
    80.0,0.8},{20.0,0.8},
    {-180.0,0.8},{-100.0,0.45},{80.0,0.7},{-
    180.0,0.7},{20.0,0.7},
    {-180.0,0.7},{-100.0,0.4},{80.0,0.6},{-1
    80.0,0.6},{20.0,0.6},
    {-180.0,0.6},{-100.0,0.35},{80.0,0.5},{-
    180.0,0.5},{20.0,0.5},        {-180.0,0.5},
    {-100.0,0.3},{80.0,0.4},{-180.0,0.4},{20
    .0,0.4},
    {-180.0,0.4},{-100.0,0.25},{80.0,0.3},{-
    180.0,0.3},{20.0,0.3},
    {-180.0,0.3},{-100.0,0.2},{80.0,0.2},{-1
    80.0,0.2},{20.0,0.2},
    {-180.0,0.2},{-100.0,0.15},{80.0,0.1},{-
    180.0,0.1},{20.0,0.1},
```

{-180.0,0.1},{-100.0,0.1},END};

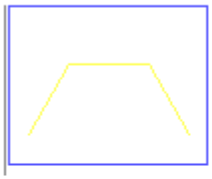
3.5.3 生成元每段线段长度与旋转方向不相同

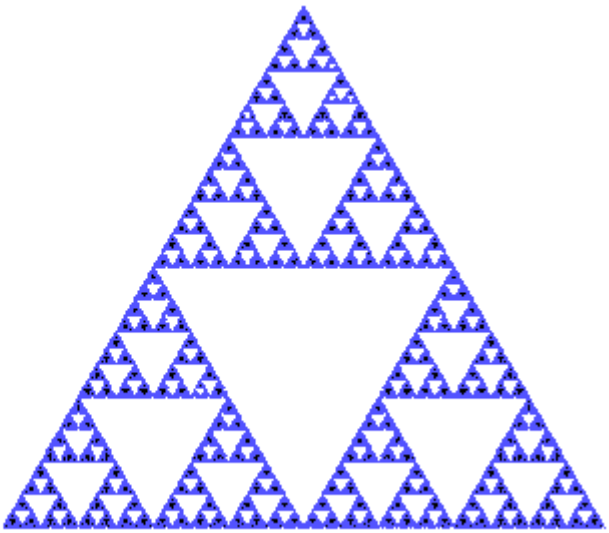
如前面所述，在嵌入生成元时，线段旋转的方向可以不同，反映在表示生成元的数组中，我们可以规定数组gene中表示各段长度的第二元素如果是正的话，正常旋转，如果是负的，则这部分反方向描画。反映在分形图形的描画上就是对应的地方被嵌入的生成元是向里折返的。例如数组gene[ ][2]={{0.0,1.0},[-90.0,-1.0],END}对应的生成元和 2 次、3 次分形图形如图所示。其对应的原程序。



生成元旋转方向不同对应的分形图形

要注意的是，生成元相同但旋转方向不同生成的分形图形是不同的，另外，当生成元各段线段长度相同时常用递归次数来终止递归，因为这比较简单，只有长度不同时才用长度终止递归。图是西来明斯基大盖帽分形图形。





西来明斯基大盖帽分形图形，对应数据为

```
int N=9;                          /*递归次数*/
double sx=170.0,sy=320.0;         /*开始点坐标 */
double leng0=300.0;               /*开始点与结束点的距离 */
double
gene[][2]={{60.0,-1.0},{-60.0,1.0},{-60.0,-1.0},END};
```

§3.6 不动点映射分形图形

我们来看生成元递归产生分形图形的递归过程，如图 3.6.1 所示。从 0 次分形图形移向 1 次分形图形时，0 次图形上几乎所有的点在 1 次

图形上都不存在了，但是可以发现几个没有移动的点（不动点），即图中的起始点 P0 和终止点 P1 及弯曲的中点 P2，而且，这些点在以后的递归中仍然存在。从 1 次递归到 2 次分形图形时，不动点增加了 2 个，即 P3 和 P4，从 2 次递归到 3 次时，又增加 4 个不动点 P5，P6，P7 和 P8。

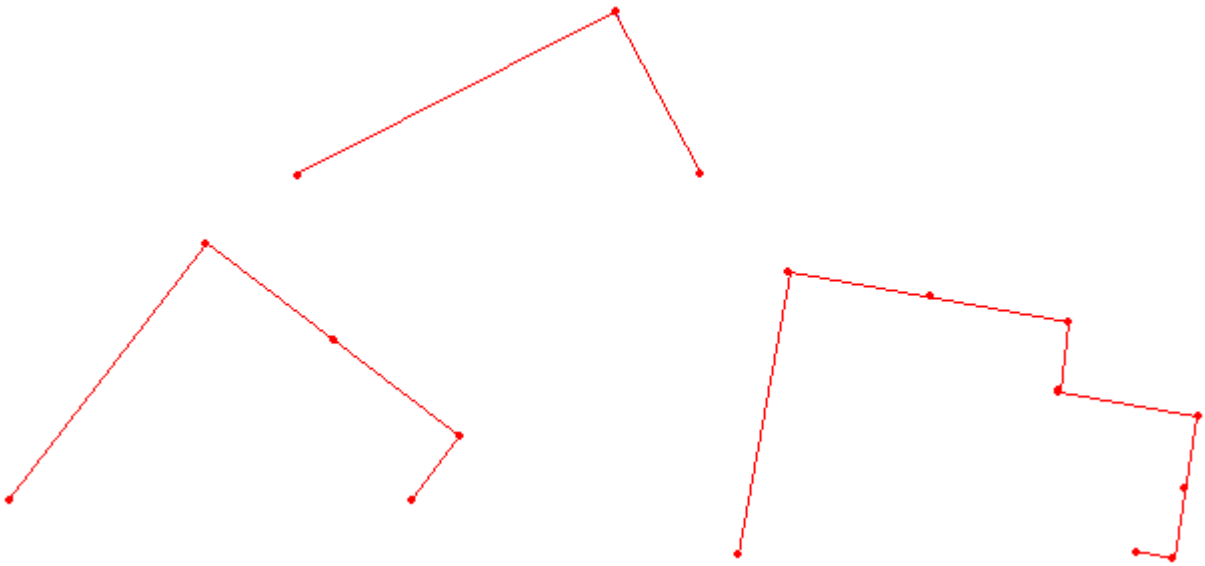


图 3.6.1 由不动点线性影射生成的 1 次/2 次和 3 次分形图形

再稍加详细地分析这些图形，在以P0 为基准时，P2 可以看成是P1 的线性映射:将P1 到P0 的距离缩小 2/sqrt(s)后围绕P0 旋转q(tgq=0.5)即可得到P2。设P0，P1 和P2 的坐标为(x0,y0),(x1,y1),(x2,y2)，则有关系式x2=x0+4(x1-x0)/5-2(y1-y0)/5，y2=y0+2(x1-x0)/5+4(y1-y0)/5。同样地，2 次递归得到的二个不动点P3 和P4 可以分别从P0 和P2，P2 和P1 得到，3 次递归得到的四个不动点P5，P6，P7 和P8 可以分别从P0 和P3，P3 和P2，P2 和P4，P4 和P1 得到...n次递归时，不动点的数目包括起始点和终止点在内一共有 2^n+1 个。这样，我们就可以从

不动点的线性映射增加不动点这种顺序构造不动点的方法来编写分形图形的生成程序。定义外部变量(sx,sy)表示描画起始点的坐标，leng0表示起始点和终止点之间的距离，即终止点位于从起始点右移leng0的位置。在主函数main中，先用set0设定原点(sx,sy)，然后用兰色描画起始点，用红色描画终止点，代入递归函数fractal中。另外，4个参数(x0,y0),(x1,y1)表示前一层相邻不动点的坐标，(px,py)表示不动点线性映射生成新的不动点的坐标，n表示递归次数，当该值大于N时递归结束，t表示tgq的值，a,b代表根据t所定的线性映射方程的系数。图3.6.2就是一个由图3.6.1所示的不动点线性映射生成的波浪状分形图形，原程序.

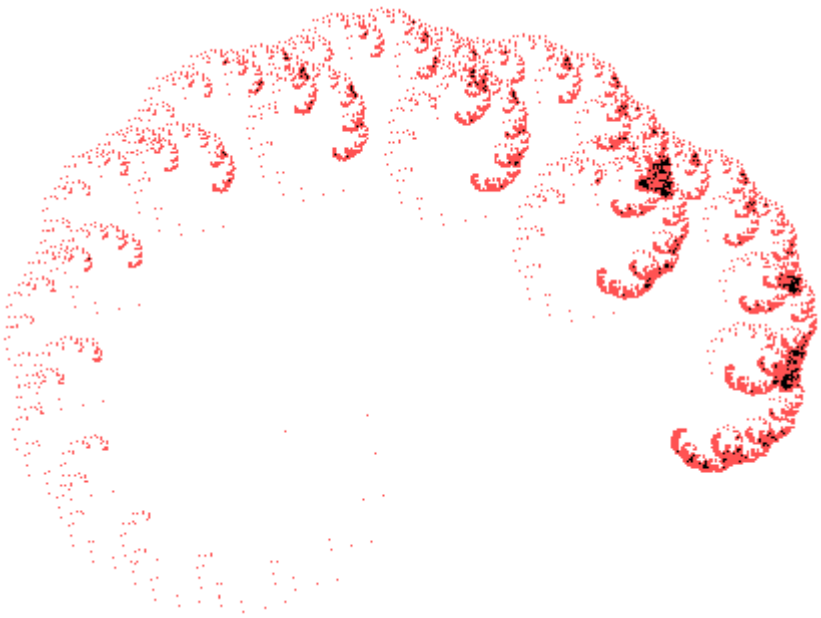


图3.6.2　由图3.6.1所示的不动点线性映射生成的波浪状分形图形

从图中可以看出，所画的分形图形不是很均匀，这主要是由于分形递归的次数被固定所故。为解决这个问题，只须根据图形固有的大小，

对递归的退出判断加以改进，即可解决这个问题，例如，可以根据两点(x0,y0),(x1,y1)间的距离平方(x1-x0)^2+(y1-y1)^2 来判断，当这个值比 2 小时就结束递归，相应地外部变量N和fractal的参数n就可以省略了。如图 3.6.3 所示的就是图 3.6.2 改进的结果，其相应的原程序.

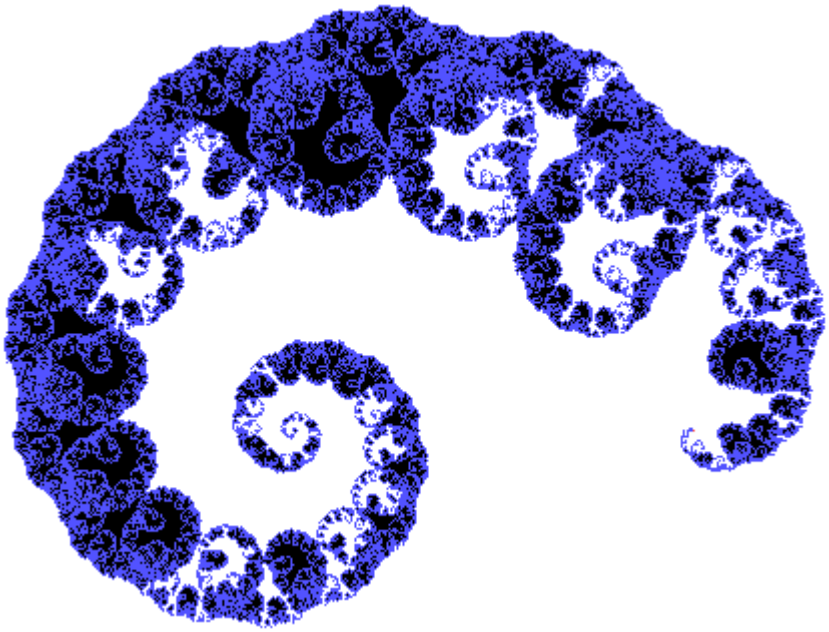


图 3.6.3　由图 3.6.1 所示的不动点线性映射生成的鹦鹉螺化石状分形图形

同样地，不动点线性映射产生新的不动点的位置也可以部分被反转，如图 3.6.4 所示，从 1 次移向 2 次时，不动点P4 就在反方向位置，3 次递归以上都这样，规定偶次编号的点所画的位置均是反方向的，其退出递归的判断条件同样可以有两种，一个对应的分形图形不均匀，一个对应的图形是均匀的：图 3.6.5 的原程序

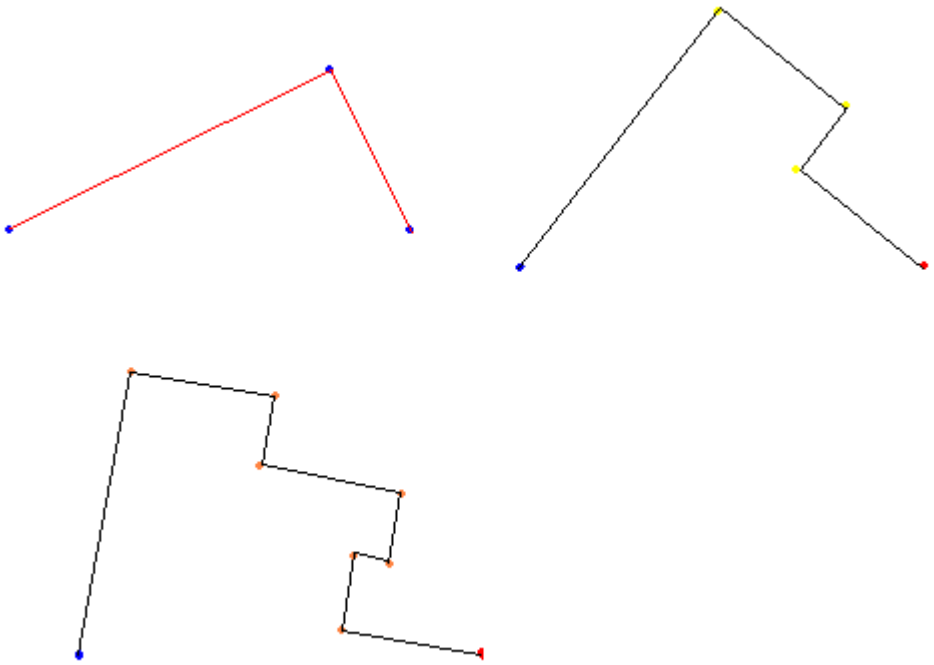


图 3.6.4　由不动点线性影射生成的 1 次/2 次和 3 次分形图形

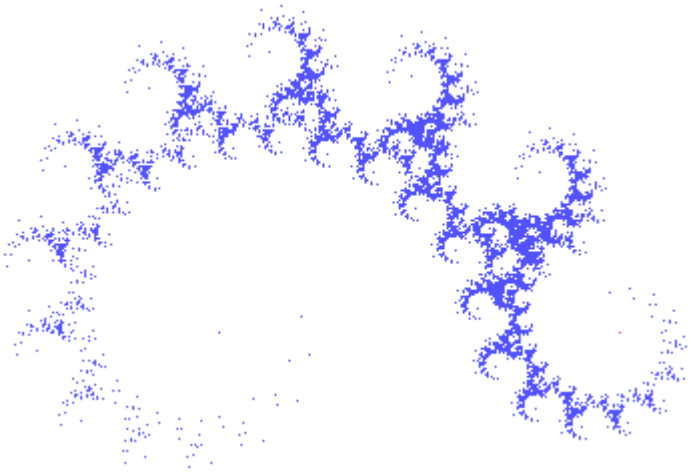


图 3.6.5　由不动点线性影射图 3.6.4 生成的不均匀分形图形

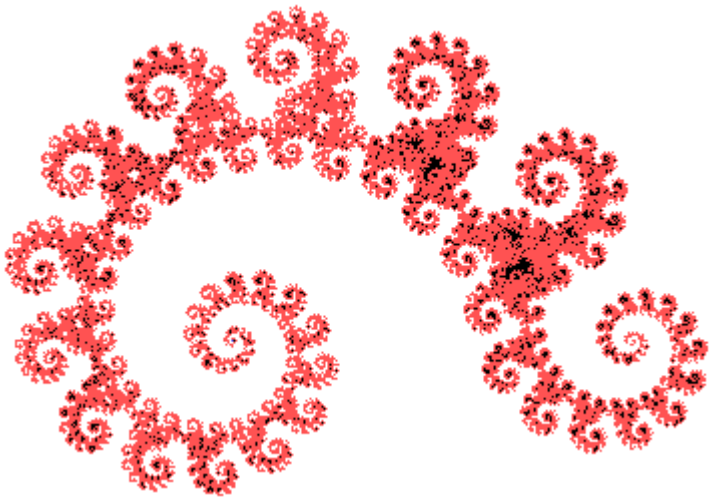


图 3.6.6　由不动点线性影射图 3.6.4 生成的均匀分形图形

图 3. 6. 6 的原程序

实际上，每次递归旋转的方向哪些是反向可以任意规定，例如可以规定每个新的不动点位置都是反向的，即偶次递归的不动点位置反向，而奇次递归的不动点位置回到正规位置上描画，应用到生成元所画的Koch雪花曲线上，生成元的 1 次递归等于不动点线性映射的 2 次递归，图 3.6.7 所示的就是用不动点线性映射产生的Koch曲线，其原程序.

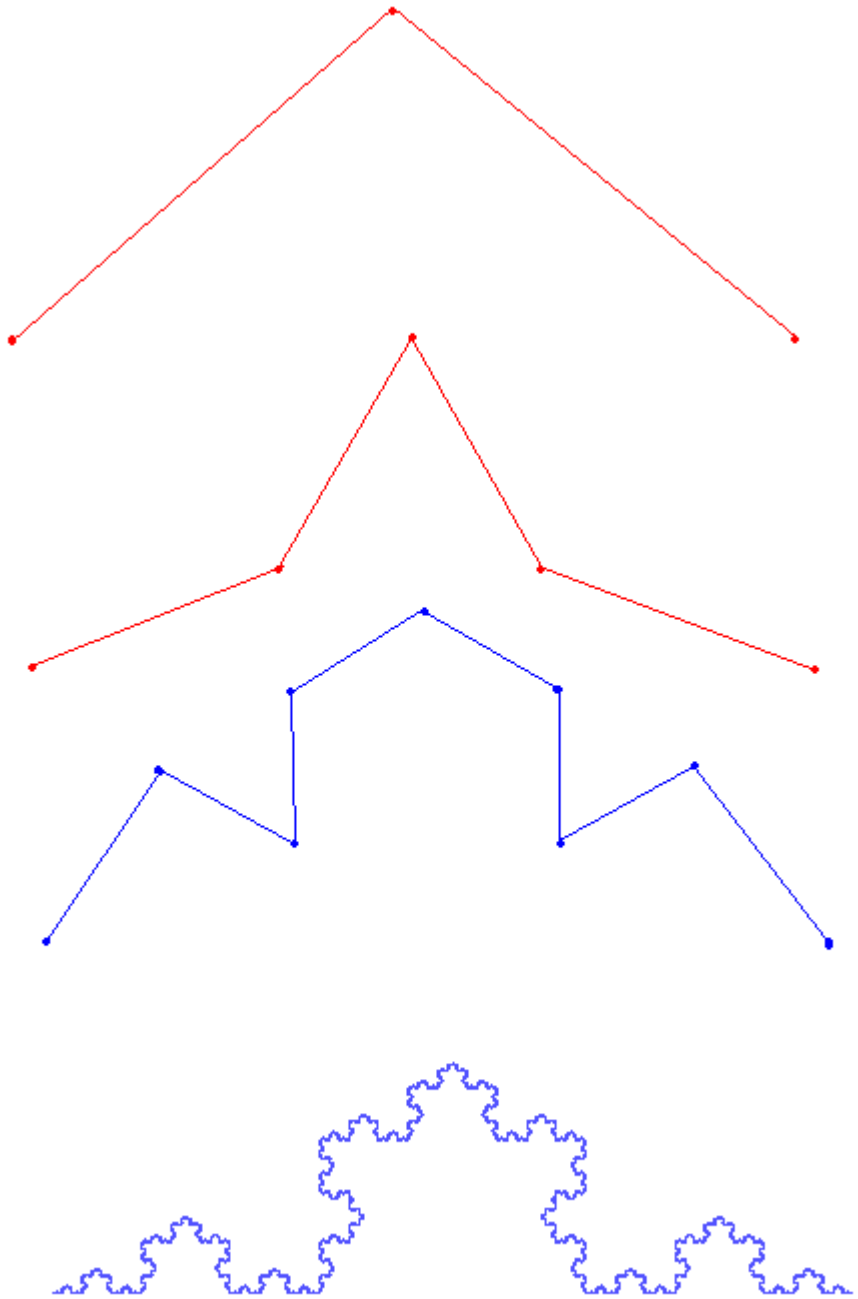


图 3.6.7　　用不动点线性映射产生的 Koch 曲线

§3.7 图案映像分形图形

本节介绍的方法有些已包含在生成元分形图形中，这时，用折线表

示的生成元被图案替代，图案上的若干点，通过递归将缩小后的图案放置于这些位置上。

基本图案可以是任意图形，为了说明问题，我们这儿先看最简单的情况：规定基本图案是围绕中心旋转的对称图形，从中心以等分 360 度的形式伸长若干分支，分支数存放在外部变量 n_br 中，在分支上成为下次描画中心的若干点等间隔地排列着，这些点的数目存放在外部变量 n_st 中，如图 3.7.1 所示的是 n_br=6，n_st=1 情况。

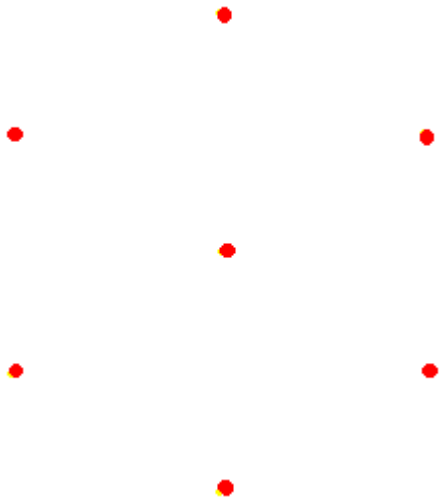


图 3.7.1　基本图案反复映像生成的分形图形

此外，还有一些外部变量：n为所画分形图形的次数，k为图案缩小率的倒数，次数每增加 1，所画图案就缩小 1/k，r0 为从中心到最远点的距离。图 3.7.1 对应的分形图形如图 3.7.2 所示，其原程序如下：

图 3.7.2　基本图案反复映像生成的分形图形 (a) n_st=1, n=5; (b) n_st=2, n=4

下面我们对基本图案增加一点复杂性，即增加二个外部变量：一是等间隔分枝上的点的位置按某种比例缩小后走向顶端，这个比例赋予外部变量scale；二是将笔直的分枝弯曲后走向顶端，弯曲的角度赋予外部变量angle，我们可以看到，增加这二个因子后，可以描画出更加复杂的图形。图 3.7.3，3.7.4，3.7.5 所示的就是几个基于基本图案反复映像生成的分形图形。图 3.7.3 对应的原程序.

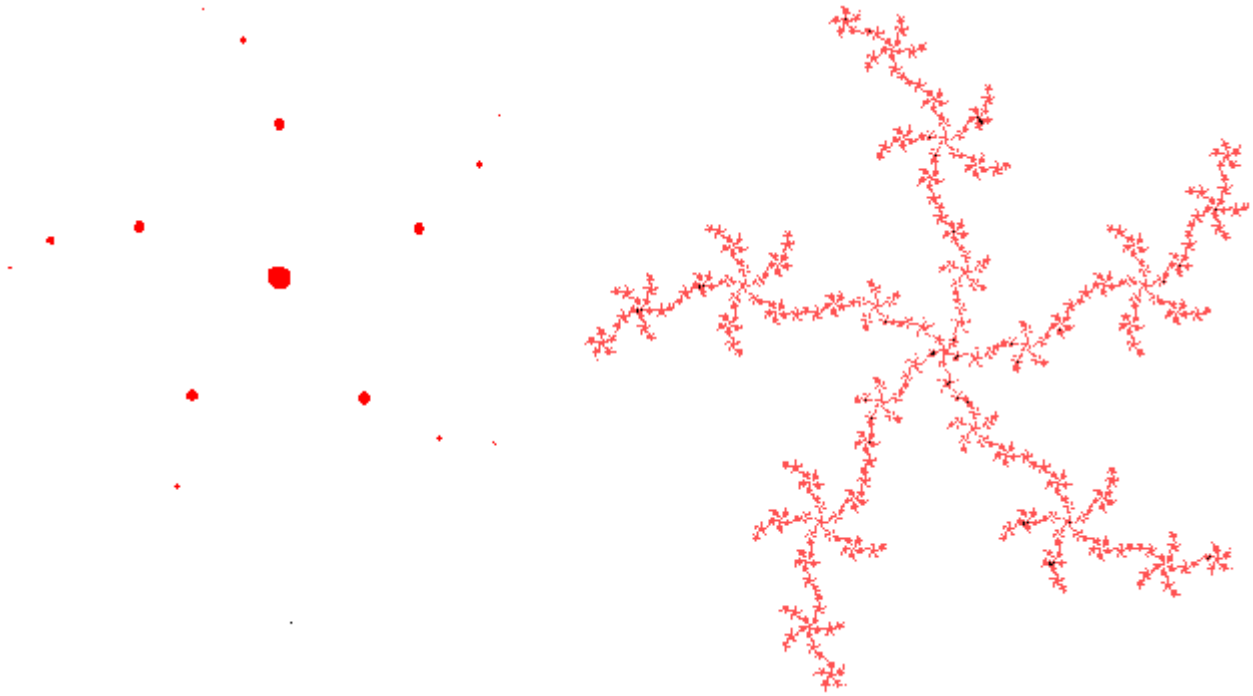


图 3.7.3　基本图案(a)反复映像生成的海星状分形图形(b)

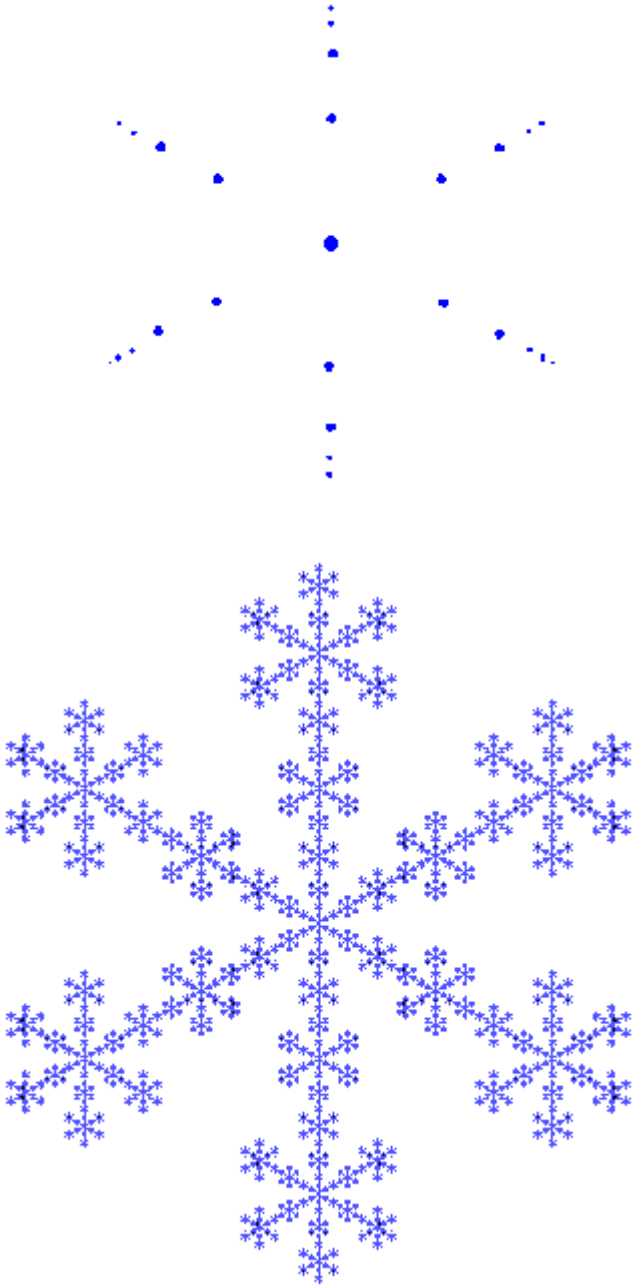


图 3.7.4　基本图案(a)反复映像生成的雪花状分形图形(b)，数据为

```
int n_br=6; /* 分枝数*/

int n_st=1 or 5; /* 分枝上小圆图数*/

double k,r0=180.0; /*r0: 外圆半径 */

double scale=0.5; /* 缩小比例 */
```

```
double angle=0.0; /* 弯曲角度 */
```

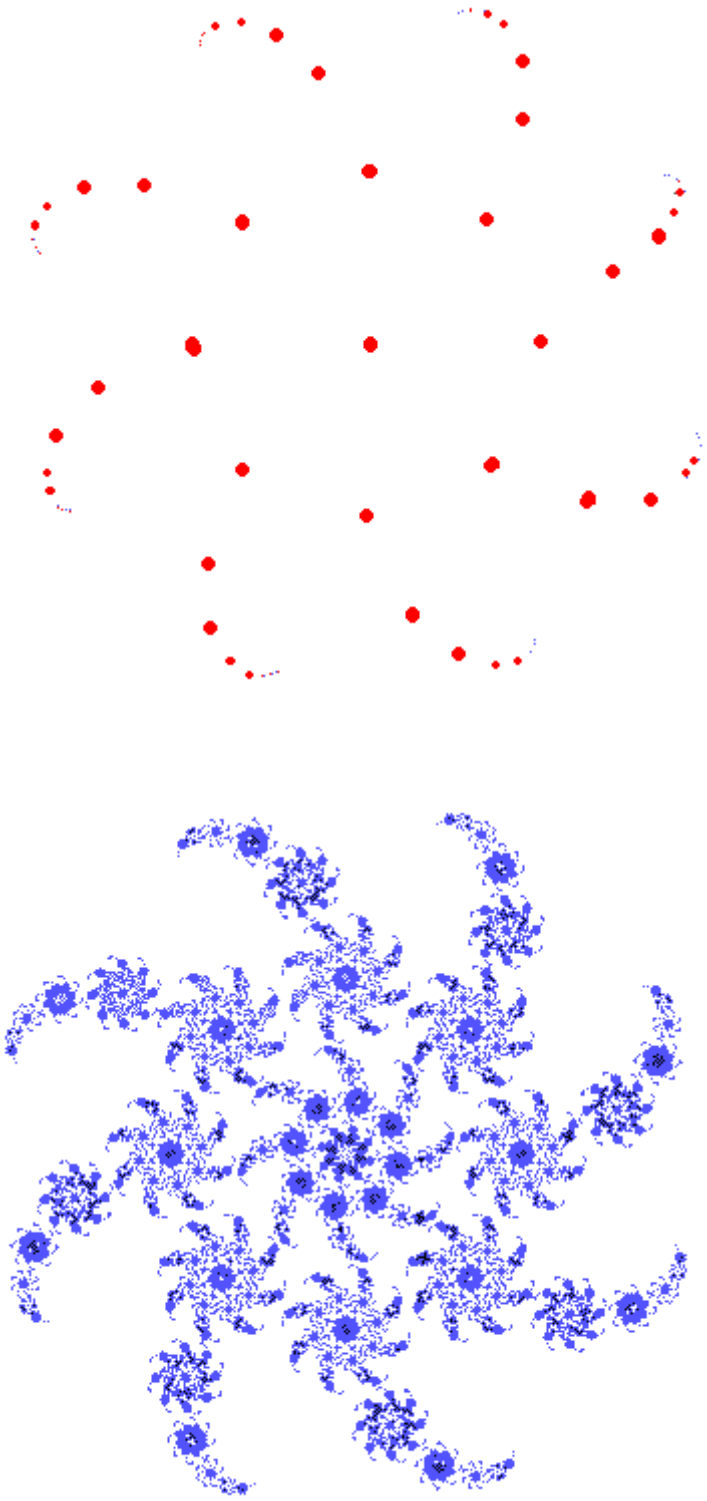


图 3.7.5　基本图案(a)反复映像生成的分形状宇宙(b)，数据为

```
int n_br=8; /* 分枝数*/

int n_st=8; /* 分枝上小圆图数*/

double k,r0=180.0; /*r0: 外圆半径 */

double scale=0.6; /* 缩小比例 */

double angle=25.0; /* 弯曲角度 */
```

§4.1 Cantor三分集

分形思想初听起来好像惊天动地，但芒德勃罗并不是神，分形并非无中生有。翻开近代数学史，已经有许多数学大师早在半个多世纪以前就构造了各种分形对象，只是他们不叫它分形，而后来数学界又很少注意他们的这些工作罢了。芒氏的贡献在于巧妙的综合，通过他的综合，历史看起来才是那样自然、那样顺理成章。我们从康托尔集合开始说起，然后介绍皮亚诺曲线、希尔伯特曲线、谢尔宾斯基三角等。

第一步，将闭区间[0,1]均分为三段，去掉中间1/3，即去掉开区间(1/3,2/3),剩下两个闭区间[0,1/3]和[2/3,1]。

第二步，将剩下的两个闭区间各自均分为三段，同样去掉中间1/3的开区间：[1/9,2/9]和[7/9,8/9]，这次剩下四段闭区间：[0,1/9]，[2/9,1/3] ，[2/3,7/9]和[8/9,1]。

第三步，重复上述操作，删除每一小闭区间中间的1/3。

不断重复上述操作,一直到第 N 步。

无限操作下去，我们看最后剩下了什么。在实变函数论这门课程中，把上述操作最后剩下的点组成的集合称作康托尔集合(Cantor set)。此集合在数学史上有重要作用，如今在分形理论中又再次辉煌，混沌理论和分形几何学处处碰到康托尔集合。

康托尔集合的性质是很有意思的。首先康托尔集合是自相似的，整体与部分十分相像。其次，它不包含任何区间，这一点容易想象出来，不断去掉中间 1/3，最后剩下的点不能构成区间。但康托尔集合是完备的闭集合。一般读者理解起来，可能稍有困难，这里略作解释。

如果在 M 的邻域 $N(M, \delta)$ 内有无穷多个点属于集合 E，则 M 是 E 的一个聚点。E 的全部聚点作成的集合叫做 E 的导集，记作 E'。 $E+E'$ 称作 E 的闭包。闭集合的含义是 E 包含 E'，即一个集合包含了它所有的聚点。

若 $E=E'$，即 E 是闭的且不含孤立点，则 E 就是完备的。完备集合的意思是说，集合 E 是闭的且每一个点都是聚点(即没有孤立点)。应当注意的是，“闭”、“聚 ”、“孤立”等用语与日常语言含义不同。

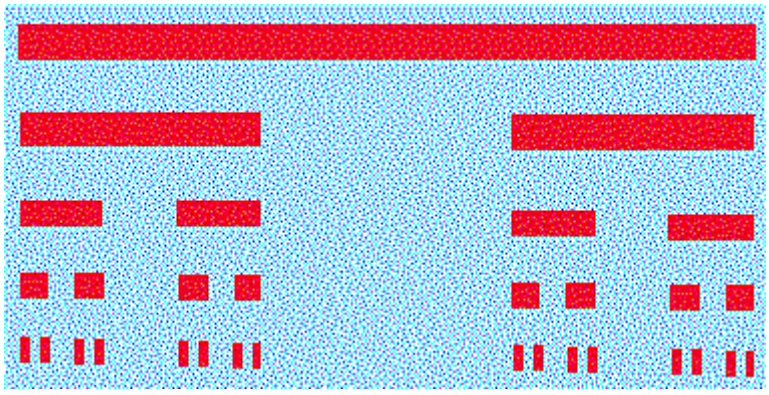


图 4.1 康托尔三分集的生成过程。每次去掉线段中间的 1/3，最后剩下的东西就是康托尔集，此图中只表示了前三个阶段。为了显示方便，无宽度的［0，1］线段在这里故意用一矩形框表示。

用 D 表示康托尔集合，可以证明 D' 包含于 D 和 D 包含 D'。前者说的是，D 是闭集；后者说的是，D 没有孤立点。康托尔集合也叫康托尔完备集。另外康托尔集合显然是非空的，所以它是“非空完备集”。

康托尔集合是可数的，还是不可数的?事实上康托尔集合 D 具有连续统的势，它可以与 $(0,1)$ 中的点一一对应起来，它是不可数的。由实变函数理论知道，任何非空完备集合都是不可数的。

我们已经习惯于 10 进制和 2 进制，现在尝试一下 3 进制。最一般地可以讨论“p 进制”（p 是大于 1 的正整数）。这些进制都是表达“数字”的方法，原则上是等价的，只是对于某类问题，某种进制显得方便一些，比如在日常生活中我们熟悉 10 进制，在计算机领域习惯用 2 进制，在计时方面甚至习惯于 12 进制、60 进制。对于康托尔集合 D，用 3 进制表示数比较方便。

为了建立集合与集合之间的对应关系，更好地说明分形集的结构，引入 p 进位小数的概念是非常有用的。设 a 为一正数，将它表示成如下收敛级数的形式：

$$a=E+n\_1/p+n\_2/p^2+n\_3/p^3+n\_4/p^4+\cdots$$

其中 E 是非负整数，p 是大于 1 的整数，n_i (i=1, 2, …) 是 0 到 p-1 的非负整数，则数 a 分解为以 p 为基底的小数，简称“p 进 位小数”或者“p 进小数”。按照通常的书写方法 a 可以写作

$$a=E.n\_1n\_2n\_3n\_4\cdots=(E.n\_1n\_2n\_3n\_4\cdots)\_p$$

特别地，(0, 1) 区间的任意实数 a 都可以写作 $a=0.n\_1n\_2n\_3n\_4\cdots$ 的形式。通常人们用到的 10 进位小数，只是 p 进位小数的一个特例。看几个实例：0.84375 用 4 进位小数表示则为 (0.312)_4，因为 3/4+1/16+2/64=27/32=0.84375；0.1525 用 20 进位小数表示则为 (0.31)_20, 因为 3/20+1/400=61/400=0.1525。

回顾康托尔集合的生成过程，开始时由 [0, 1] 中间去掉 1/3。设 $x\in$ [0, 1]，在第一步，若 x 落在头 1/3 内 (即 (0, 1/3)，则 x 的 3 进制小数展开式小数点的第一位一定是 0，不可能是 1 或者 2，否则它就落入第二个 1/3 (即 (1/3, 2/3) 或者第三个 1/3, 即 (2/3, 1) 之中了。

同样，如果 x 落在第二个 1/3 (即 (1/3, 2/3) 中，则 x 的 3 进制小数展开式小数点后的第一位一定是 1，即 x=(0.1…)_3。如果 x 落在第三个 1/3 (即 (2/3, 1) 内，则 x 的 3 进制小数展开式小数点后的第一位一定是 2，即 x = (0.2…) _3。

再深入一层，即考虑 x 小数展开式小数点的第二位。假设在上一层次 x 落在 (0, 1/3) 之内，现在仔细看，x 不但落在 (0, 1/3) 内，而且落在 (0, 1/9) 之内，则肯定 x 的小数展开式小数点后的第二位数字也是 0，即 x=(0.00…)_3。如果这次落在了 (2/9, 1 /3) 之内，则 x=(0.02…)_3。

如果在某一层次上看 x 落在了 $(8/9,1)$ 内，则 $x=(0.22\cdots)_3$。

依此类推，不断仔细看，相当于拿放大镜看，我们追踪了康托尔集的生成过程，也追踪了一个数的小数展开式中数字的含义。我们注意到，剩下的所有点，在 3 进制小数表示中一定不含有数字 1，为什么?因为凡是含有数字 1 的小数所对应的点，最终都被删除了！

反过来，可以重新定义康托尔集合 D: 康托尔集合是由 0 和 1，以及 $(0,1)$ 内那样的一些点组成的集合，这些点在 3 进制小数表示中不出现数字 1。

§4.2 Peano曲线与Hilbert曲线

什么是曲线?直观上人们会说，有长无宽的线叫曲线。但这不是定义，细分析起来这种说法甚至是矛盾的，数学家确实找到了奇特的曲线，它们能够充满平面，即这样的曲线是有面积的!皮亚诺曲线就是一个典型的例子。这种例子在数学分析、拓扑学中一直是作为反例来讲授的，教师告诉学生，不要简单化地理解曲线，于是就举出了这样一些奇奇怪怪的东西，然后再放心地说通常不会遇到这些怪物。过去就是这样做的。然而现在不同了，分形几何学兴起后，当年的怪物时兴起来，一下子由反例跃居为主角。

意大利数学家皮亚诺 1890 年构造了一种奇怪的曲线，它能够通过正方形内的所有点。此曲线的这种性质很令数学界吃惊。如果这是可能的，那么曲线与平面如何区分?于是当时数学界十分关注这件事。次年

(即 1891 年)大数学家希尔伯特也构造了一种曲线，它比皮亚诺的曲线简单，但性质是相同的。这类曲线现在统称为皮亚诺曲线，它们的特点是：1)能够填充空间；2)十分曲折，连续但不可导 ；3)具有自相似性。

图 4.2 生成希尔伯特曲线的第一步和第二步。按一定顺序相继穿过每一个小正方形的“中位线”。

希尔伯特曲线是怎样作出来的呢?我们发现，希尔伯特曲线的实质从数学上看，相当于找到了线段与整个正方形的一种连续映射，即指出线段与整个正方形可以一一对应。

用 Q 记单位正方形，[0,1]表示单位线段，现在就是想找如下形式的连续映射:

$$f:[0,1]\to Q$$

其实有许多办法，只举一个例子。第一步将正方形四等分成四个小正方形，画出小正方形的“中位曲线”(见图 4.2)。第二步将原正方形作 16 等分,按图所示次序再次画出中位曲线 。第三步将原正方形作 64 等分,同样画出中位曲线。依次类推，将原正方形 4^n 等分，画出中位曲线。当 n 趋于无穷时，正方形迷宫中的中位曲线就充满了整个正方形，成为

希尔伯特曲线。容易证明，在第一步，[0,1]被分为四段(初看好像是三段，实际上分别处于四个小正方形内，由四部分接起来的。注意，四部分的划分以处于哪个小正方形为准，不管是否是折线)，我们可以认为[0,1]被分成[0,1/4]，[1/4,1/2]，[1/2,3/4]和[3/4,1]四段，分别对应于四个小正方形。这个映射可叫 $f\_1(t)$，其中 $t\in[0,1]$。

在第二步，可以找到 $f\_2(t)$，它将[0,1]分为16段，依次对应于16个小正方形。无穷作下去，我们得到一个映射序列 $f\_1(t)$，$f\_2(t)$，$f\_3(t)$，…，$f\_n(t)$，序列的极限就是要找的连续映射 f：[0,1]→Q.

取1/3∈[0,1]，看一下它在 f 的作用下被映射到何处。我们发现1/3∈[1/4，2/4]包含于[5/16，6/16]包含于[21/64，22/64]包含于[85/256,86/256]包含于…包含于[4^n+1/3×4^n,4^n+2/3×4^n]包含于…,按照图中的虚线追踪1/3这一点在 f_1，f_2，…，f_n 作用下所对应的小正方形的位置，发现它始终被映射到正方形的左上角的小正方形中。我们按曲线行走顺序给每个小正方形先编上号(号码为1，2，…，n，不同于下面要讲的4进制编号)。

图 4.3 生成希尔伯特曲线的第四步

在第一步，它对应于编号为 2=(4^1+2)/3 的小正方形。

在第二步，它对应于编号为 6=(4^2+2)/3 的小正方形。

在第三步，它对应于编号为 22=(4^3+2)/3 的小正方形。

在第四步，它对应于编号为 86=(4^4+2)/3 的小正方形。

……

在第 *n* 步，它对应于编号为(4^*n*+2)/3 的小正方形，等等。

小正方形变得越来越小，根据压缩映象定理，最后 1/3 一定被映射为原正方形的左上角顶点。读者可以自己验证，2/3 这一点正好对应于右上角顶点。实际上对任意 $x\in[0,1]$ 都能确定它映射到 Q 中的哪一点。

用 4 进制小数，可以更好地表示皮亚诺曲线与正方形的对应关系。在 4 进制小数表示中，我们使用四个符号 0，1，2 和 3。[0,1] 之间的所有数字都可表示为“(0.…)_4”的形式， 两个特殊点 0 和 1，用 4 进制小数表示为(0.0…)_4 和(0.3…)_4。为了表示方便，省略小数点之前的 0 和小数点。比如“(0.11202)_4”简记作“11202”，“(0.00231)_4”简 记作“00231”。作了这些约定后，生成希尔伯特曲线的步骤可以表示如图 4.4。

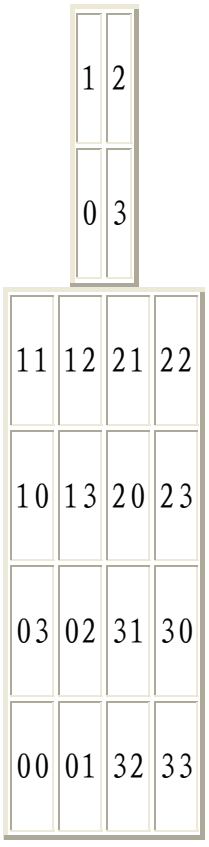


图 4.4 用 4 进制小数表示希尔伯特曲线生成过程的第一步和第二步。每个小正方形的编号顺序就是画出希尔伯特曲线的顺序。每次相继画出小正方形的“中位线”，就得到此阶段的希尔伯特曲线。

2211	2212	2221	2222
2210	2213	2220	2223
2203	2202	2231	2230
2200	2201	2232	2233

22221	22222
22220	22223

图 4.5 取上图右上角，再作两次划分(已进入第三步、第四步，但本图只表示其中的一小部分)，用 4 进制小数表示每个小正方形的编号。左图相当于 4.6 图右侧图中标号为“22”的小正方形放大两次。右图是取左图右上角一小块再作一次划分(已进入第五步)得到的。右侧图相当于左侧标号为“2222”的小正方形的放大图。

一旦使用了 4 进制小数表示，［0,1］中的点与正方形的对应关系就明朗了。我们看 1/3 所对 应的 4 进小数恰好是（0.11111…）_4，而此无限循环小数显然等于 1/3，因为

(0.1111…)_4=1/4+1/4^2+1/4^3+1/4^4+…+1/4^n=lim(n→∞)4^n-1/3×4^n=1/3

同样(0.2222…)_4 和(0.3333…)_4，分别等于 2/3 和 1:

(0.2222…)_4=2/4+2/4^2+2/4^ 3 +2/4^4+…+2[]4^n=lim(n→∞)2×（4^ n-1）/3×4^n=2/3

(0.3333…)_4=3/4+3/4^2+3/4^ 3 +3/4^4 +…+3/4^n=lim(n→∞)3×（4^n-1）/3×4^n=1

现在随便就某一层次划分中的一个小正方形，问它对应于［0,1］中什么样的数字。比如从第五步划分中选一个小方块“21023”。令 N=(0.21023)_4，则 N 一定处于(0.210230…) _4 与(0.210233…)_4 之间，于是

(0.210230…)_4≤N≤(0.210233…)_4

展开后得

(2/4+1/4^2+2/4^4+3/4^5)+0+…≤N≤(2/4+1/4^2+2/4^4+3/4^5)+3/4^6+3/4^7+…+3 /4^n+…

取极限后得

355/1024≤N≤355/1024+1/1024=356/1024

也就是说“21023”这个小正方形可表示的数字范围是[355/1024,356/1024]，这里面不但有有理数，也有无理数，并且无理数比有理数多得多。

§4.3 Koch曲线

瑞典数学家柯赫于1904年构造了如今称之为“柯赫曲线”(Koch curve)的几何对象，这一年他一共发表了两篇论文描述这种曲线，他画

出了此曲线的图形，给出了生成步骤。如果首尾闭合，这种曲线常称作柯赫“雪花曲线”(snowflake curve)，因为它酷似雪花，也很像海岸线。

柯赫曲线的生成过程很简单，以雪花曲线为例，先给出一个正三角形(作为原始形状)，然后使每一个边中间 1/3 向外折起，这一操作常称作迭代规则，于是生成了一个有 6 个角 12 个边的对象。第二步在此基础上，将每个小边中间 1/3 去掉并向外折起。以后重复此操作。经过无穷次操作就得到极限图形——柯赫曲线。

柯赫曲线是很复杂的，首先它有许多折点，到处都是“尖端”，用数学的语言讲，曲线虽然连续，但处处不可微，即没有切线。

总结一下康托尔集合、希尔伯特曲线和柯赫曲线的生成过程，我们发现它们都是从一个“原形”(initiator)开始，按照“生成元”(generator)的操作规则，不断“迭代”(iterations)得到的。实际上一大类规则分形都可以这样生成出来，这种过程具有一般性，并可以用几套语言类似地表示出来:

分形=原形+生成元+迭代;

分形=公理+产生式+解释;

分形=初条件+输入+反馈;

后面讲 L 系统时还要专门涉及上述第二种描述方式。

现在已经接触了几种传统分形对象，它们的维数用传统概念去描述都很别扭，用分数维数描述是一个好主意。对于规则的分形体，通常可

用相似性维数代表它的分数维数。设对象可 以剖分为 N个局部单元，每个单元以相似比 β 与整体相似，则对象的相似性 维数可以定义如下：

$$D=\log N/\log(1/\beta)=-\log N/\log\beta$$

在具体计算时，也可以反过来理解：如果将分形对象的一部分(用 S代表)的“线度” 放大 $1/\beta$ 倍，对象放大了 N倍(即出现了 N个 S)，则此分形的 相似性维数是 $D=-\log N/\log\beta$。 以柯赫雪花曲线为例，看其中的一 部分(以 S记之)，由生成元可知，线度放大 3 倍，对象 S放大了 4 倍(即出现了 4 个 S)，则柯赫雪花曲线的相似性维数为 log4/log3=1.2618…。

图 4.7 柯赫曲线生成过程

丢勒正五边形分形维数 $D=\log5/\log(3+SQRT(5)/2)=1.672\cdots$

康托尔集分数维数 $D=\log2/\log3=0.6309\cdots$，

希尔伯特曲线分数维数 $D=\log4/\log2=2.0$，

柯赫雪花曲线分数维数 $D=\log4/\log3=1.2618\cdots$，

柯赫岛边界线分数维数 $D=\log18/\log6=1.6131\cdots$，

柯赫十字岛分数维数 $D=\log32/\log8=1.666\cdots$，

谢尔宾斯基三角垫片分数维数 $D=\log3/\log2=1.585\cdots$，

谢尔宾斯基四方垫片分数维数 $D=\log8/\log3=1.8927\cdots$，

门格尔海绵分数维数 $D=\log20/\log3=2.7268\cdots$.

§4.4 Sierpinski地毯

波兰著名数学家谢尔宾斯基在1915-1916年期间，为实变函数理论构造了几个典型的例子，这些怪物常称作“谢氏地毯”、“谢氏三角”、“谢氏海绵”、“谢氏墓垛”。如今，几乎任何一本讲分形的书都要提到这些例子。它们不但有趣，而且有助于形象地理解分形。

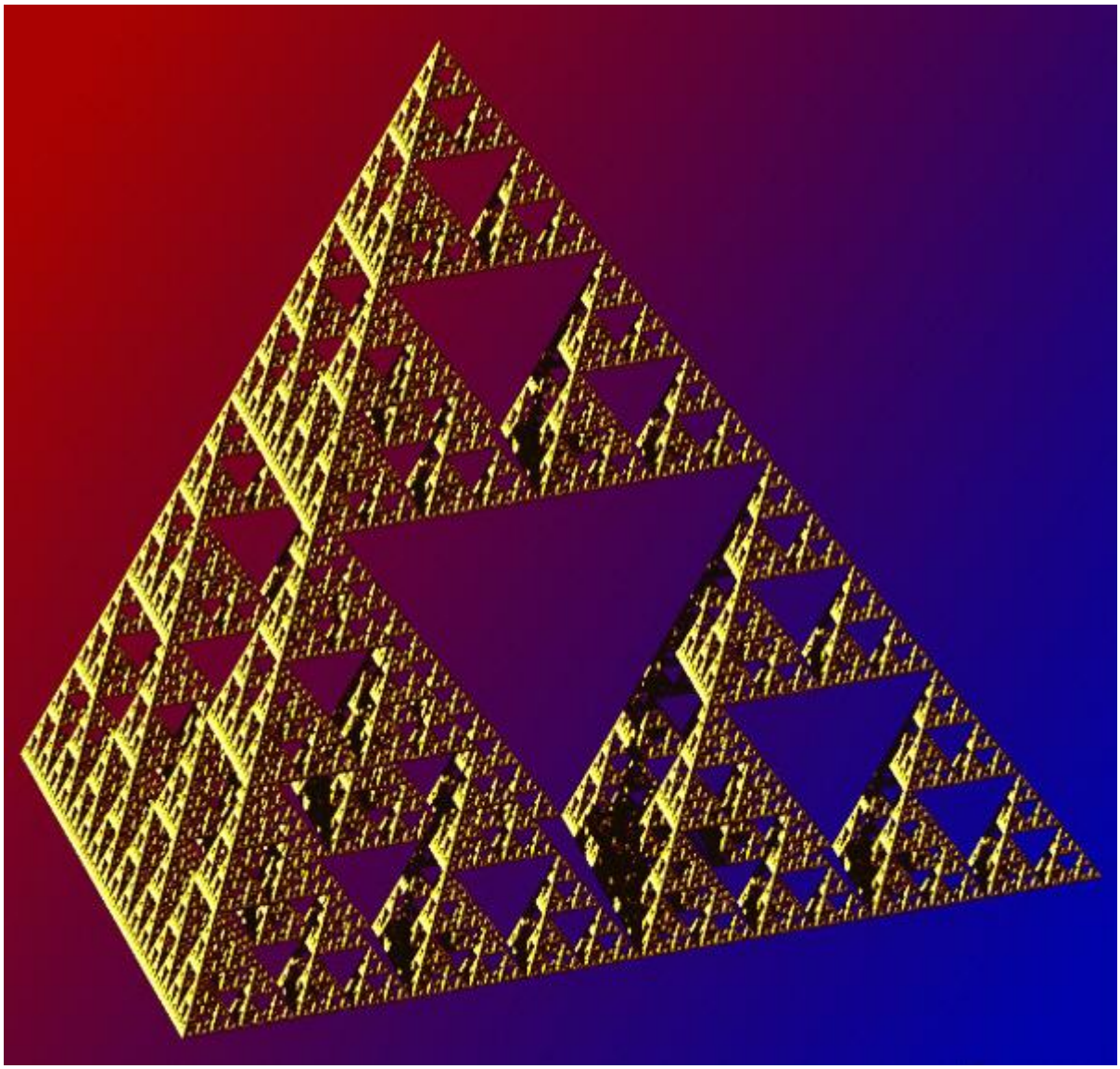


图 4.8 三维谢氏塔的自相似结构

我们首先看谢氏三角形。取一个大的正三角形，即等边三角形。连接各边的中点，得到 4 个完全相同的小正三角形，挖掉中间的一个，这是第一步。

然后将剩下的三个小正三角形按照上述办法各自取中点、各自分出 4 个小正三角形，去掉中间的一个小正三角形，这是第二步。

依次类推，不断划分出小的正三角形，同时去掉中间的一个小正三角形。这就是谢氏三角形的生成过程。数学家很关心当步数趋于无穷大时最后剩下了什么。的确，最后仍然剩下一些东西。

直观上可以想像，最后得到的极限图形面积为零。设初始三角形面积为 S，则第一步完成后去掉的面积为 $1/4S$。第二步完成后去掉的面积为 $1/4S+3\times(1/4)^2\ S$。第三步完成后总共去掉的面积为 $1/4S+3\times(1/4)^2S+3^2\times\ (1/4)^3S$。第 n 步后去掉的总面积为：

$$S\_n(\text{去掉})=S/4\times\ [\ 1+3/4+\cdots+(3/4)^{n-1}\ ]=S\times\ \ [\ 1-(3/4)^n\ ]$$

显然，当 $n\to\infty$ 时，$S\_n(\text{去掉})\to S$，即剩下的面积为零。读者朋友最好拿一张纸，亲自试一试挖取三角形的过程，挖掉的部分涂黑，用不了几步，就会发现差不多一片黑了。重要的是，在挖取操作中能够体会分形的生成过程：生成分形一点也不难，小 孩子也会做！

我们大多不是数学家，所以不必真的关心极限图形，观察前 8 步就足够了。实际上，无限精度只是个理想，作为物理学工作者(大家都是广义的物理学工作者!)，接触的永远是有限世界，更感兴趣的是有限到

无限的过程。当步数 n 比较大时，我们就可以近似认为达到了无穷。在几乎所有规则分形的生成过程中，n 取 20 便足可以认为是∞了！

在挖取三角形的过程中，我们发现，每一步骤构造出的小三角形与整个三角形是相似的，特别是当步数 n 较大时，相似性更是明显，有无穷多个相似，每一小三角形与任何其他三角形也都是相似的。

上面是以正三角形说明的，那么换成一般的三角形是否可以呢?当然可以。如果最初选一个非常一般的三角形，每次也取中点，去掉中间一个小三角形，最后得到的结论完全一样。那么不用三角形是否可以呢?也当然可以。比如开始时取一个正方形，将它 9 等分，去掉中间一个小正方形。以上都是在二维平面上操作，增加一维可以吗?当然可以。其实数学家就是这样想问题的：不断推广，力求得到更一般性、更普适的结论。

在计算机上实际生成谢氏三角形，有许多种方法。先介绍一种最难理解但又最简单的方法。

这是一个奇妙的算法，只用到了一个关键语句“IF (x and y)=0 THEN PutPixel……”。初看起来这似乎不可能，实际上这里用到了帕斯卡(B.Pascal,1623-1662)三角形二进制坐标表示的理论。要严格证明此算法的有效性，需要回顾大数学家库默尔(E.E.Kummer,1810-1893)于 1852 年的一个研究结果，这里只简要作些解释。此算法并不实际去计算谢氏三角形，而是就平面上的所有点(x,y)，判断它们的颜色，确定每一点是“白”还是“黑”。如果对某一点判断的结果是“白”，则

该点不属于帕斯卡三角形；如果结果是“黑”，则该点属于帕斯卡三角形，给它标上颜色(当然不限于黑色了)。因此需要做两个循环。

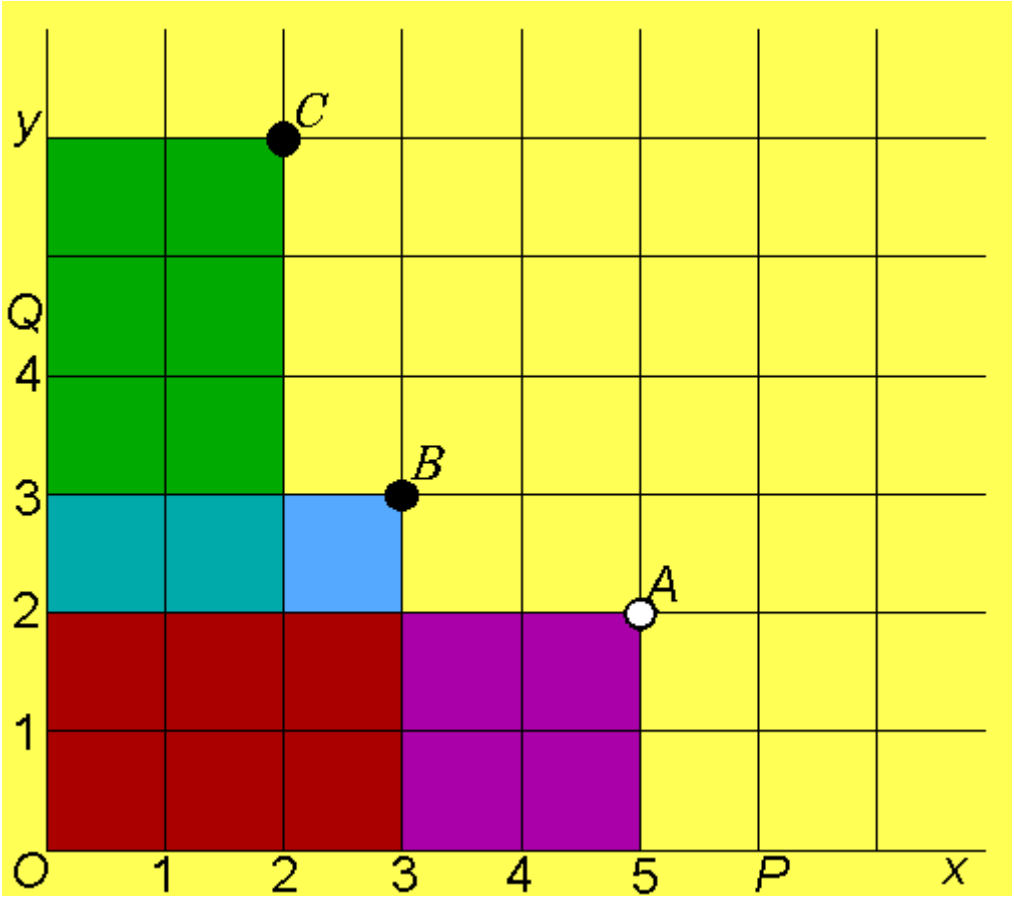


图 4.9 格点坐标及其着色，坐标轴可以不垂直

那么对于平面上的任一点 (x, y) 是如何判断其颜色的呢?先说明针对帕斯卡三角形 POQ 而定义的一种坐标系。设沿 OP 边为 x 轴，沿 OQ 轴为 y 轴，此坐标系一般说来不是直角的，x 轴与 y 轴的夹角 $\angle POQ$ 可以任意取。假设图上三个点的坐标是：$A(5，2)$，$B(3，3)$，$C(2，6)$。把它们的坐标写成二进制形式有：

$A(101，010)$，$B(011，011)$，$C(010，110)$

把各个二进制坐标上下对齐进行比较，如果某栏出现两个1，则该点标为“白”，否则该点标为“黑”。照此规则，*A*点为“白”，*B*点为“黑”，*C*点为“黑” 。

```
Program Sierpinski_Gasket_1996_11_14;
uses Graph;
var
x,y:integer;
Gd,Gm:integer;
begin
Gd:=VGA; Gm:=VGAHi;
Initgraph(Gd,Gm,'D:\\PASCAL');
if GraphResult <> grOK then Halt(1);
for y:=0 to 511  do {可以是0到255}
for x:=0 to 511 do{可以是0到255}
   begin
     If (x and (y-x))=0 then{本程序只有这一句是最关键的}
     Putpixel(x+100,500-y,2);
   end;
```

```
readln;
CloseGraph;
end.
```

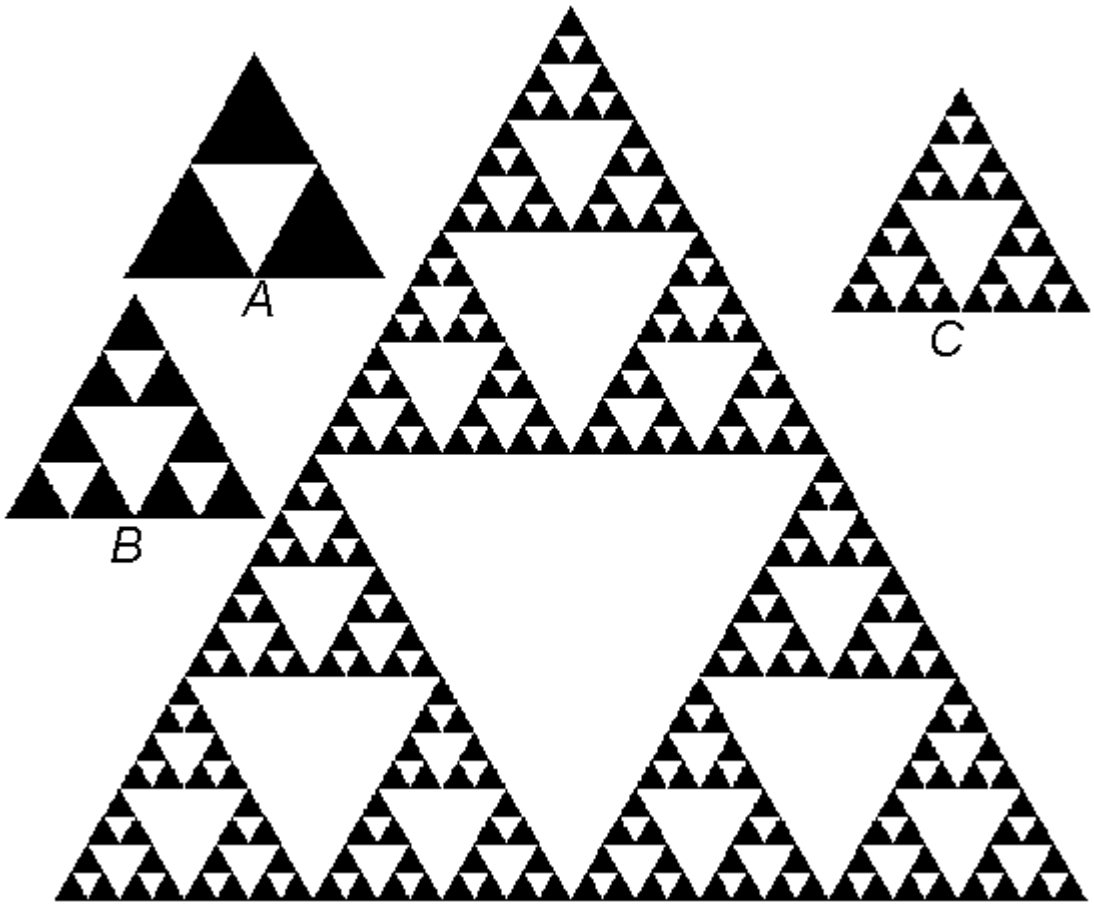


图 4.10 谢尔宾斯基三角形生成过程，只示意了前四步。

现在介绍第二种方法(也叫“中点法”)，它比上一种容易理解一些，程序也不难。本质上这种方法用到了迭代函数系统(IFS)的思想，关于 IFS 请参见后面章节。

```
Program SierpGasket1993;
uses graph,Crt;
```

```
var
p,a1,a2,b1,b2,c1,c2,x,y:real;
Gd,Gm:integer;
begin
a1:=8;   a2:=253;
b1:=253;   b2:=8;
c1:=253;   c2:=220;
x:=15;  y:=15;
Gd:=VGA;  Gm:=VGAHi;
Initgraph(Gd,Gm,'D:\\PASCAL');
if GraphResult <> grOK then Halt(1);
randomize;
repeat
p:=random(100);  {产生一个随机数 p}
if p<33 then
  begin
     x:=(x+b1)/2;  y:=(y+b2)/2;  {中点法由此得名}
  end;
```

```
if (p<66) and (p>33) then
  begin
     x:=(x+c1)/2;  y:=(y+c2)/2;
  end;
if p>66 then
  begin
     x:=(x+a1)/2;  y:=(y+a1)/2;
  end;
Putpixel(round(x*2),round(y*2),15);
until KeyPressed;
end.
```

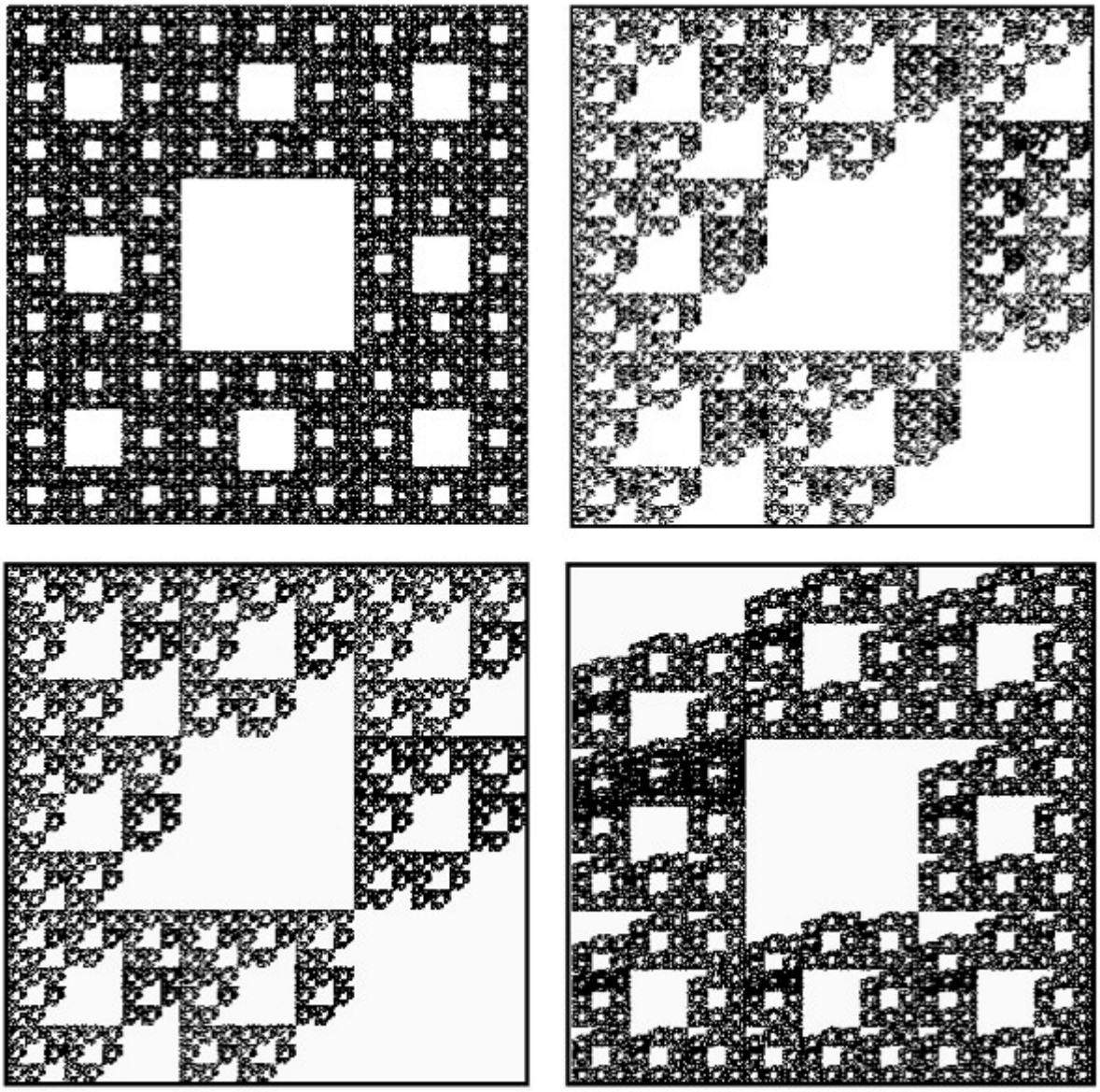


图 4.11 谢氏四方垫片

生成谢氏四方形地毯、垫片的小程序为 Carpet.PAS,您可以尝试按注释行提示的那样改变 d[1,6]、d[4,6]的赋值,这样会得到不同的花样。

```
{Carpet.PAS}
program SierCarpetFour;
uses
```

```
Graph, Crt;
var
Gd,Gm,ErrorCode,i,k,MaxY : integer;
x,y: real;
d:array[1..8,1..6] of real;
begin
GD:= Detect;InitGraph(Gd,Gm,'d:\pascal');
ErrorCode := GraphResult;
if ErrorCode <> grOk then exit;
MaxY := GetMaxY;
for i:=1 to 8 do
begin
   d[i,1]:=0.3333333333;
   d[i,2]:=0;d[i,3]:=0;
   d[i,4]:=0.3333333333;
end;
d[1,5]:=1;  d[1,6]:=1; {可试验 d[1,6]:=MaxY;}
d[2,5]:=MaxY; d[2,6]:=1;
```

```
d[3,5]:=1;   d[3,6]:=MaxY;

d[4,5]:=MaxY; d[4,6]:=MaxY div 1; {可尝试除以 2, 3, 4}

d[5,5]:=MaxY div 2; d[5,6]:=1;

d[6,5]:=MaxY; d[6,6]:=MaxY div 2;

d[7,5]:=1;   d[7,6]:=MaxY div 2;

d[8,5]:=MaxY div 2; d[8,6]:=MaxY;

MaxY := GetMaxY;

randomize;

x := 0;y := 0;

repeat

i:=i+1;

k := random(8) + 1;

x := d[k,1]*x + d[k,2]*y + d[k,5];

y := d[k,3]*x + d[k,4]*y + d[k,6];

putpixel(round(2*x/3)+50,round(2*y/3),4)



readln;

end.
```

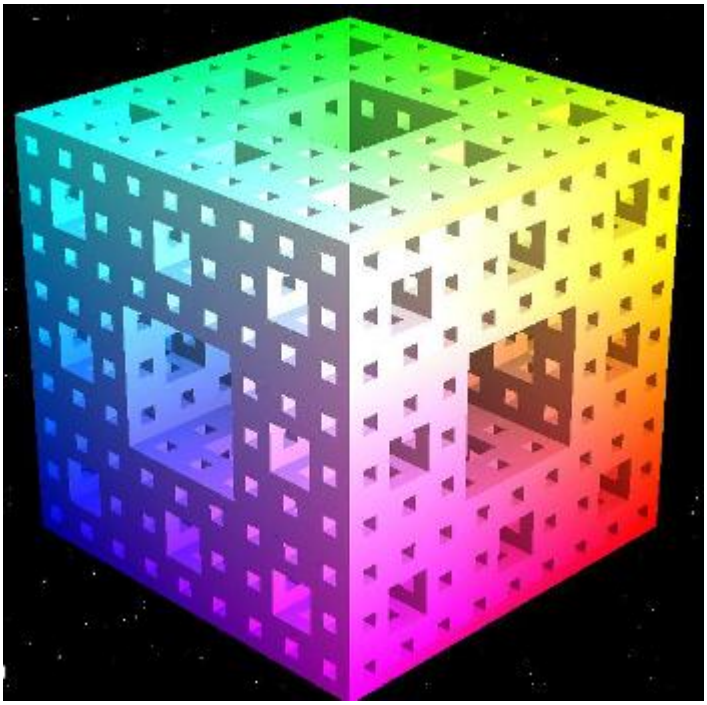


图 4.12 谢尔宾斯基/门格尔海绵

§4.5 Durer五边形

第一个分形例子是由文艺复兴时期德国著名画家丢勒(朱光潜译作“杜勒”)给出的。丢勒祖籍匈牙利，他 1471 年 5 月 21 日生于德国纽伦堡(Nuremberg)。其父精于金银细工，丢勒自幼随父学艺，除金银细工外兼习绘画。1490-1494 年游学德国南部及瑞士，1495 年到意大利威尼斯学习艺术。1505-1507 年第二次前往意大利学习美术技艺和理论。

丢勒在木版画、铜版画、油画、绘画理论方面多有建树，他各个时期的自画像是最全的也是最细致准确的。恩格斯(F.Engels,1820-1895)曾高度评价过他，将他与达芬奇并称：“在思维能力、热情和性格方面，在多才多艺和学识渊博方面的巨人。”他的学生伊拉斯谟 (D.Erasmus,约 1466-1536)称他为“黑线阿贝勒斯”(the Apelles of black lines)，

阿贝勒斯是公元前 4 世纪希腊著名画家。据说这一评语是能给予一个艺术家的最高褒奖。

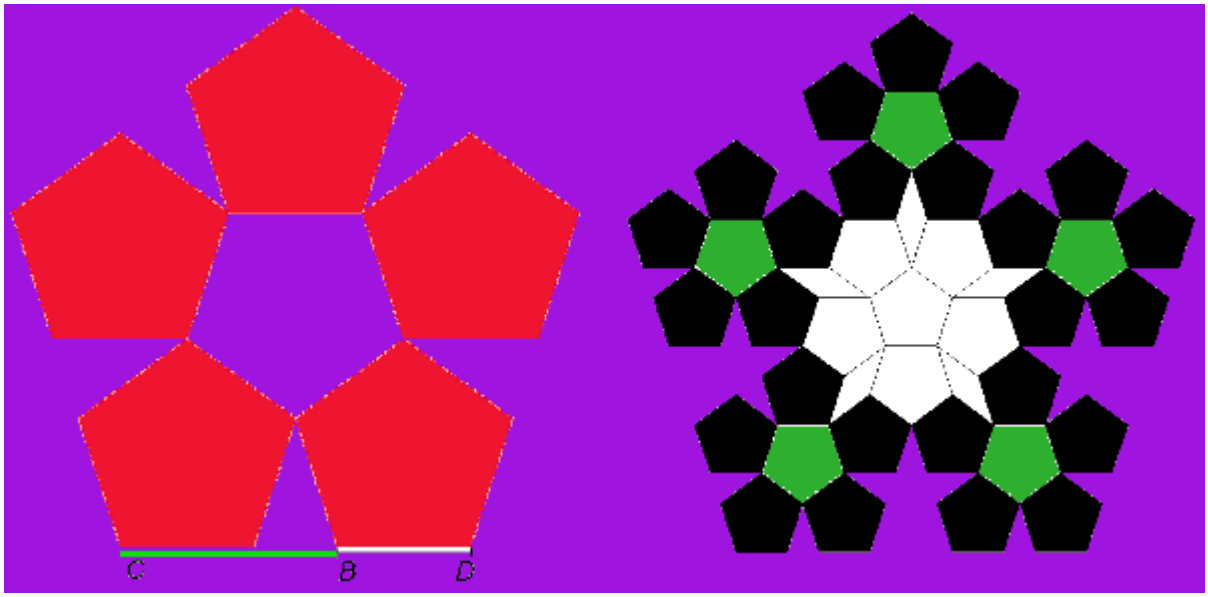
丢勒的艺术创作有一个重要特点：精细，讲究科学和数学。这也许与他自幼随父操持金银细工有关，做这种手艺来不得半点马虎。另外，这也与当时文艺复兴时期重视自然科学和数学的时代风气有关。朱光潜曾说：“意大利绘画在文艺复兴时代之所以能达到欧洲第一次高峰，在很大程度上是科学技术进展的结果。当时一些重要的艺术家都同时是科学家，……文艺复兴时代对形式技巧的追求，尽管有它的形式主义的一面，……它毕竟是艺术发展史上的一个进步运动，因为它使艺术技巧结合到自然科学，实际上起了推动西方艺术向前迈进的作用 。”

丢勒热爱数学，仔细研究过“比例”问题，并把它们用于绘画艺术。他说：“美究竟是什么，我不知道。”“我不知道美的最后尺度是什么。”但他认为这个问题可以用数学来解决。他说：“如果通过数学方式，我们就可以把原已存在的美找出来，从而更接近完美这个目的。”这与当时意大利艺术界对于美具有普遍性、规律性的看法是一致的，也表明他们对于自然科学和数学的信心。他们当时认识到艺术既然是摹仿自然、再造第二自然(达芬奇用过“第 二自然”这一术语)，就要把艺术摆在自然科学的基础上。“这句话有两层意义：头一层是对自然本身要有精确的科学的认识，其次要把所认识到的自然逼真地再现出来，在技巧和手法上须有自然科学的理论基础。”(朱光潜)

丢勒的艺术创作对今天的艺术界似乎仍有几点启示：1)细致观察，画风严谨；2)主动用自然科学和数学武装自己；3)恰当处理自然主义与形式主义的关系。

最后看一下丢勒构造的第一个分形实例(仅限于目前的资料)。他首先画出一个正五边形，然后沿每边向外再作 5 个正五边形，这样又构成了另一个更大的正五边形轮廓。原边长 *BD* 与新边长 *CD* 之比为：

$$r=BD/CD=1/(2+2\cos 72^\circ)=(3-\mathrm{SQRT}(5))/2$$



现在反过来操作，给定一个很大的正五边形，设边长为 *CD*，由 *CD* 计算出内嵌的 5 个小正五边形的边长 *BD*，画出这 5 个小正五边形(如果算上中间的一个小正五边形 ，则正好是 6 个)。不断重复此过程，确实可以得到有无穷自相似结构的分形对象。丢勒正五边形分形显然有五次旋转对称性。

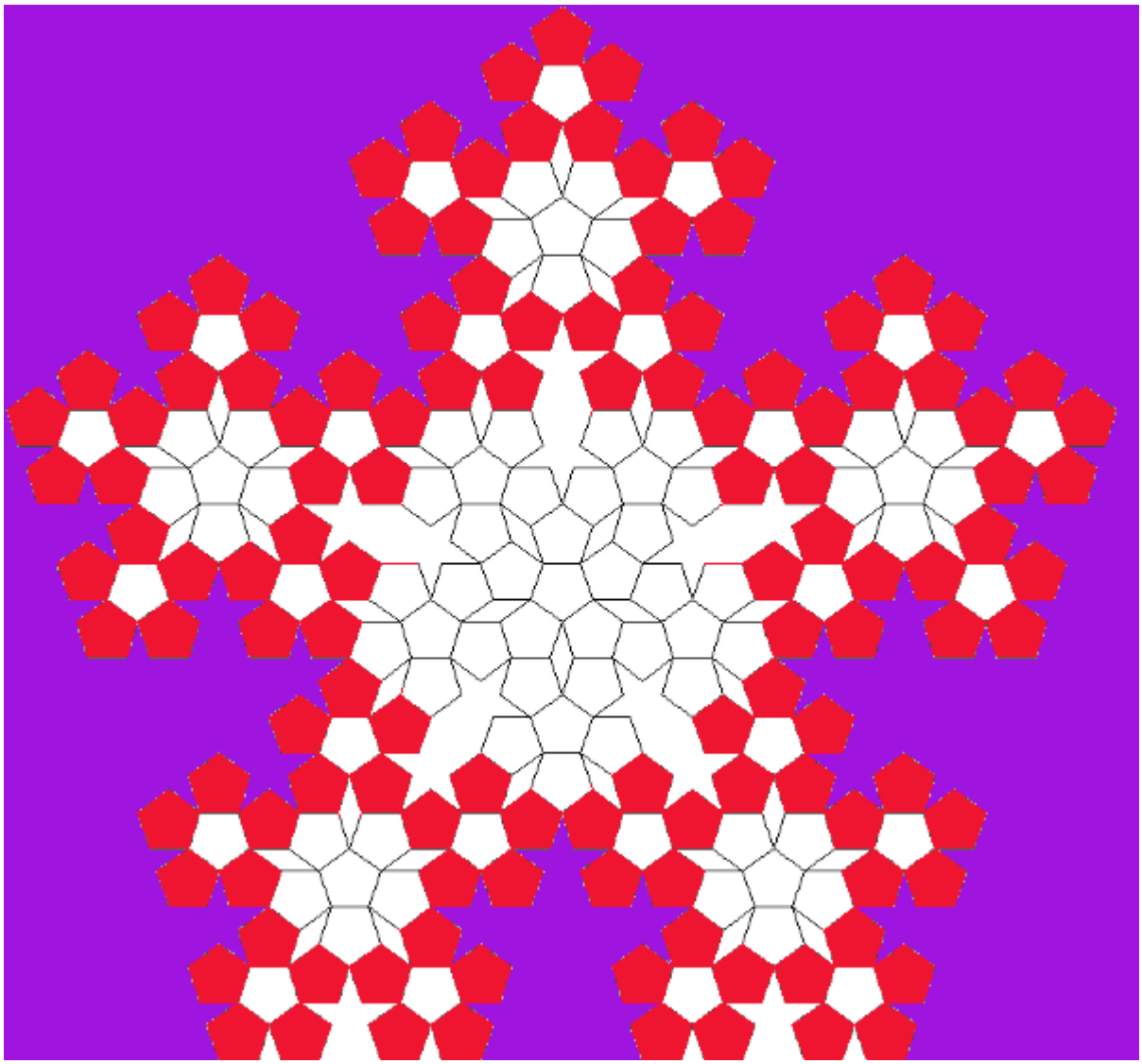


图 4.13 按照著名画家丢勒的想法构造的五边形分形

在 20 世纪，英国著名科学家彭罗斯(R. Penrose, 1931-)发展了“铺砌理论”(tiling theory)，有许多铺砌图案有五次旋转对称性，非常类似于当年丢勒的构造。等到讲“L 系统”时， 还要专门提“彭罗斯铺砌”。

§5.1 树木曲线

树木曲线是通过递归、反复分枝产生的图形，既有适当的复杂性又有微妙的均衡性，给人以不可思议的美感。

我们用二维数组 branch 指定分枝角度和长度的缩小率，第一元素为分枝角度，第二元素为缩小率，如

branch[ ][2]={{30.0,0.7}, {-30.0,0.7}, END}

来指定即表示这样的分枝：一枝朝向对于当前角 30 度的方向，缩小率为 0.7，另一枝朝－30 度的方向，缩小率也是 0.7，从而朝二方向分枝，左右对称地生长。END 的意思与前面相同，是表示数组结束的符号常量。在主函数 main 中，设定描画开始点（根的位置及当前角），根的位置由外部变量 sx,sy 赋予。假定在描画垂直于地面直立的树木时，当前角均设定为 90 度，然后调用递归函数 tree 来描画从根到第一个分枝点的树干，由外部变量 leng1 指定其长度，这就是一次树木曲线，如图 5.1.1 所示，到此为止和 branch 的值无关。与 branch 发生关系从下面开始，随着第一次函数 tree 的递归，树干的顶端伸长出由 branch 指定的二个分枝，这就是形成的二次树木曲线，以后由于不断调用函数 tree，从而不断伸长出分枝，这便描画出 3 次、4 次...的树木曲线。

图 5.1.1 树木曲线

描画树木曲线的次数与前一样由外部变量N决定，同样，对递归的计数参数n是tree的参数。由每次递归来描画的这种程序可以有效地使用各种各样的色彩，用此程序可以对长度不同的分枝涂以不同的颜色，如

树干用 1 号蓝色,最初的分枝用 2 号绿色,下面的分枝用 3 号浅蓝色,...,最后的分枝顶端的果实涂成红色状，如图 5.1.2 所示，其源程序如下

图 5.1.2 树木曲线

当然，树木曲线也可以不对称，例如图 5.1.3 就是不对称的，它对应的数据见说明。

图 5.1.3 不对称树木曲线

```
int N=7;        /* 递归次数 */
double sx=320.0,sy=390.0; /* 开始点坐标 */
double leng1=130.0;   /*根到第一分支点的长度*/
double
branch[ ][2]={{20.0,0.7},{-30.0,0.6},END}
```

另外，树木的分枝也可以任意规定，如图 5.1.4 就是一个 4 分枝树木曲线，其形状象秋天盛开的花，其对应的数据见说明。

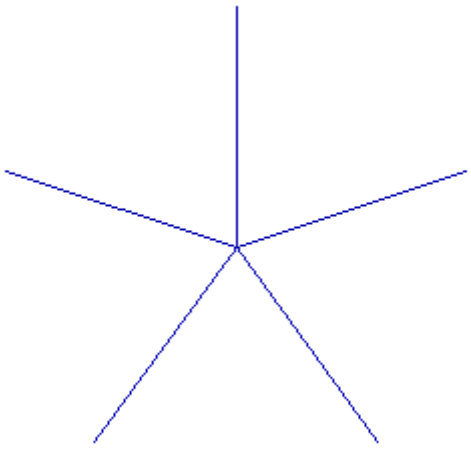
图 5.1.4 象秋天盛开的花形状的树木曲线

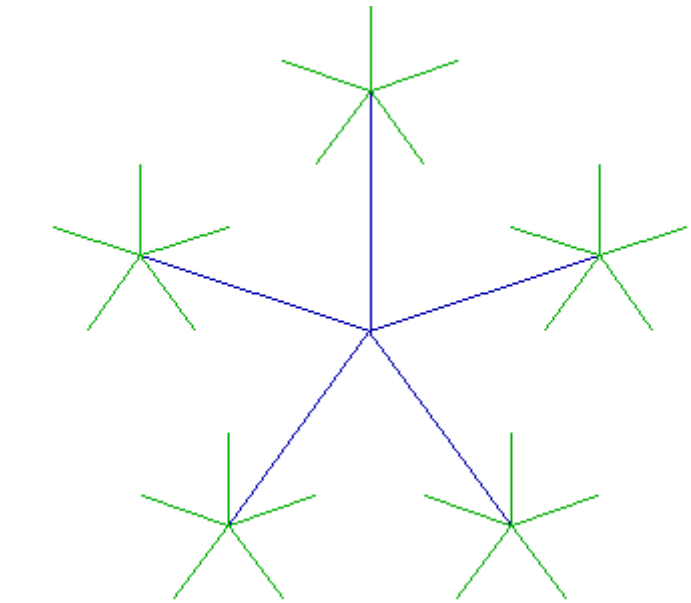
```
int N=7;        /* 递归次数 */
```

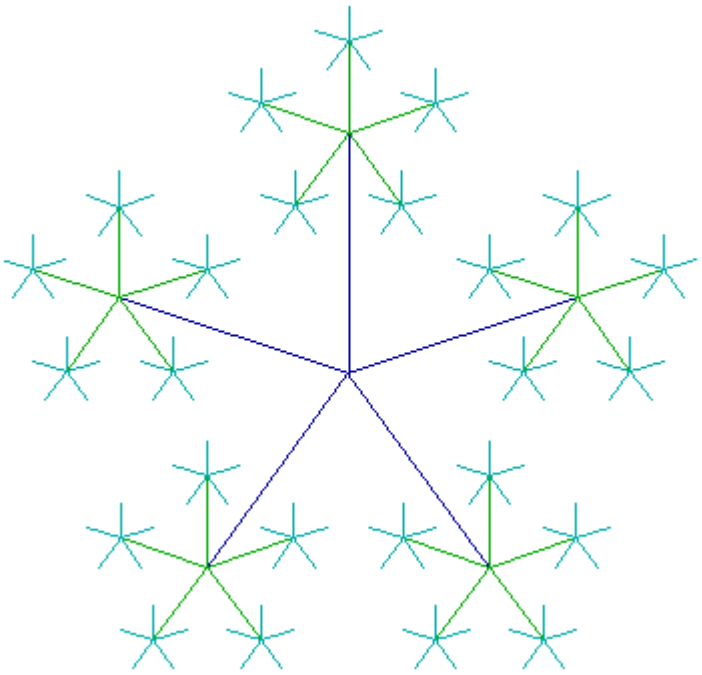
```
        double sx=320.0,sy=390.0; /* 开始点坐标 */
        double leng1=120.0;   /*根到第一分支点的长度*/
        double
branch[][2]={{45.0,0.7},{15.0,0.7},{-15.0,0.7},{-45.0,0.7},EN
D}
```

§5.2 以线段为构成元素的图形

前一节描画树木曲线时，我们先描画树干，再从树枝顶端开始描画分枝。现在，我们删去树干，即从分枝处直接描画，而程序并没有很大变化，但是，所画的图形一看上去就会感到有很大的不同。这时，描画的开始点总固定在画面的中央，整个图形成放射状，再加上旋转的对称性这个条件，这时没有必要改变每次分支的缩小率，其枝长的缩小率统一由外部变量scale给定。如图 5.2.1 就是一个以线段为构成元素的 5 次对称递归图形，其源程序如下:







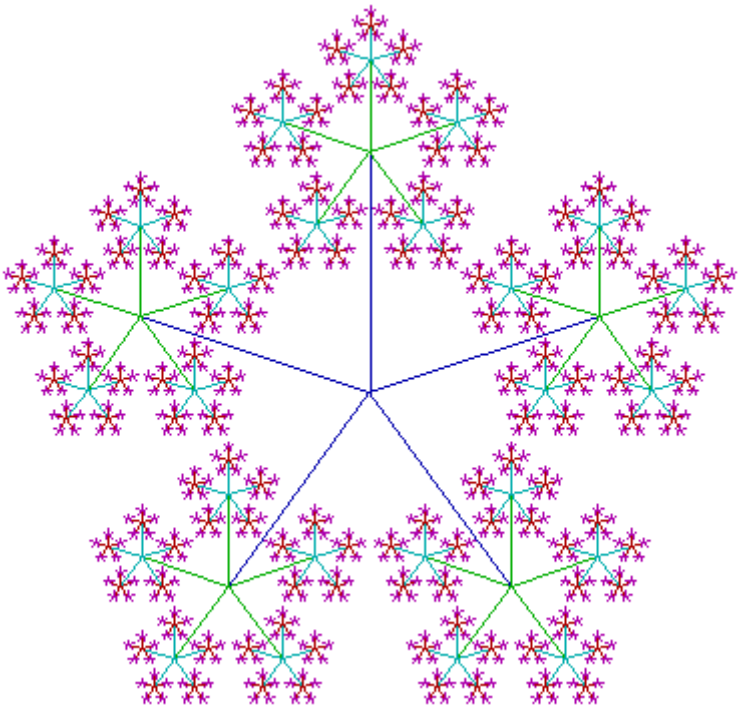


图 5.2.1 线段为构成元素的 5 次对称递归图形

要注意的是，对树枝的分枝角度的处理应根据不同的情况作不同的处理。如果描画 n 次对称图形的话，那么分枝角度也就定下来了，用外部变量 n_rot 指定要描画的对称图形的次数，因此，指定分枝的数组 branch 就不需要了。另外，要根据 n_rot 是奇数还是偶数，其分枝角度要错开，以免重画。规定 N 为描画的递归图形的次数，leng1 为中心到最近分枝点的距离。图 5.2.2 及图 5.2.3 分别为 6 次和 8 次对称的递归图形，其相应的数据见说明。

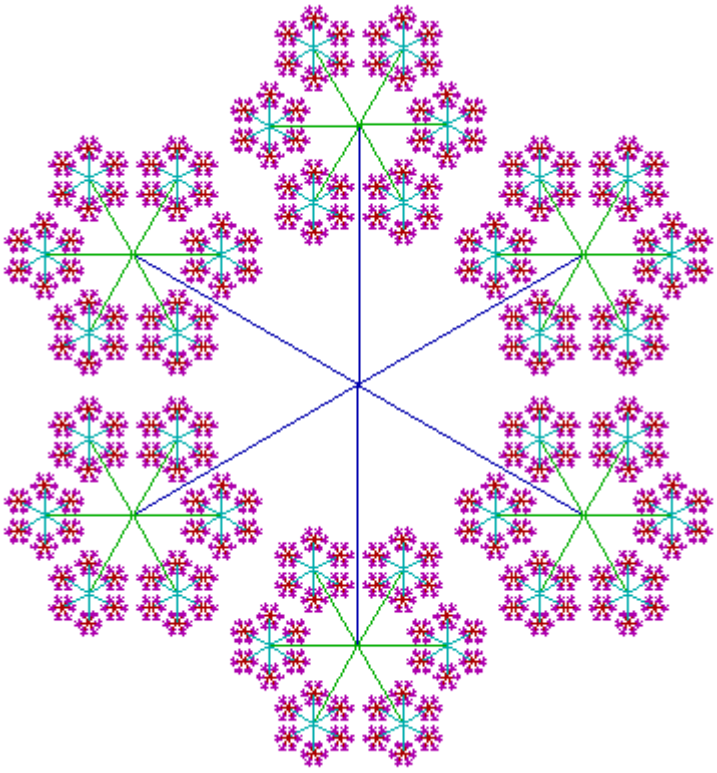


图 5.2.2 线段为构成元素的 6 次对称递归图形

```
int n_rot=6; /*旋转对称次数*/

int N=5;       /* 递归次数 */

double ang, scale=0.34; /* scale 缩小率 */

double leng1=130.0;   /*中心到下一分支点的长度*/
```

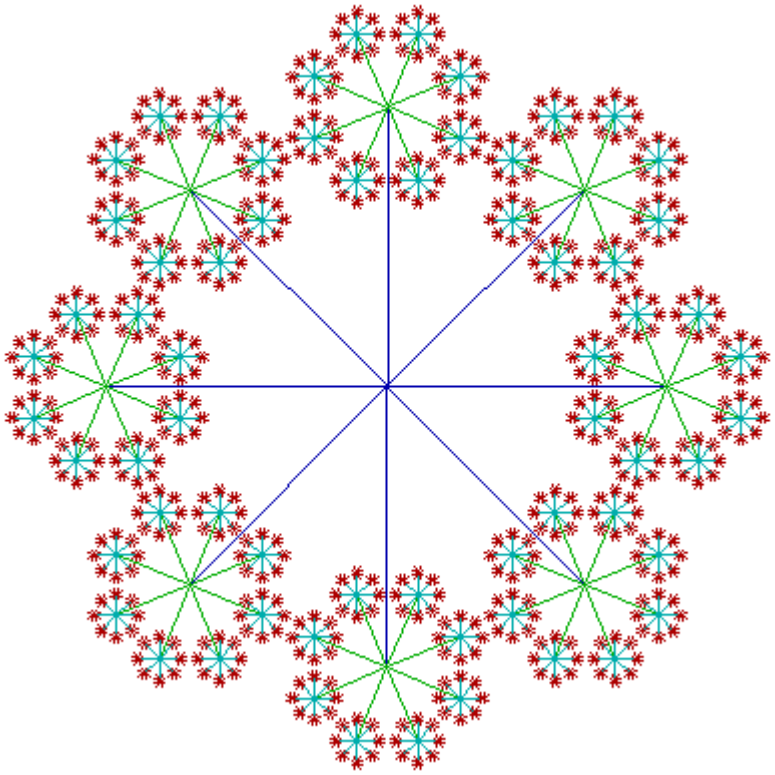


图 5.2.3 线段为构成元素的 8 次对称递归图形

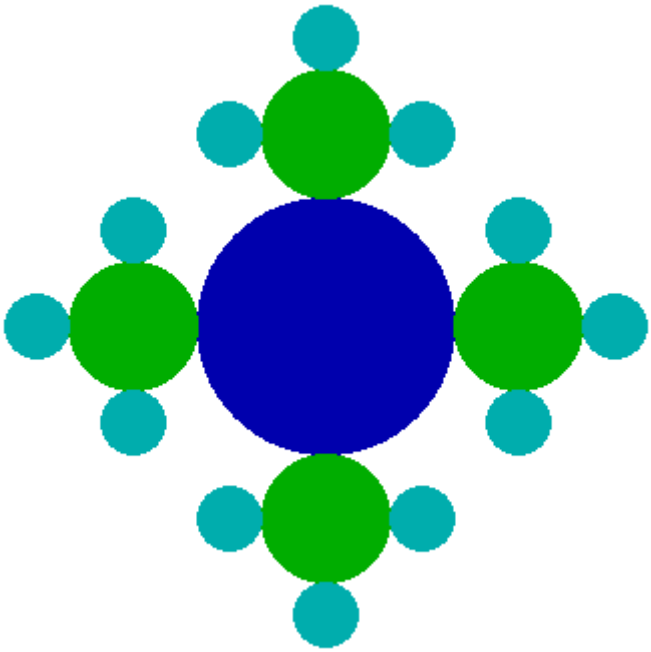
```
int n_rot=8; /*旋转对称次数*/

int N=4;       /* 递归次数 */

double ang,scale=0.28; /* scale 缩小率 */

double leng1=140.0;  /*中心到下一分支点的长度*/
```

§5.3 以圆为构成元素的图形

前一节的递归图形的构成元素是线段，这儿我们把线段改为圆，即要描画不同半径的圆，首先在画面中央描画半径为 r1 的圆，然后在其

周围描画由 n_rot 指定个数的 2 次圆与其圆周外接，半径的缩小率由 scale 指定，因为圆以现在位置为中心描画，所以这时半径这部分在函数 warp 的参数要尽可能大。

要注意的是，多次递归后有些圆会与其旁边的圆重叠，所以，真正描画时会描画少一些圆，而其颜色则用其递归次数对应的颜色来区别。如图 5.3.1 是 4 次对称的递归图形，源程序如下。图 5.3.2 和图 5.3.3 为 5 次、6 次对称图形，数据见说明。



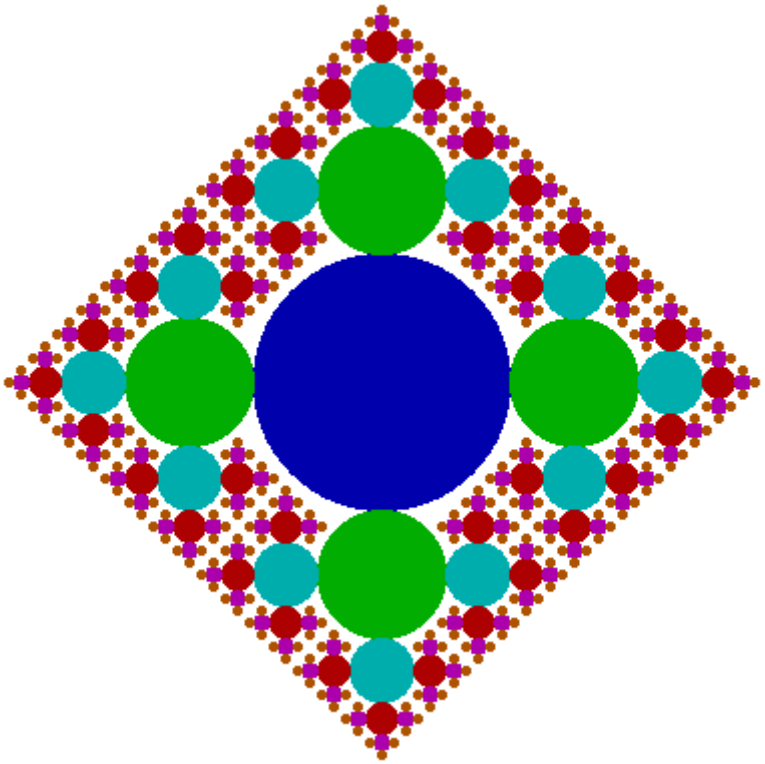


图 5.3.1　以圆为构成元素的 4 次对称的递归图形

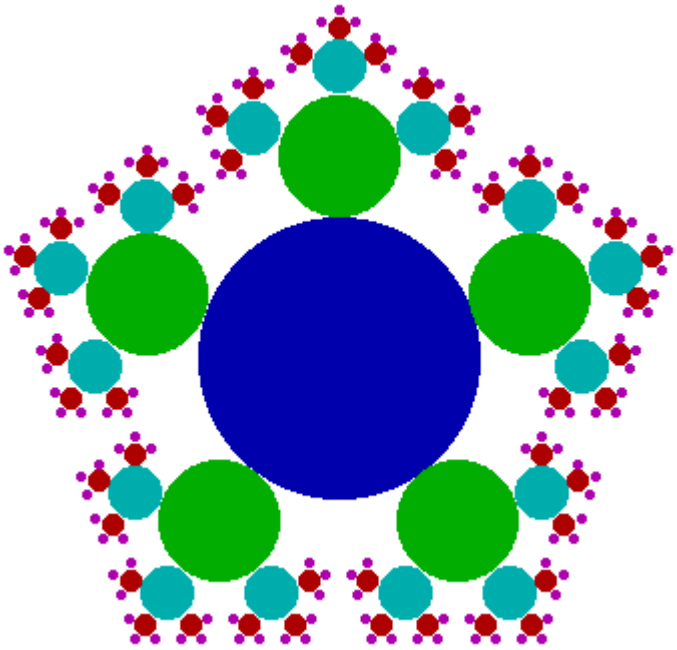


图 5.3.2　以圆为构成元素的 5 次对称的递归图形

```
int n_rot=5; /*旋转对称次数*/
```

```
int N=5;        /* 递归次数 */
double ang,scale=0.44; /* scale 缩小率 */
double r1=70.0;   /*一次圆的半径*/
```

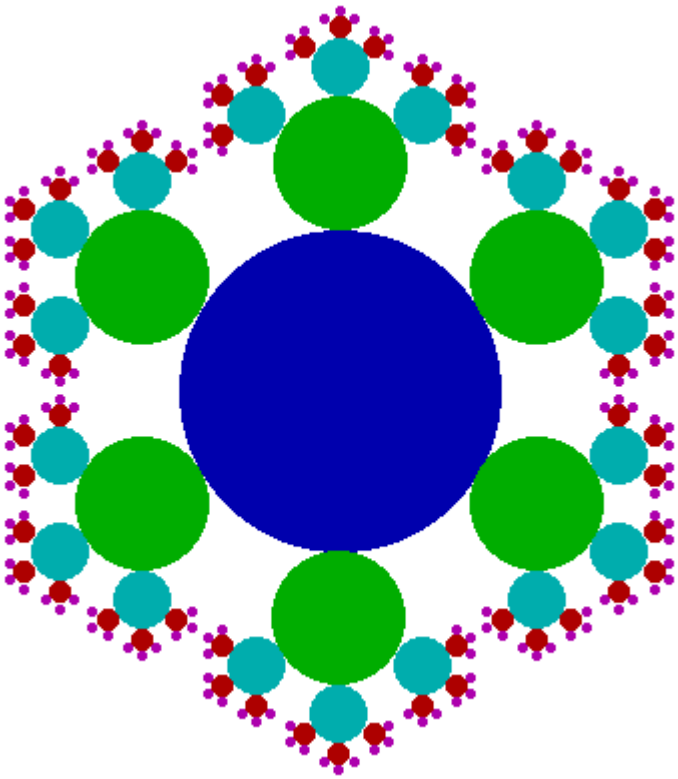


图 5.3.3　以圆为构成元素的 6 次对称的递归图形

```
int n_rot=6; /*旋转对称次数*/
int N=5;        /* 递归次数 */
double ang,scale=0.42; /* scale 缩小率 */
double r1=80.0;   /*一次圆的半径*/
```

前面的递归是向外扩展的，同样地，我们也可以向内收缩，即在 1 次圆画过之后再画的二次圆与一次圆内接且与相邻的 2 次圆也相切，因

此，其缩小率scale自动定出。另外，在中心的空白部分也填入圆，结果是 2 次圆的个数为n_rot加 1。从 2 次圆到 3 次圆时也一样，如此描画出的递归图形很象曼陀罗。如图 5.3.4 是 6 次对称递归图形，源程序如下：

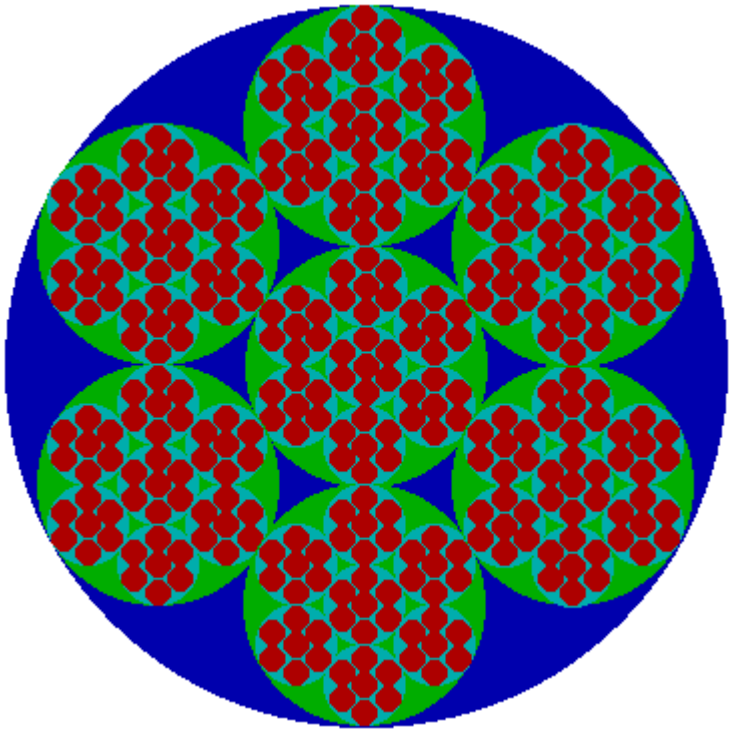


图 5.3.4　6 次对称递归图形

图 5. 3. 5 是 4 次对称递归图形，图 5. 3. 6 是 8 次对称递归图形，其对应的数据见说明。

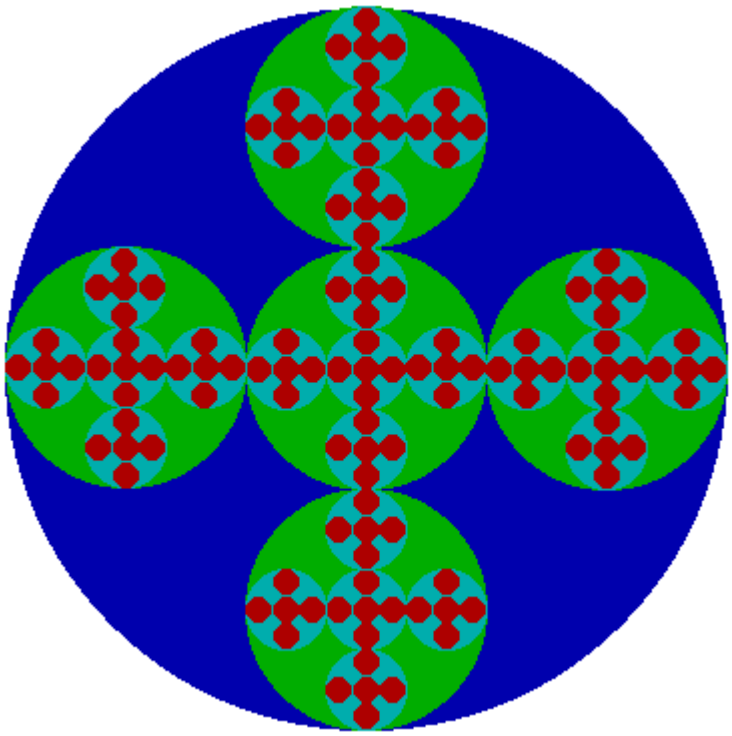


图 5.3.5 4 次对称递归图形

```
int n_rot=4; /*旋转对称次数*/

int N=4;       /* 递归次数 */

double ang,scale=1.0/3; /* scale 缩小率 */

double r1=180.0;   /*一次圆的半径*/
```

注意：这时指定了缩小率，因而沿着 1 次圆排列的 2 次圆，相互之间并不相接，对应的 main 函数中的计算部分要删除。

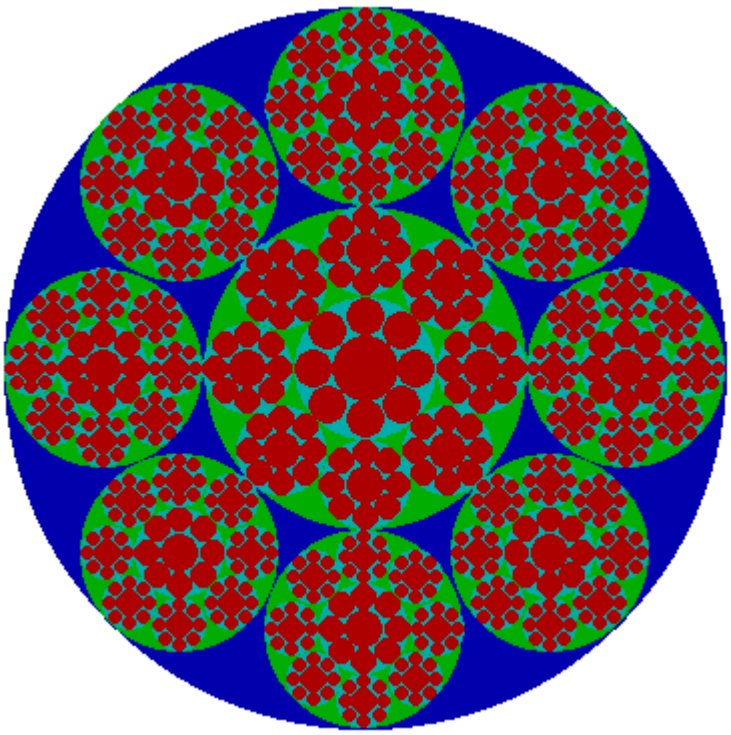


图 5.3.6 8 次对称递归图形

```
int n_rot=8; /*旋转对称次数*/

int N=4;       /* 递归次数 */

double ang,scale; /* scale 缩小率 */

double r1=180.0;  /*一次圆的半径*/
```

§5.4 以多边形为构成元素的图形

现在考虑构成元素为正多边形的递归图形，我们只画外轮廓并进行内部着色。正多边形的边数由外部变量n_rot指定，中心描画的最大正多边形(1 次正多边形)的外接圆的半径由外部变量r1 指定，对称移动操

作所需的最小旋转角由外部变量指定等于除以的值，多边形的一边长度与图 5.4.1 所示的是正三角形为构成元素的递归图形，图 5.4.2 所示的是正方形为构成元素的递归图形，图 5.4.3 所示的是正 5 边形为构成元素的递归图形，其源程序如下：

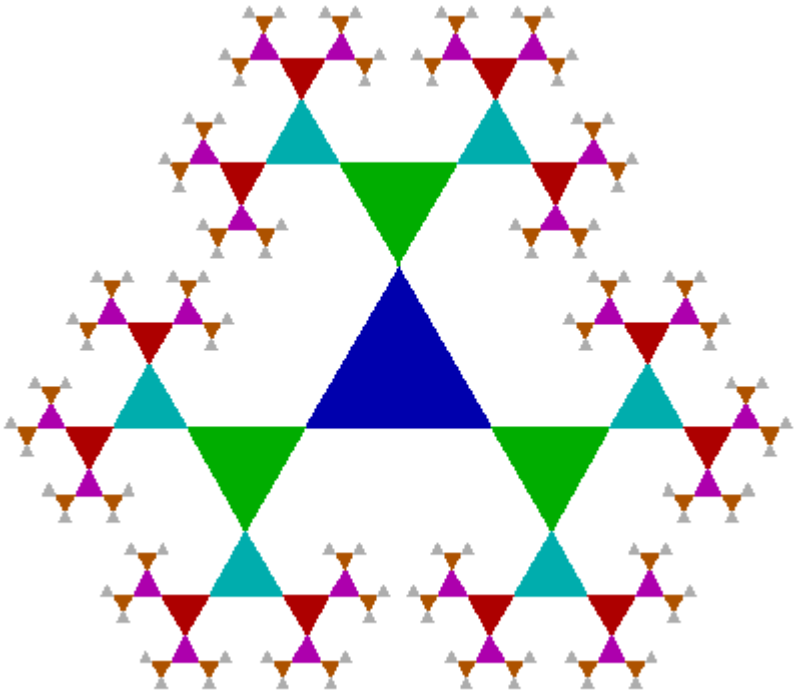


图 5.4.1　正三角形为构成元素的递归图形

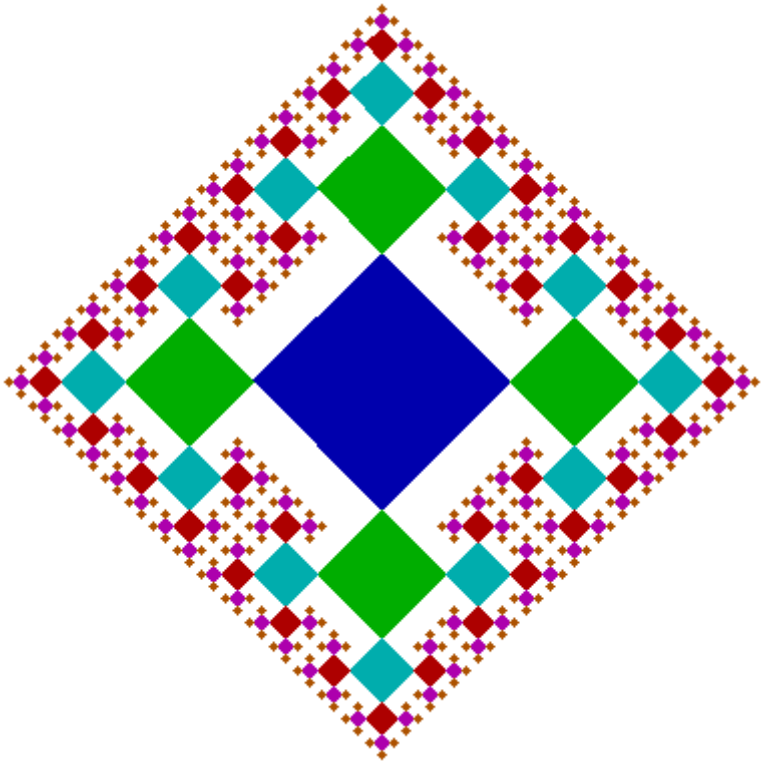


图 5.4.2　正方形为构成元素的递归图形

```
int n_rot=4;   /* 旋转对称次数 */

int N=6;       /* 递归次数 */

double scale=0.5; /* 缩小比例 */

double r1=64.0;   /*一次圆的半径*/

double ang, ks;
```

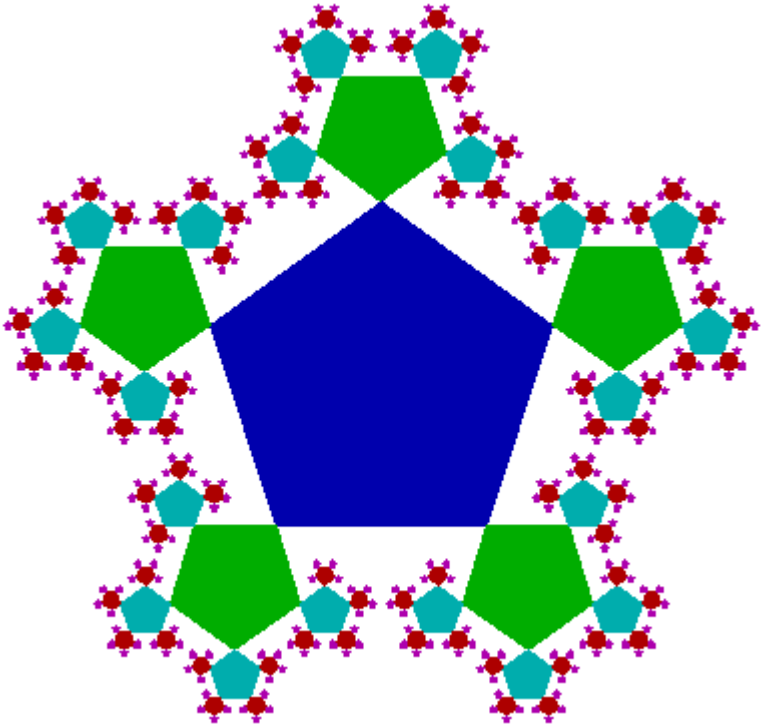


图 5.4.3　正 5 边形为构成元素的递归图形

```
int n_rot=5;   /* 旋转对称次数 */

int N=5;       /* 递归次数 */

double scale=0.38; /* 缩小比例 */

double r1=90.0;   /*一次圆的半径*/

double ang, ks;
```

§5.5 以星形为构成元素的图形

现在考虑构成元素为星形多边形的递归图形，我们只画外轮廓。图 5.5.1 所示的是正五边形为构成元素的递归图形，图 5.5.2 所示的是正

六边形为构成元素的递归图形，图 5.5.3 所示的是正八边形为构成元素的递归图形，其源程序如下：

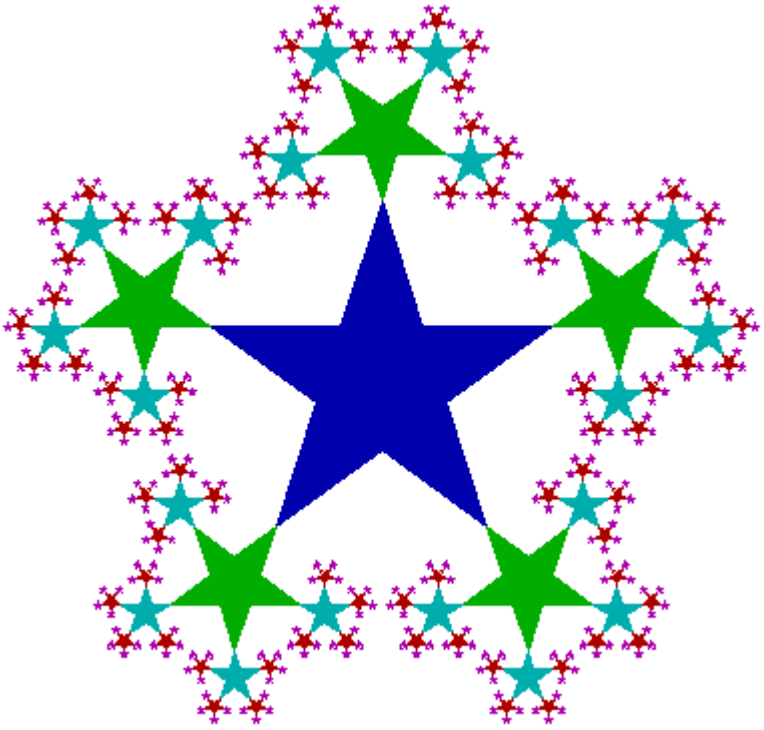


图 5.5.1 星形正五边形为构成元素的递归图形

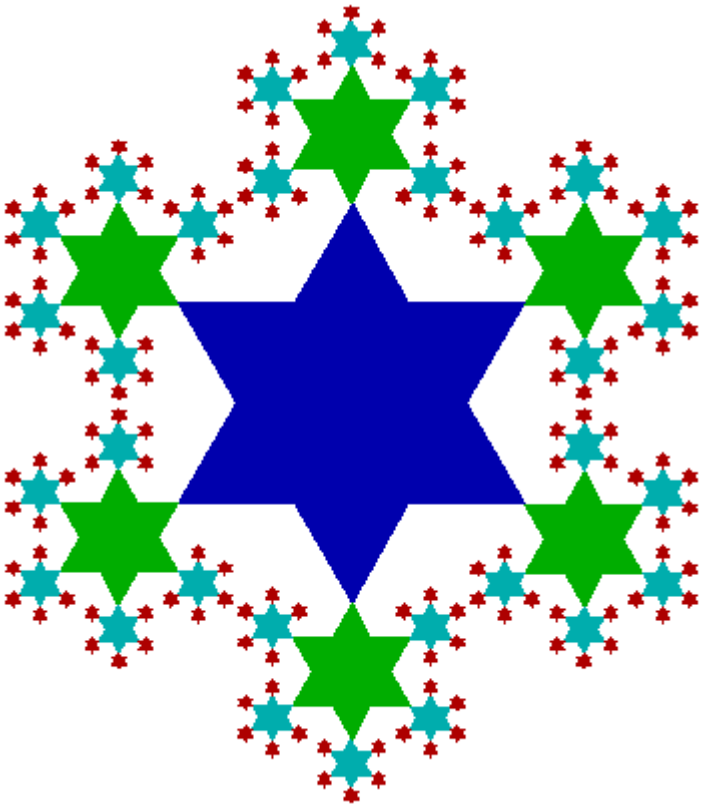


图 5.5.2 星形正六边形为构成元素的递归图形，对应数据如下

```
int n_rot=6; /*旋转对称次数*/
int N=4;    /* 递归次数 */
double scale=0.34; /* scale 缩小率 */
double r1=100.0;  /*一次圆的半径*/
```

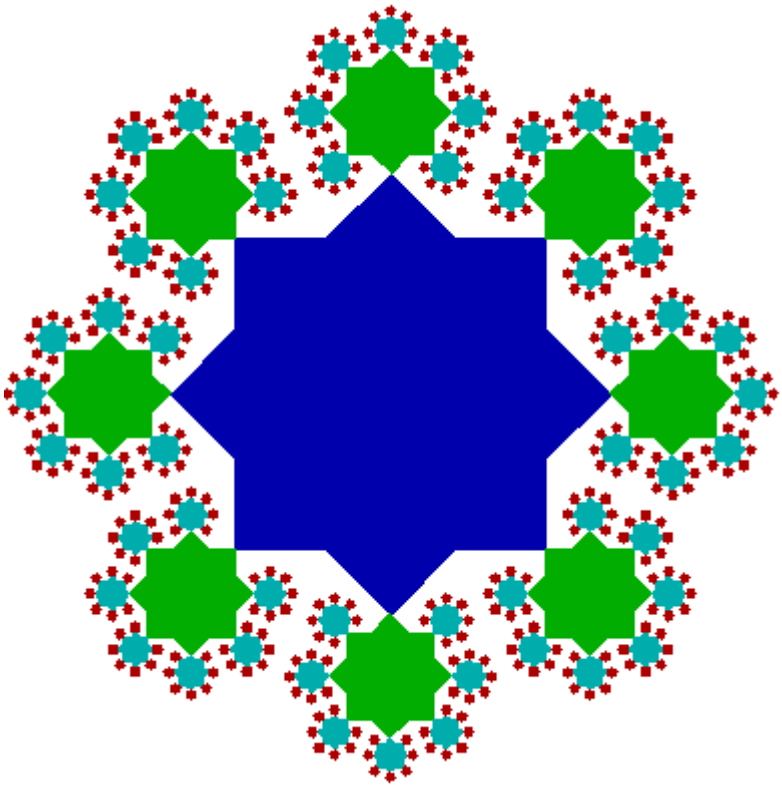


图 5.5.3 星形正八边形为构成元素的递归图形，对应数据如下

```
int n_rot=8; /*旋转对称次数*/
int N=4;    /* 递归次数 */
double scale=0.28; /* scale 缩小率 */
double r1=110.0;  /*一次圆的半径*/
```

前面是对图形内部进行着色，如果我们把所有边都画出来，而着色只局限于图形的中心部分，那么与前面图形相比，它的一个一个突起角变得尖锐了，图 5.5.4 所示的是以星形正七边形为构成元素的递归图形，图 5.5.5 所示的是以星形正八边形为构成元素的递归图形，图 5.5.6 所示的是以星形正九边形为构成元素的递归图形，其相应的源程序如下：

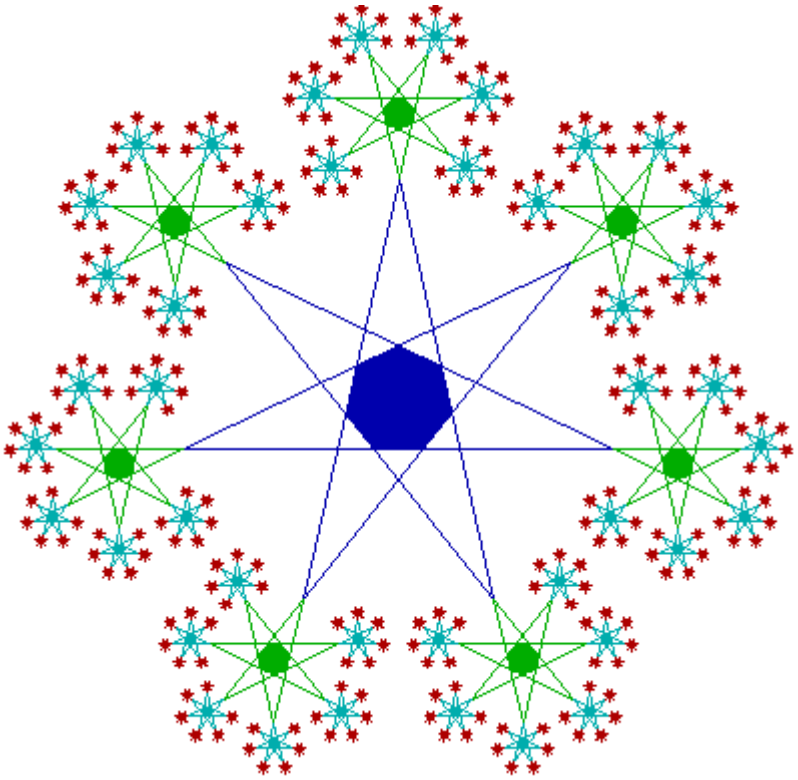


图 5.5.4 星形正七边形为构成元素的递归图形

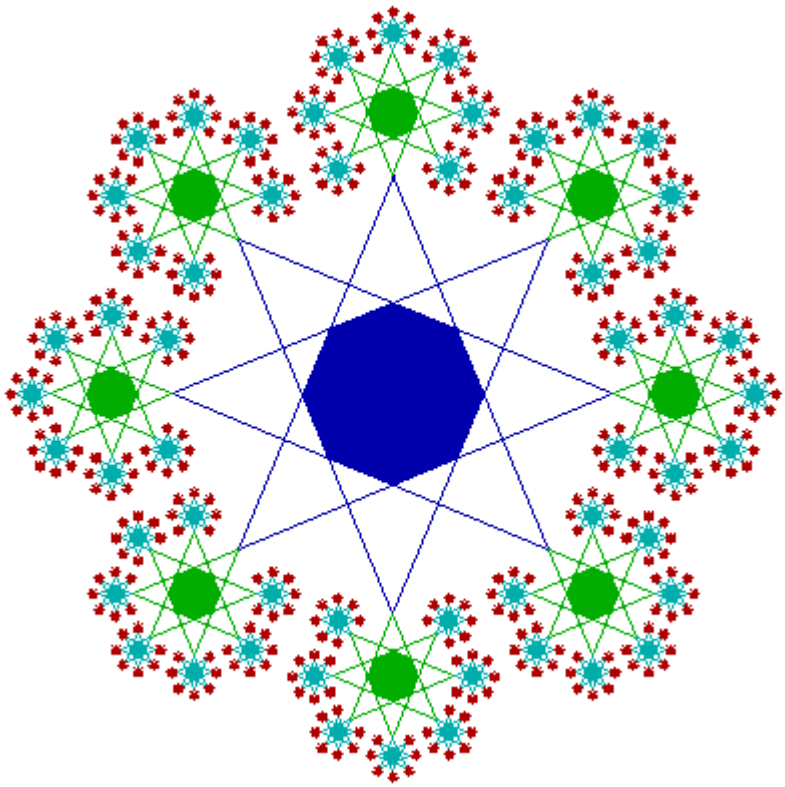


图 5.5.5 星形正八边形为构成元素的递归图形，对应数据如下

```
int n_rot=8; /*旋转对称次数*/
int N=4;    /* 递归次数 */
double scale=0.28; /* scale 缩小率 */
double r1=110.0;  /*一次圆的半径*/
```

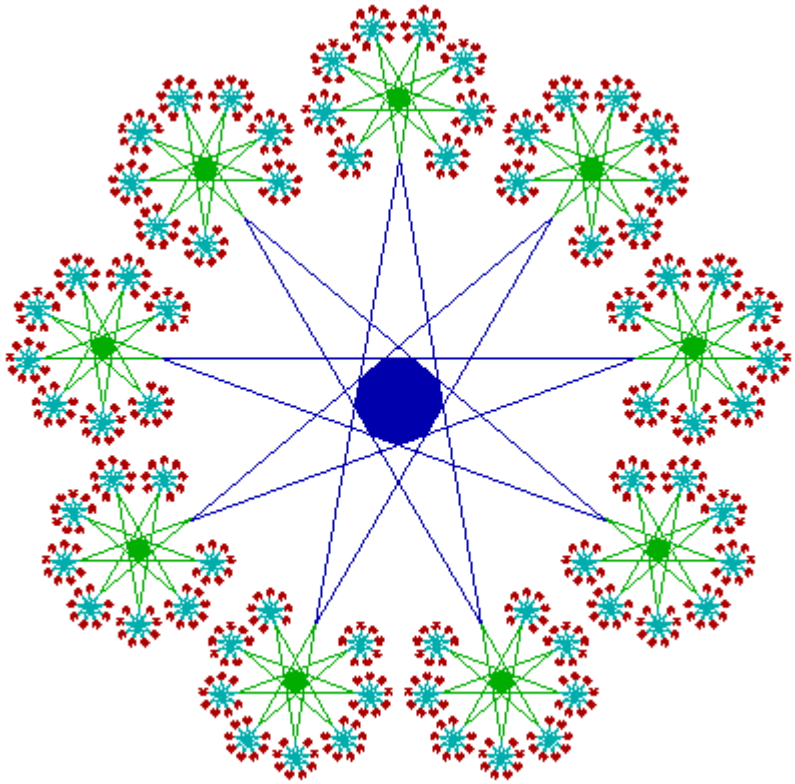


图 5.5.6 星形正九边形为构成元素的递归图形，对应数据如下

```
int n_rot=9; /*旋转对称次数*/
int N=4;       /* 递归次数 */
double scale=0.25; /* scale 缩小率 */
double r1=120.0;   /*一次圆的半径*/
```

要注意的是，当星形的边为 4 的倍数加 2 时(n_rot=6,10,14,…)，星形就不能一笔描画出来，例如要画正六边形的星形时，它就变成相互颠倒的两个正三角形，相信大家能自己找出解决的办法。

§6.1 林氏系统

林氏系统(通常称 L 系统)是林德梅叶 1968 年为模拟生物形态而设计的，后来史密斯于 1984 年 、普鲁辛凯维奇于 1986 年，分别将它应用于计算机图形学，引起生物学界和计算机界人士 极大兴趣，一时发表了许多论文和专著。在国内《植物形态的分形特征及模拟》(常杰(1962-)等著)、《分形及其计算机生成》(齐东旭著)等书，都将 L 系统作为重要内容讲解。

L 系统实际上是字符串重写系统。我们把字符串解释成曲线(或者更准确地说，称作图形)， 于是只要能生成字符串，也就等于生成了图形。L 系统是极其有趣的，第一，用这种方法能 够生成许多经典的分形，第二，用它可以模拟植物形态，特别是能很好地表达植物的分枝结 构。

从一个初始串(叫做公理)开始，将变换规则多次作用于其上，最后产生一个较长的命令串， 用它来绘图。作用一次，称作一级(order)，一般说来选择的级数不宜太高，通常选 2-8 级， 最多 15 级。

L 系统细说起来也有若干种，通常指“0L 系统”。0L 系统可定义为一个三元组〈V, ω, P〉，设 V 表示“字母表”(alphabet)， V^*表示 V

上的所有“ 单词”(words)， $\omega \in V^*$是一个非空的单词，称作公理(axiom)，P 包含于 $V \times V^*$是产生规则的有穷集。

显然 0L 系统是确定性系统，当且仅当对每一个 $c \in V$,存在一个 $s \in V^*$，使得 $c \rightarrow s$.

L 系统的符号串也称“龟图”(turtle),龟图的状态用三元组(X, Y, D)表示，其中 X和 Y分别代表横坐标和纵坐标,D代表当前的朝向。令 δ 是角度增量，h是步长。符号串的图形学解释 (也可以有其他解释，这里只是一种可能的解释)见表 5.1。

表 5.1 L 系统的符号规定与解释

符号	图形解释
F	从当前位置向前走一步，同时画线
G	从当前位置向前走一步，但不画线
+	从当前方向向左转一个给定的角度
-	从当前方向向右转一个给定的角度
|	原地转向 180°
[	Push,将龟图当前状态压进栈(stack)
]	Pop,将图形状态重置为栈顶的状态， 并去掉该栈中的内容
\nn	增加角度 nn 度
/nn	减少角度 nn 度

Cnn	选择颜色 nn
＜nn	在此基础上增加颜色 nn
＞nn	在此基础上减少颜色 nn
!	倒转方向(控制+,-,/)
@nnn	将线段长度乘以 nnn，nnn 也可以是简单函数
其他	也是合法的，主要用于获得复杂的解释

§6.2 实例与伪码

自己编写一个好的 L 系统程序不是很难，但没有什么必要。我们直接借用 Fractint 19.5 中的 L 系统(参见后面章节)，并采用那里的伪码约定。用“伪码”表示各种 L 系统的生成过程，这样编写程序就简单得多。伪码虽然不能直接运行，但解释程序可以直接读取这种 ASCII 文件，然后进行计算、绘图。这样做的好处是显然的，更改规则轻而易举，不懂得 C 或者 PASCAL 编程也没什么关系。

在这种伪码中，先规定几个符号。设角度增量值通过 360/n 的方式获得，比如角度增量为 90° 时，可取 n=4。每一行中“；”之后是程序注释，不参与编译。每一个程序 均以“程序名”开始，然后是以“{ }”括起来的程序体。下面看

例 1:

```
Koch1 {                              ; 程序名
```

Angle 6 ; 规定角度增量为 360/6=60
。

Axiom F--F--F ; 通过公理给出初始图形：一个
倒置的正三角形

F=F+F--F+F ; 通过产生式给出代换规则;

}

以上述柯赫曲线为例，L 系统的“代入”过程可以通过“符号串替换”表示。将公理“F--F- -F”中的“F”不断用“F+F--F+F”代换，第一次代入得到：F+F--F+F--F+F--F+F--F+F-- F+F。第二次代入得到：

F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F-
-F
+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F

可以看出，符号串的增长速度是相当快的。第三次代换便得到如下符号串：

F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--
F+F--
F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--
F+F+
F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F
+F+F--

F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--

F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--

F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F--

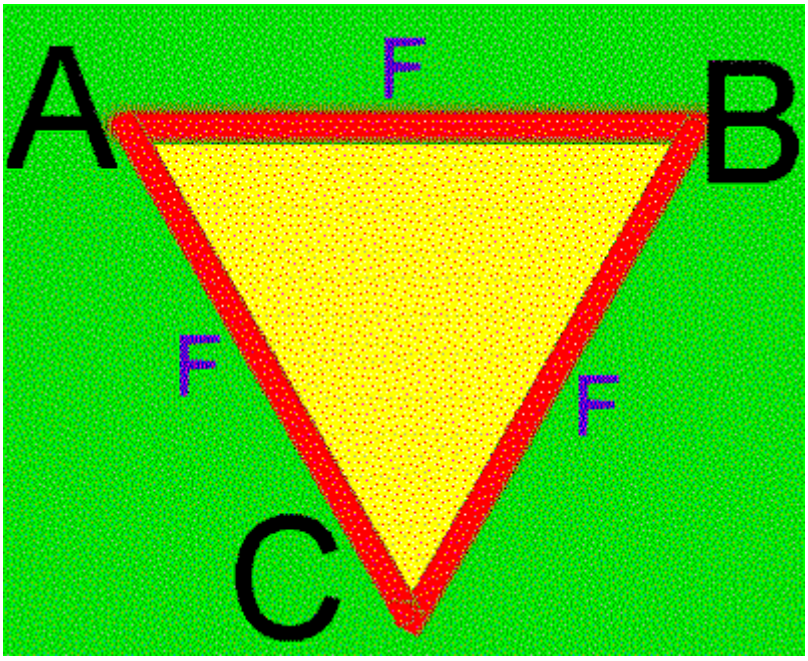
F+F+

F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+

F+F-- F+F--F+F--F+F+F+F--F+F

对于某一给定的替换级别(不宜太高，一般不超过 15 级)，有了这样的符号串，按照每个符号的解释，就可以作图了。可以随便找一个“编辑器”(如 WPS，FE，EDIT，或者 PASCAL 的 IDE) ，通过编辑器的“查找替换”功能方便地得到任意级别的符号串，然后将符号串存成一个文本文件，绘图程序从此文件中读取信息，便可完成 L 系统的绘图。当然，也可以在一个单一的程序内部完成这一系列操作。

我们先看一下公理为什么是一个倒置的正三角形。公理为“F--F--F”，第一个“F”表示向前走一个单位线段(规定从左向右)，得到的是 AB 线段。然后是两个“-”，表示从当前方向开始算起向右转两个 60°。第二个“F”表示沿当前方向再走一个单位长度，得到 BC 线段 。接下去又右转两个 60°，再画一个单位线段，得到 CA 线段。于是得到一个正三角形 ABC。



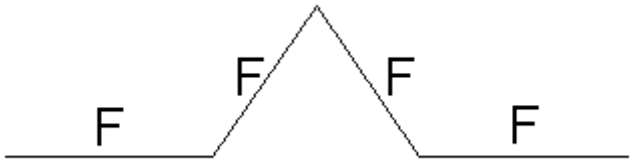


图 6.2.1 上图为例 1 的公理，下图为例 1 或例 2 的生成规则(产生式或代入规则)。

例 2：与例 1 不同之处只在于公理不同，生成规则是一样的。此例可以生成柯赫曲线，初始图形是一条线段，生成过程是将线段中间 1/3 向外折起。程序伪码如下：

```
KochCurve {                ; 柯赫曲线
Angle 6                    ; 角度增量是 60°
Axiom F                    ; 初始图形是一单位线段
F=F+F--F+F                 ; 产生式是将线段中间 1/3 折起
```

```
}                                    ; 结束
```

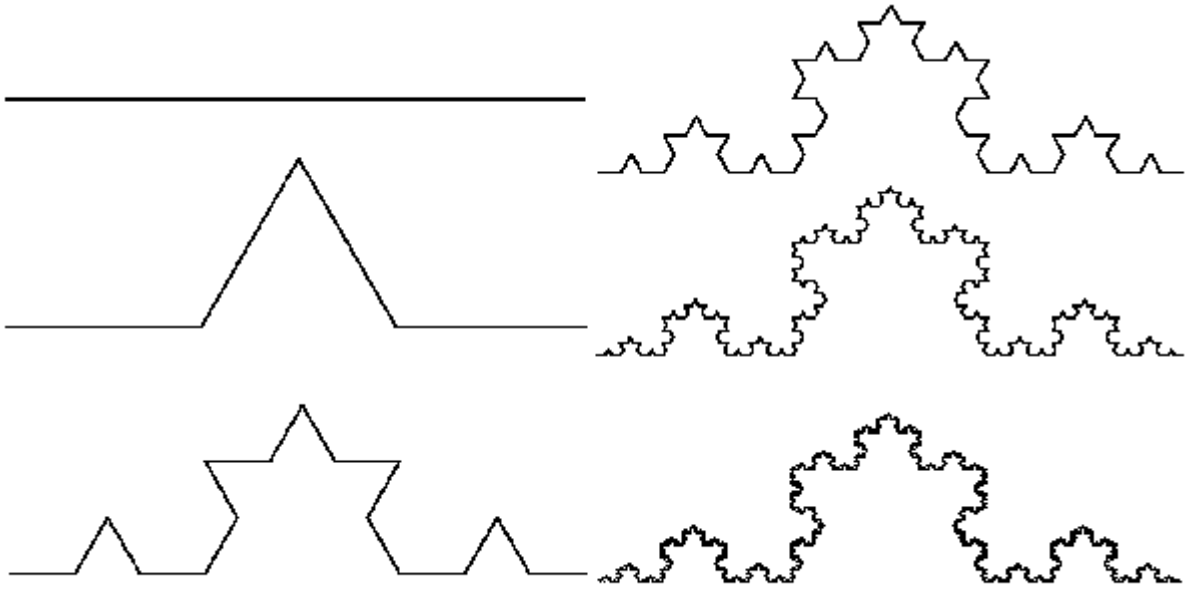


图 6.2.2　柯赫曲线生成过程(见例 2)

例 3: 公理(初始图形)为一正方形，生成元也很简单，向前走两步，右转走一步，回转，走一步，右转，再走一步。图形结构见图 6.2.3。

```
FourTreeTest {                       ; 四方内生树，1995-12
Angle 4                                  ; 角度增量为 90°
Axiom F-F-F-F                        ; 初始图形为正方形
F=FF-F--F-F
}
```

例 4: 我们用 L 系统再次生成希尔伯特曲线。生成希尔伯特曲线的伪码如下:

```
Hilbert{                                    ; 希尔伯特曲线,
1996-12
Angle 4
Axiom Y                                     ; 初始串为任意字母 Y
X=-YF+XFX+FY-                               ; 第一个生成规则
Y=+XF-YFY-FX+                               ; 第二个生成规则, 由以上规则
不断代换
}
```

例 5: 分形龙。迭代 13 次生成的图形见图 6.2.4。生成分形龙的伪码如下:

```
LiuDragon {                                 ;变形的分形龙, 1996-12
Angle 9                                     ; 角度增量为 40°
Axiom FX
F=                                          ; 第一个规则, 含义
为删除 “F”
Y=+FX--FY+                                  ; 第二个规则
```

```
X=-FX++FY-                              ; 第三个规则
}
```

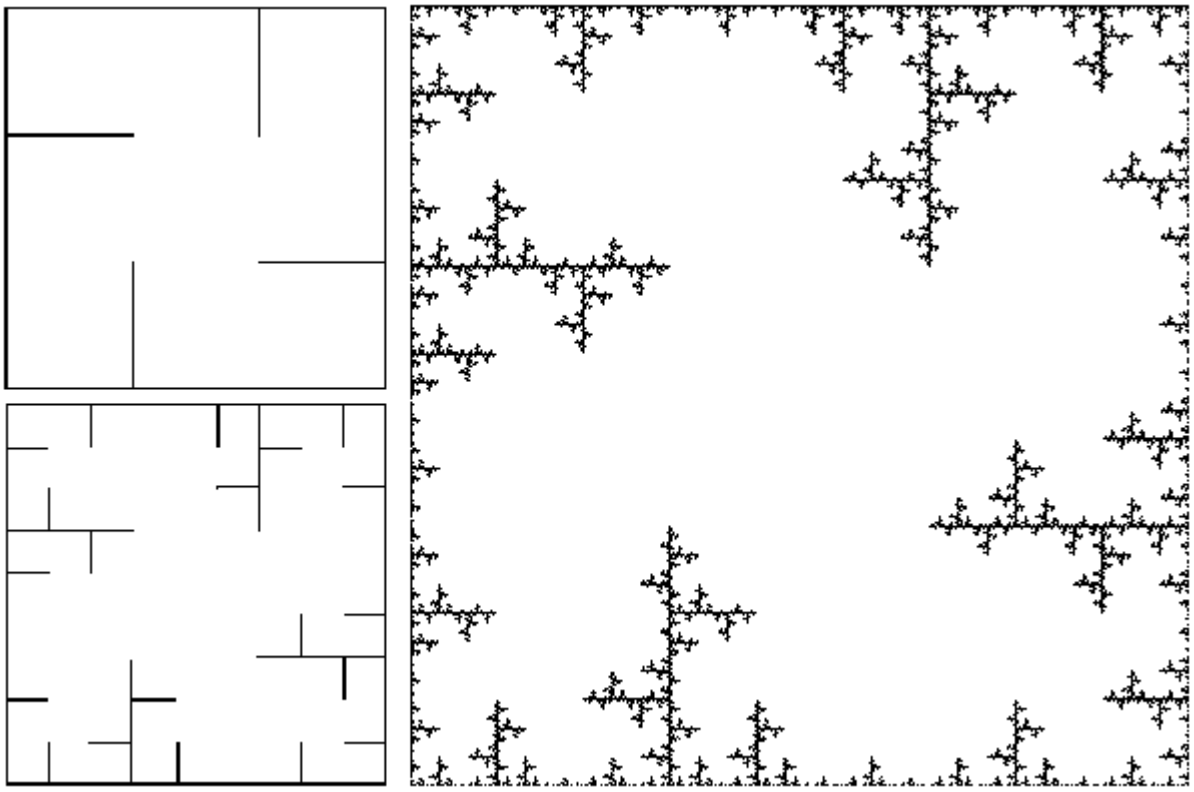


图 6.2.3　四方内生树，右图为代入 8 次后的图形。

例 6: 模拟草本植物。注意这里出现了“括号”——可以方便地表示树枝，见图 6.3.2 右图。伪码如下:

```
HerbPlant {                 ; 生成植物，本程序使用了括号
Angle 14
Axiom Z
Z=ZFX [ +Z ] [ -Z ]
```

```
X=X [ -FFF ] [ +FFF ] FX
}
```

例 7：模拟侧柏形态，见图 6.3.3 左图，伪 码如下：

```
OriArbo {
Angle 18
Axiom F
F=F [ +F ] F [ -F ] [ F ]
}
```

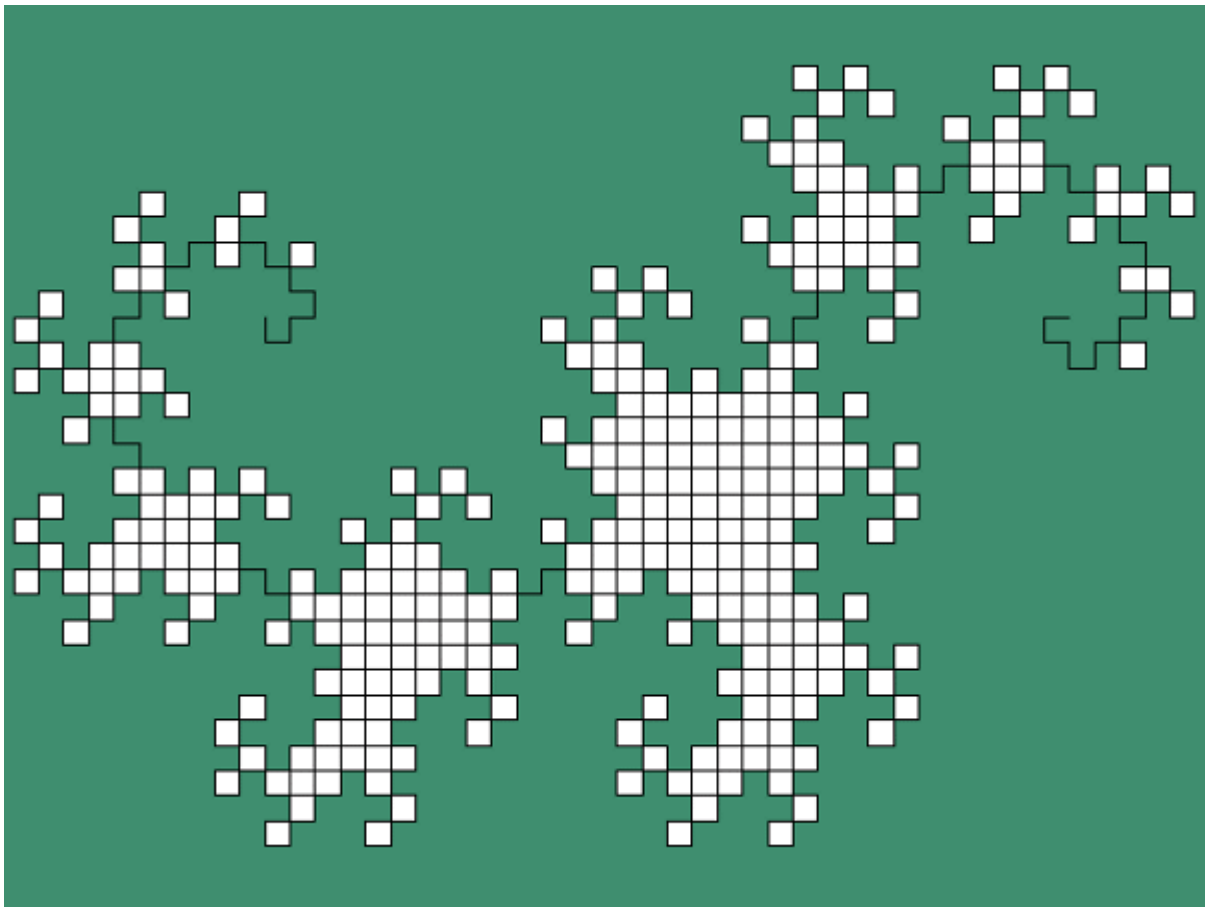


图 6.2.4　分形龙，代入 10 次后的图形。

§6.3 L系统数据表

有了上面两节的基本知识，读者可以自己设计各种类型的 L 系统，从而得到一批新的美妙图形。不过，在此之前还是了解一下前人试验过的数据为好。

下面给出有关 L 系统的一些很有用的数据，包括角度增量(以 n 的形式给出，其中 n 是指 360/n 中的 n)、公理、规则集(对于多个规则，只在首次出现时给出 p_1, p_2, p_3 等标志，后面的例子均省略)等。有关用 L 系统方法生成平面铺砌图形的问题，后面还会讲到。

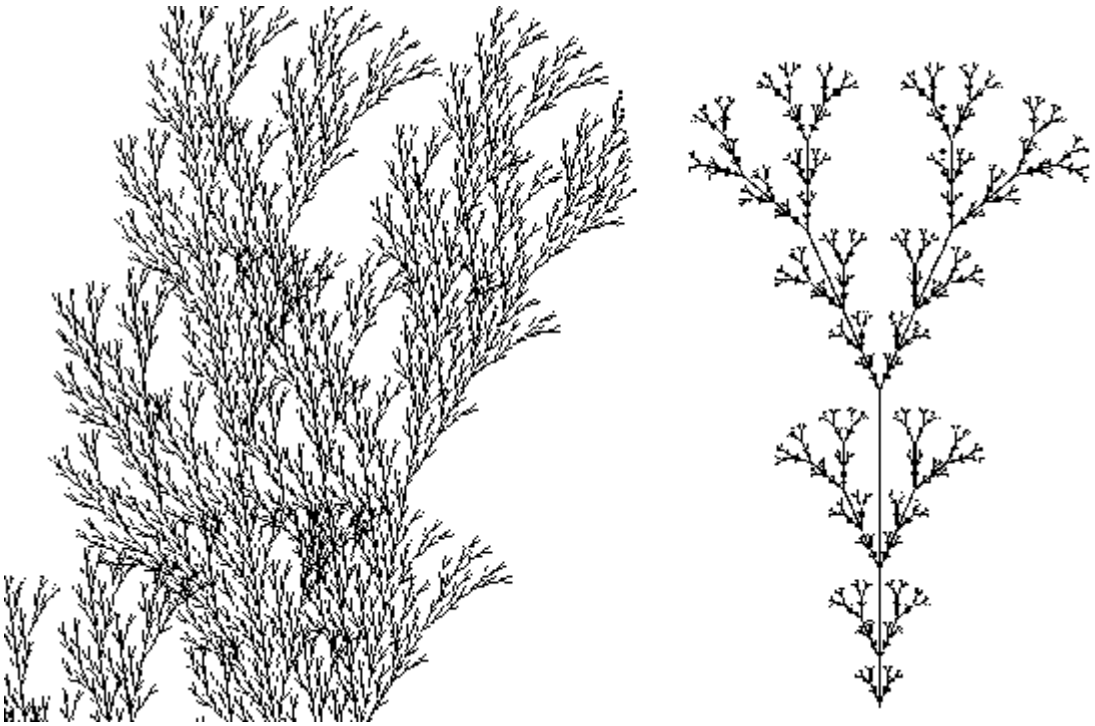


图 6.3.1 用 L 系统方法模拟植物形态

表 6.3.2 各种 L 系统数据一览表

(角度增量指 360/n 中的 n，每个等式表示一个规则)

程序名	角度增量	公理 ω	生成规则 $P=(p\_1, \ldots, p\_n)$
Koch1	6	F--F--F	F=F+F--F+F
Koch2	12	F---F---F---F	F=-F+++F---F+
Koch3	4	F-F-F-F	F=F-F+F+FF-F-F+F
Koch4	12	F++++F++++F	F=+F--F++F-
Koch5	4	F+F+F+F	F=F+F-F-FFF+F+F-F
Koch6	4	F+F+F+F	F=F-FF+FF+F+F-F-FF+F+F-F-FF-FF+F
Dragon	8	FX	p_1: F= p_2: Y=+FX--FY+ p_3: X=-FX++FY- 注：后面不再写 p_1，p_2 等标志
Peano1	4	F-F-F-F	F=F-F+F+F+F-F-F-F+F
Cesaro	34	FX	F=X=----F!X!++++++++F!X!----
DoCesaro	4	D＼90D＼90D＼90D＼	D=＼42!D!/84!D!＼42

		90	
FSnake	6	FL	L=FL-FR--FR+FL++FLFL+FR-",R=+FL-FRFR--F R-FL++FL+FR",F=
CantDt	6	F	F=FGFG=GGG
SnowFk2	12	F	F=++!F!F--F--F@IQ3|+F!F--F=F--F!+++@Q3 F@QI3|+F!F@Q3|+F!F
SnowFC1r	12	F	F=--!F＜1!F＜1++F＜1++F＜1@IQ3|-F＜1!F ＜1++ F=F＜1++F＜ 1!---@Q3F＜1@QI3|-F＜1!F＜1@Q3|-F＜1!F＜1 F=
Island1	4	F+F+F+F	F=FFFF-F+F+F-F [-GFF+F+FF+F] FFG =@8G@I8
Island2	4	F+F+F+F	F=F+gF-FF-F-FF+g+FF-gF+FF+F+FF-g-F FF g=@6G@I6
Quartet	4	Fb	A=FBFA+HFA+FB-FA B=FB+FA-FB-JFBFA F=H=-J=+
SnowFk1	12	FR	R=++!FRFU++FU++FU!---@Q3FU|-@IQ3!FRFU! U=!FRFU!|+@Q3FR@IQ3+++!FR--FR--FRFU!-- F=
SnowFk3	12	Fx	x=++F!x!Fy--Fx--Fy|+@iq3FyF!x!++F!y!++ F!y!Fx@q3+++F!y!Fx y=FyF!x!+++@iq3FyF!x!++F!x!++F!y!Fx@q3|+Fx--Fy--FxF!y!++

			F=
Tree1	12	+++FX	X=@.6 [-FX] +FX
Peano2	8	FXY++F++FXY++F	X=XY@Q2-F@IQ2-FXY++F++FXY Y=-@Q2F-@IQ2FXY
Peano3	4	x	x=XFYFX+F+YFXFY-F-XFYFX y=YFXFY-F-XFYFX+F+YFXFY
Plant07	14	Z	z=zFX [+Z] [-Z] x=x [-FFF] [+FFF] FX
Plant08	20	SLFFF	s= [+++Z] [---Z] TSz=+H [-Z] Lh=-Z [+H] L t=TL1= [-FFF] [+FFF] F
Spski1	3	F	F=FXF X=+FXF-FXF-FXF+
Spski2	6	FXF--FF--FF	F=FF x=--FXF++FXF++FXF--
Spski3	3	F-F-F	F=F [-F] F
SpskiSq	4	F+F+F+F	F=FF+F+F+F+FF
Pentag	10	Fx++Fx++Fx++Fx++Fx	F=F [++++@1.618033989F] x= [++++@i1.618033989F@.618033989F!x!@i.618033989F]
QuadKoch	4	F-F-F-F-	F=F+FF-FF-F-F+F+FF-F-F+F+FF+FF-F
Fass1	4	-1	L=LF+RFR+FL-F-LFLFL-FRFR+ R=-LFLF+RFRFR+F+RF-LFL-FR

Fass2	4	-1	L=LFLF+RFR+FLFL-FRF-LFL-FR +F+RF-LFL-FRFRFR+ R=-LFLFLF+RFR+FL-F-LF+RFR+ FLF+RFRF-LFL-FRFR
QGosper	4	-Fr	1=F1F1-Fr-Fr+F1+F1-Fr-FrF1+Fr+F1F1Fr-F 1+Fr+F1F1+Fr-F1Fr-Fr-F1+F1+FrFr- r=+F1F1-Fr-Fr+F1+F1Fr+F1-FrFr-F1-Fr+ F1FrFr-F1-FrF1+F1+Fr-Fr-F1+F1+FrFr F=
Plant01	14	F	F=F [+F] F [-F] F
Plant02	18	F	F=F [+F] F [-F] [F]
Plant04	18	x	X=F [+X] F [-X] +XF=FF
Plant05	14	x	X=F [+X] [-X] FXF=FF
Plant06	16	x	X=F- [[X] +X] +F [+FX] -XF=FF
Plant09	14	y	x=X [-FFF] [+FFF] FXy=YFX [+Y] [-Y]
Plant10	10	F	F=F [+FF] [-FF] F [+FF] [-FF] F
Plant11	12	F	F=F [+F [+F] [-F] F] [-F [+F] [- F] F] F [+F] [-F] F
Curve1	4	F-F-F-F-	F=FF-F-F-F-F-F+F
Curve2	4	F-F-F-F-	F=FF-F+F-F-FF

Curve3	4	F-F-F-F-	F=F-FF--F-F
Curve4	6	yF	x=YF+XF+Yy=XF-YF-X
Leaf1	8	x	a=n, n=o,o=p, p=x b=e, e=h, h=j, j=y x=F [+A(4)] Fy y=F [-B(4)] Fx F=@1.18F@i1.18
Leaf2	8	a	a=F [+x] Fb b=F [-y] Fa x=a, y=b F=@1.36F@i1.36
Bush	16	++++F	F=FF- [-F+F+F] + [+F-F-F]
MyTree	16	++++F	F=FF- [XY] + [XY] X=+FY, Y=-FX
CTGasket	6	--X	X=++FXF++FXF++FXF>1 F=FF
SGasket	4	X	X=+FXF+FXF+FXF+FXF F=FF
DrCurve	4	X	X=X-YF- Y=+FX+Y

Square	4	F+F+F+F	F=FF+F+F+F+FF
KochCurve	6	F	F=F+F--F+F
Penrose1	10	+WF--XF---YF--ZF	W=YF++ZF----XF [-Y F----WF] ++ X=+YF--ZF [---WF--XF] + Y=-WF++XF [+++YF++ZF] - Z=--YF++++WF [+ZF++++XF] --XF F=
CPenrose1	10	+WC02F--XC 04F---YC04F--ZC02F	W=YC04F++ZC02F----XC04F [-YC04F----WC02F] ++ X=+YC04F--ZC02F [---WC02F--XC04F] + Y=-WC02F++XC04F [+++YC04F++ZC02F] - Z=--YC04F++++WC02F+Z [C02F++++XC04F] --XC04F F=
Penrose2	10	++ZF----XF-YF----WF	W=YF++ZF----XF [-YF----WF] ++ X=+YF--ZF [---WF--XF] + Y=-WF++XF [+++YF++ZF] - Z=--YF++++WF [+ZF++++XF] --XF F=
Penrose3	10	[X] ++ [X] ++ [X] ++ [X] ++ [X]	W=YF++ZF----XF [-YF----WF] ++ X=+YF--ZF [---WF--XF] + Y=-WF++XF [+++YF++ZF] - Z=--YF++++WF [+ZF+++...+XF] --XF

			F=
Penrose4	10	[Y] ++ [Y] ++ [Y] ++ [Y] ++ [Y]	W=YF++ZF----XF [-YF----WF] ++ X=+YF--ZF [---WF--XF] + Y=-WF++XF [+++YF++ZF] - Z=--YF++++WF [+ZF++++XF] --XF F=
DPenrose	10	[X] [Y] ++ [X] [Y] ++ [X] [Y] ++ [X] [Y] ++ [X] [Y]	W=YF++ZF----XF [-YF----WF] ++ X=+YF--ZF [---WF--XF] + Y=-WF++XF [+++YF++ZF] - Z=--YF++++WF [+ZF++++XF] --XF F=
Sphinx	6	X	X=+FF-YFF+FF--FFF|X|F--YFFFYFFF|Y=-FF+X FF-FF++FFF|Y|F++XFFFXFFF|F=GG G=GG
PPlexity	10	F++F++F++F++F	F=F++F++F|F-F++F
CTile	24	X+X+X+X+X+X+X+X+ X+X+X+X+X+X+X+X+ X+X+X+X+X+X+X+X	x= [F+F+F+F [---X-Y] +++++F++++++++F-F-F-F] y= [F+F+F+F [---Y] +++++F++++++++F-F-F-F]

§6.4 迭代函数系统

魏尔(C.H.H.Weyl,1885-1955)在《对称》一书中就指出过，“*Turritella duplicata* ”的壳与“鹦鹉螺”的壳可用简单的伸缩变换刻划。不过，在80年代以前，只有少数杰 出人物洞察到复杂的生物形态背后可能隐藏着简单的规则。分形几何学兴起后，通过分形迭代法模拟生物形态发生过程，已成为倍受科学界注目的一件事情。

分形模拟自然形态所涉及的主要学科包括计算机图形学、仿射几何、形式语言等，它们也都因此而突然变得活跃起来。科学家早已可以轻而易举、维妙维肖地生成“人工生物”了。只要拥有一台286以上的微机，经过努力人人都可以尝试造一些生物，一方面模拟已有的物种，另一方面可以任意创造新的物种。你的绘画技巧可能不甚高明，但计算机可以助你一臂之力，只需调系数、参数就可以生成各种不同的生物形态，其精细程度不亚于素描大师的作品 。

迭代函数系统(IFS，简称迭代函数系)方法是美国佐治亚理工学院的巴恩斯利教授首创的。I FS方法的魅力在于它是分形迭代生成的“反问题”，根据拼接定理(collage theorem)，对于一个给定的图形(比如一幅照片)，求得几个生成规则，就可以大幅度压缩信息。

这里采用确定性算法与随机性算法相结合的办法生成植物杆茎或叶片。“确定性”指用以迭代的规则是确定性的，它们由一组仿射变换(如 R_1，R_2，R_3 等等) 构成；“随机性”指迭代过程是不确定的，每一次究竟迭代哪一个规则，即R_i中具体哪一个，不是预先定好的，

而要靠掷骰子的办法来决定。设最终要生成的图形(植物形态图)为 M ，它要满足下述集合方程:

$$M=R\_1 \cup R\_2 \cup \cdots \cup R\_N$$

上式的含义是,随机地从 $R\_i(i=1,\cdots,N)$ 中挑选一个迭代规则迭代一次，然后再随机地在 $R\_i(i=1,\cdots,N)$ 中选一个规则迭代一次，不断重复此过程，最后生成的极限图形 M 就是欲求的植物形态图。

每个迭代规则 R_i 都是一个仿射变换。正交变换保持几何图形的度量性质(向量的夹角，点与点之间的距离，图形的面积等)不变；而仿射变换一般会改变图形中向量的夹角、 点对间的距离、图形的面积等，但仿射变换不改变共线、平行、相交、共线点的顺序、中心对称、二次曲线的次数等。举例来说，原来平行的线段，经仿射变换后仍然是平行的，但长度可能发生了变化，之间的距离也可能改变了。一个图形经仿射变换后面积变小了，则此变换是收缩的，如果变大了则是扩张的，若保持不变则是恒等的。这里只用到收缩性仿射变换，因为极限图形 M 应当是所有迭代 R_i 的吸引子，每个仿射变换是收缩性的才能保证迭代收敛到 M 上。

仿射变换是一种线性变换，在二维平面上进行讨论，二维仿射变换的形式为:

$$R:\quad x' = ax+by+e \qquad y' = cx+dy+f$$

上式中未知数有两个：x和y；系数有四个：a, b, c 和 d；常数有两个：e 和 f。

设平面上一有面积区域为 D，经仿射变换 R 变换后变成 D'，则 D 与 D' 的面积关系为：

$$S(D') = |\det(A)| \cdot S(D)$$

这里 $|\det(A)|$ 表示矩阵 A 的行列式的绝对值，它是小于 1 的正数，其中 A 为系数矩阵。

对于不同的 R_i（i=1，2，…，N），有相应的 A_i（由 a_i，b_i，c_i，d_i 组成），相应的常数 e_i 和 f_i。通常 N取 2，3，4，有时高达 16。

剩下的一个问题是怎样实现掷骰子操作。在计算机上可以很容易地用伪随机数发生器不断生成随机数，用来代替人工抛掷有 N个面的骰子。不失一般性，设 N=4，每次用计算机生成一个随机数 $E\in(0, 100)$，设 $0 < \beta\_1 < \beta\_2 < \beta\_3 < 100$，作如下规定：

若 $0 < E < \beta\_1$，则选择规则 R_1，

若 $\beta\_1 \leqslant E < \beta\_2$，则选择规则 R_2，

若 $\beta\_2 \leqslant E < \beta\_3$，则选择规则 R_3，

若 $\beta\_3 \leqslant E < 100$，则选择规则 R_4.

指定 β_i 的过程，相当于为每一种迭代规则 R_i 指派一个概率 p_i。虽说具体每一次用哪个规则迭代不确定，但从长期行为看，每种规则使用的频率是一定的。 当然这种概率可以随意指定，但是为了使得迭代高

效、迅速，概率的大小应与仿射变换的图 形压缩率成正比。图形压缩率就是前文说的｜det(*A*)｜，对于 *N*=4 的情况， 它有 4 个。

至此可以知道，每一种规则 R_i 都包括 7 个参数(或系数)： *a_i*，*b_i*， *c_i*，*d_i*，*e_i*，*f_i*，*p_i*。要求 *p*_1+ *p*_2+…+*p_N*=1。如果生成某一种植物形态需要 4 种规则，则共有 4×7=28 个参数。只要给定了这些参数，实际上极限图形 *M*就完全确定了。开始迭代时可能还一时看不出 *M*的面貌，但过了一会以后，*M*的轮廓就可以看清楚了， 最后 *M*的精细结构就展示出来了。浙江大学数学系陈刚(1963-)最近还研究了准仿射 变换的一些性质。用准仿射变换也能生成各种分形图形。

图 6.4.1(上左)就是用 IFS 方法生成的图象，此图象的算法由美国佐治亚工学院的巴恩斯利首创，它像不像蕨类植物的叶子?生成此叶子用了 4 个仿射变换 *R*_1，*R*_2， *R*_3 和 *R*_4。7 个参数的取值见表 6.4.2。

表 6.4.2 巴恩斯利蕨的参数表

R	*a*	*b*	*c*	*d*	*e*	*f*	*p*
1	0	0	0	0.16	0	0	0.01
2	0.85	0.04	－0.04	0.85	0	1.6	0.85
3	0.2	－0.26	0.23	0.22	0	1.6	0.07
4	－0.15	0.28	0.26	0.24	0	0.44	0.07

表 6.4.3 树叶 B 的参数表

R	*a*	*b*	*c*	*d*	*e*	*f*	*p*
1	0	0	0	0.5	0	0	0.05
2	0.12	-0.82	0.42	0.42	0	0.2	0.4
3	0.12	0.82	-0.42	0.42	0	0.2	0.4
4	0.1	0	0	0.1	0	0.2	0.15

增减规则 *R_i*，可以改变最终植物 M 的形态。即使不改变迭代规则，采用同样的程序，只改变参数也可以生成完全不同的植物形态。为了使程序具有通用性，可以设计如下“过程”：

```
Procedure AFF(a,b,c,d,e,f,S,T:real);
Var lins:real;
Begin
  lins:=a*S+b*T+e;
  y:=c*S+d*y+f;
  x:=lins;
End;
```

程序中不断调用此过程 AFF，即可完成迭代。这样，每次修改参数，只需改变数组中的数据就可以了。由此得到一个启示：外表极不相同、极复杂的形态，其背后控制其生长的规律可能是相同(或相似)的、简单的。

再推一步，分子水平上控制形态发生的 DNA 遗传编码，也应当是简单的。这一猜想有许多合理之处，直观上想象一下，DNA 分子不可能承载过多的形态控制信息。

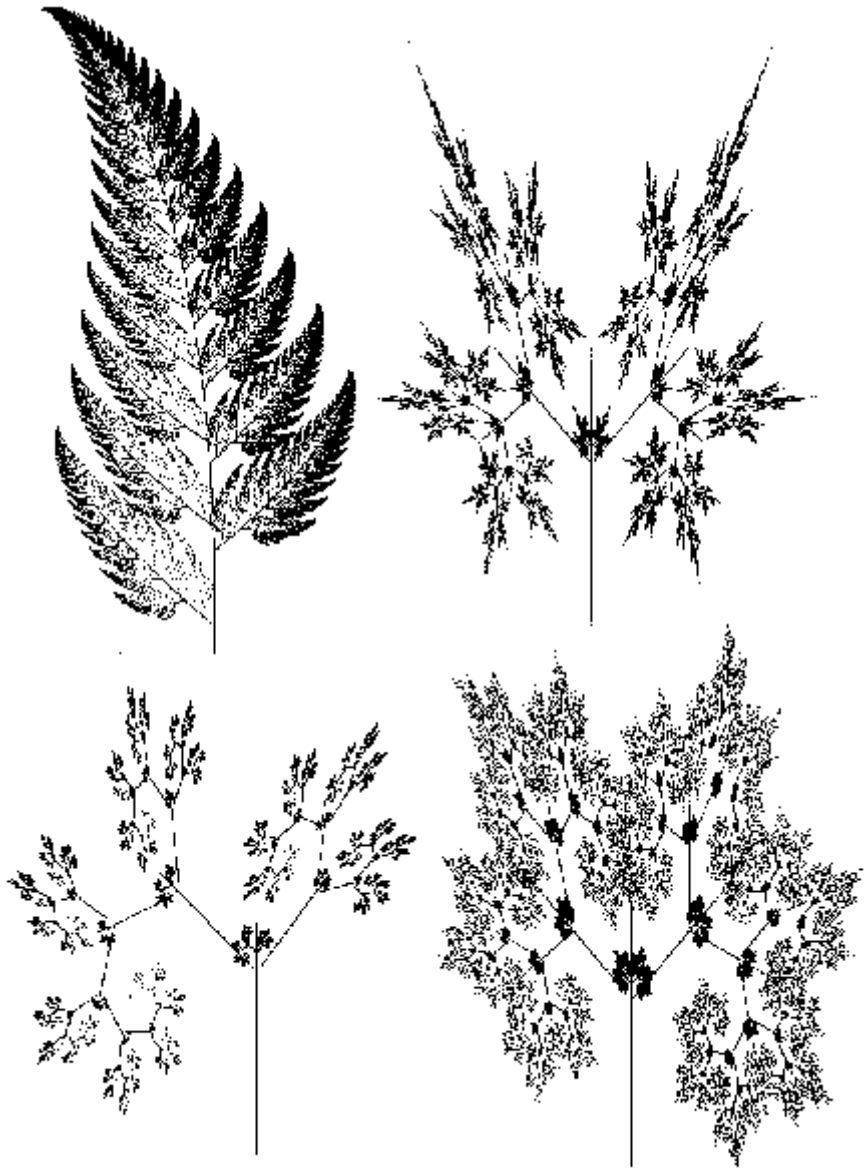


图 6.4.1　用 IFS 方法生成的植物形态

对于表 6.4.2，如果把其中的第二行中表项(*a*, *b*, f)的值由(0.85, 0.04, 1.6)改为(−0.85, 0.09, 2.6)，则会生成蒿草。对于表 6.4.3，若把第二、第三行的表项 c 分别由 0.42 改为 0.2(保留符号)，则会生成

六角枫叶；若把(*a*,*b*)两列的 0.12 和 0.82 都改成 0.42(保留符号)，则会生成树冠。读者也可以自己改变参数，试验着生成各种精美的分形植物形态。

```
{IFS 生成植物形态 PASCAL 程序}
Program FractalPlant28;
Uses Graph,Crt;
var
x,y,E,u:real;
Gd,Gm:integer;
Begin
  Gd:=VGA; {用 VGA 方式 640×480}
  Gm:=VGAHi;
  Initgraph(Gd,Gm,' D:\\PASCAL' ); {初始化图形}
  if GraphResult <> grOK then Halt(1); {若出错则停机}
  x:=0;y:=0; {赋初值}
  Randomize; {初始化随机数发生器}
  Repeat
```

```
    E:=Random(100); {产生一个介于 0 至 100 之间的随机数}
    if E < 1 then
    Begin
      x:=0;y:=0.16 * y; {用 R_1 迭代}
    End;
  if (E >=1) and (E < 86) then
    Begin
      u:=0.85*x+0.04*y;y:=-0.04*x+0.85*y+1.6; {用 R_2 迭
代}
      x:=u;
    End;
  if (E >= 86) and (E < 97) then
    Begin
      u:=0.2*x-0.26*y;y:=0.23*x+0.22*y+1.6; {用 R_3 迭代}
      x:=u;
    End;
  if E >=97 then
```

```
    Begin
      u:=-0.15*x+0.28*y; y:=0.26*x+0.24*y+0.44; {用 R-4 迭代}
      x:=u;
    End;
  PutPixel(Round(50*x)+300,500-Round(50*y),2); {描点，绿色}
  Until KeyPressed; {按任意键退出}
  CloseGraph; {关闭图形方式}
End.
```

IFS 的规则实际上可以自由设计，下面再给出一个稍复杂的例子。令 $\Gamma=(x, y)^T=R^2$，取 S_i，$i=1,2,3$，为下列函数:

$$S_1(\Gamma)=\begin{vmatrix} 0 & 0 \\ 0 & c \end{vmatrix}\Gamma+\begin{vmatrix} 1/2 \\ 0 \end{vmatrix}$$

$$\begin{vmatrix} r\cos\varphi & -r\sin\varphi \\ \end{vmatrix} \quad \begin{vmatrix} 1/2-r/2\cos\varphi \end{vmatrix}$$

S_2(Γ) = | | Γ + |

|

|rsinφ rcosφ | |c-r/2sinφ

|

|qcosψ -rsinψ | |1/2-q/2cosψ |

S_3(Γ)= | | Γ+ |

|

|qsinψ rcosψ | |3/5c-q/2sinψ|

其中 c=0.255, r=0.75, q=0.625, ϕ=-π/8, ψ= π/5, $p_1=p_2=p_3=1/3$.

用 PASCAL 实现这个 IFS，则有如下程序:

```
Program FractalTree9611;
uses Graph,Crt;
var
   x,y,ran,u,aa,bb,mm,nn,c,r,q,phi,psi:real;
Gd,Gm:integer;
```

```
begin
  Gd:=VGA;
  Gm:=VGAHi;
  InitGraph(Gd,Gm,'D:\PASCAL');
  if GraphResult <> grOK then Halt(1);
  x:=0;
  y:=0;
  c:=0.255;
  r:=0.75;
  q:=0.625;
  phi:=-pi/8;
  psi:=pi/5;
  aa:=sin(phi);
  bb:=cos(phi);
  mm:=sin(psi);
  nn:=cos(psi);
```

```
Randomize;
Repeat
  ran:=Random(100);
  if ran < 33 then
     begin{ if p1}
       x:=c*x+2/3; {x=ax+by+e}
       y:=c*y; {y=cx+dy+f}
     end;
  if (ran >33) and (ran < 66) then
     begin
       u:=r*bb*x-r*aa*y+1/2-r/2*bb;
       y:=r*aa*x+r*bb*y+c-r/2*aa;
       x:=u;
     end;
  if ran > 66 then
     begin
```

```
            u:=q*nn*x-r*mm*y+1/2-q/2*nn;
            y:=q*mm*x+r*nn*y+3/5*c-q/2*mm;
            x:=u;
          end;
        PutPixel(round(445*x+50),400-round(445*y),2);
      until KeyPressed;
      CloseGraph;
    end.
```

可以修改函数 $S\_i(\Gamma)$，得到许多意想不到的分形树。在修改迭代函数的过 程中，PutPixel 语句中的放大倍数也应作适当调整。

通过 IFS 方法还可以绘制朱丽亚集 J。设 $z\rightarrow z^2+c$，求出两个逆变换：

$w\_1(z)$=SQRT($z-c$)，$w\_2(z)$ =-SQRT($z-c$)，

取概率(p_1,p_2)=(1/2,1/2)，迭代后生成的吸引子实际上就是各种朱丽亚集!您觉得有些不可思议?是的，细想一下，IFS 方法、L 系统、混沌动力学以及有关 M 集和 J 集的迭代，事实上都是相互关联着的，这种关联体现了非线性科学的内在逻辑。

最后我们给出一个用 IFS 方法生成山脉地形的小程序 Mount.PAS:

```
{Mount.PAS}

Program IFSMountain;

uses

Graph,Crt;

var

Gd,Gm,ErrorCode : integer;

x,y: real;

k,MaxY:integer;

i:longint;

d: array[1..4,1..6] of real;

begin

Gd:= Detect;

InitGraph(Gd,Gm,'d:\pascal');

ErrorCode := GraphResult;

if ErrorCode <> grOk then exit;

MaxY := GetMaxY;
```

```
d [ 1, 1 ] : =0. 5;  d [ 1, 2 ] : =0;  d [ 1, 3 ] : =0;

d [ 1, 4 ] : =0. 5;  d [ 1, 5 ] : =0; d [ 1, 6 ] : =0;

d [ 2, 1 ] : =0. 5;   d [ 2, 2 ] : =0;   d [ 2, 3 ] : =0;

d [ 2, 4 ] : =0. 5;   d [ 2, 5 ] : =2;  d [ 2, 6 ] : =0;

d [ 3, 1 ] : =-0. 4;   d [ 3, 2 ] : =0;   d [ 3, 3 ] : =1;

d [ 3, 4 ] : =0. 4;   d [ 3, 5 ] : =0;  d [ 3, 6 ] : =1;

d [ 4, 1 ] : =-0. 5;   d [ 4, 2 ] : =0;   d [ 4, 3 ] : =0;

d [ 4, 4 ] : =0. 5;   d [ 4, 5 ] : =2;  d [ 4, 6 ] : =1;

randomize;

x : = 0;  y : = 0;

repeat

i: =i+1;

k : = random(4) + 1;

x : = d [ k, 1 ] *x + d [ k, 2 ] *y + d [ k, 5 ] ;

y : = d [ k, 3 ] *x + d [ k, 4 ] *y + d [ k, 6 ] ;

if i > 10 then
```

```
putpixel(round(MaxY*x/2),MaxY-round(MaxY*y/2),15)
until keypressed;
closegraph;
end.
```

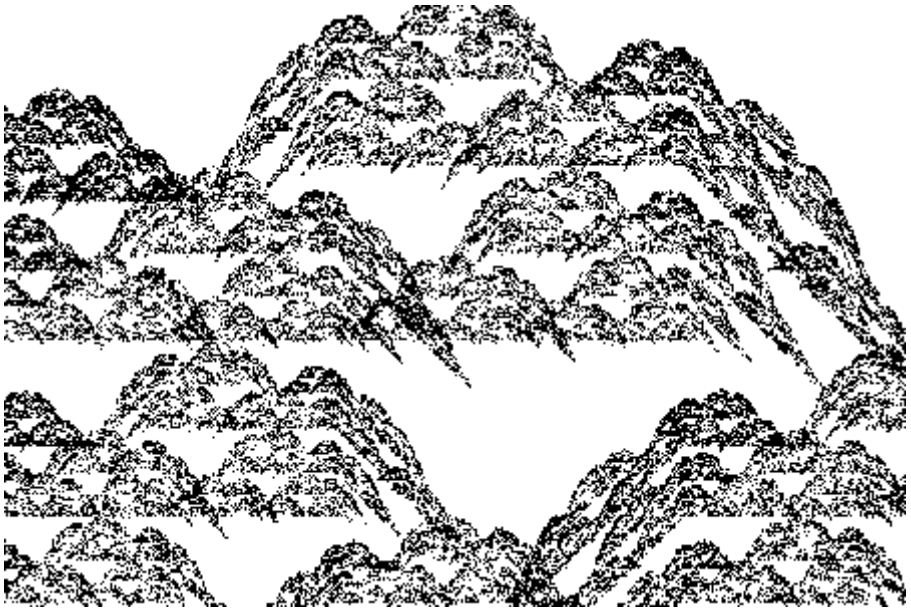


图 6.4.4 用 IFS 方法生成的地貌

§6.5 扩散置限凝聚模型

北方的朋友冬天都见过玻璃上的霜花、树上的雾凇，学地质的都见过树枝状自然铜、自然银，吃过中国传统食品松花蛋的都见过美丽的“松花”，也许你还从电视中看到过美丽的珊瑚树。可是你知道它们有什么联系吗?其形成机理是什么?这个问题现在已经能够清楚地说明了 。

1981 年维腾和桑德在《物理评论快报》上发表文章，提出了扩散置限凝聚(diffusion-limited aggregation)模型,简称 DLA 模型，他们在计算机上用随机扩散方法成功地模拟了复杂的在电解 ZnSO_4 试验中也实际做出了 DLA 模型的结果，而且发现了屏蔽效应。

DLA 模型的思想是很简单的,假设一平面方形点阵(也可以是别的形状)的中央先放入一个静止粒子，在区域边界随机释放一个新粒子，粒子做无规行走，如果碰到中央的粒子则凝聚不动；如果再次碰到外边界则不再考虑它，这时在区域中再产生一个新粒子，同样做无规行走，碰到中间已存在的粒子则凝聚，等等。我们释放 1000 个、25000 个粒子看看会有什么现象。用计算机很容易模拟上述过程，模拟结果发现能够生成各种各样的分形结构，它们很像自然存在的树枝状物体。

图 6.5.1 DLA 模型生成的分形图形

用 DLA 方法得到的凝聚集团具有统计意义上的自相似性，它的密度随集团大小的增加按幂指数规律减小。用 R 表示 DLA 集团的特征长度，则集团的数密度 $\rho(R)$ 为

$\rho(R)=N(R)/R\hat{}d\sim R\hat{}(D-d)$，

其中 N 代表粒子个数，D 为分形维数，对于平面区域 $d=2$，对于三维空间区域 $d=3$。大量计算统计结果显示，当 $d=2$ 时，$D=1.66\pm0.02$；当 $d=3$ 时，$D=2.50\pm0.02$。开始时人们猜测 D 只与 d 有关，而与点阵结构无关，后来证明不是这样，D 与点阵形状有关，并且粒子数越多，差别越大。

DLA 模型提出后人们马上试图推广它，首先考虑让中央的粒子也运动起来，或者干脆一次放入 N 个粒子，使它们按一定的规律运动。1983 年麦金(P.Meakin)给出如下规则：1)二维方形点阵中 N 个粒子均可在点阵上做无规行走，相碰后就结成集团；2)集团整体做无规运动，碰撞后凝聚成更大的集团，不断进行下去；3)对于不同大小的集团指定不同的迁移概率。用这种办法能更好地模拟烟灰、胶体、云尘的聚合过程。

我们这里考虑比标准 DLA 模型更具一般性的过程(根据日本学者 M.Nakagawa 和 K.Kobayashi 的文章“具有局部粒子漂移的 DLA”)。在方形点阵上考虑问题，以 (x,y) 表示粒子当前的坐标，以 Δx 和 Δy 分别表示粒子当前沿 x 和 y 方向的相对位 移分量，f_x 和 f_y 是涨落力，d_x 和 d_y 是漂移力。α 是漂移参数，表示 (f_x,f_y) 和 (d_x,d_y) 之间的比率关系。β 的值等于+1 或者-1，分别表示“吸引”和“排斥”作用。f_x、f_y、d_x、d_y 的取值范围都是(-1/2,1/2)，α 的取值是(0,1)。我们可以将 Δx 和 Δy 的表达式写成

$\Delta x=\mathrm{sgn}[(1-\alpha)f\_x+\beta\ \alpha d\_x]$，

$\Delta y=\mathrm{sgn}[(1-\alpha)f\_y+\beta\ \alpha d\_y]$，

其中 sgn(*x*)表示一种函数，其定义为 sgn(*x*＜0)=−1, sgn(0)=0, sgn（*x*＞0)=+1。上述涨落力 *f*_*x*, *f*_*y* 可由两个独立的伪随机实数产生。若 α=0，则此模型还原为标准的 DLA 模型。漂移变化由下述公式定义:

$$d\_x=\{\sum\_(i,j)W\_(xij)\ \delta(i+x,j+y)\}/\{2\sum\_(i,j)W\_(ij)\ \delta(i+x,j+y)\}$$

$$d\_y=\{\sum\_(i,j)W\_(yij)\ \delta(i+x,j+y)\}/\{2\sum\_(i,j)W\_(ij)\ \delta(i+x,j+y)\},$$

其中 $\delta(p,q)$是局部密度函数,当某格点上有粒子时 δ=1,否则 δ=0。三个矩阵的定义为

$$W\_(xij)=\mathrm{sgn}(i)\,W\_(ij),$$

$$W\_(yij)=\mathrm{sgn}(j)\,W\_(ij),$$

$$W\_(ij)=1,\text{在窗口之内};\ =0,\text{其他}$$

窗口 $W_(ij)$的大小表征作用程的大小。窗口 $W_(ij)$的面积增加对应于长程相互作用，面积减小对应于短程相互作用。如果只对运动粒子 而言 $W_(ij)$为 1，其他情况下为 0，则此模型又还原为标准的 DLA 模型 。

以上说的是从中心的一个“点”开始生长的 DLA 模型。也可以考虑从一根“线”(好比土地) 开始一齐生长，当然也可以考虑从洞穴的不规则“内壁”向内生长。这样一来，DLA 模型就 能模拟许多自然界的生长过程。

一个想法：利用 DLA 方法结合化学分析，研究中国传统食品松花蛋中“松花”的成因，再有意识地淹制生长有各种颜色松花的优质(无毒)松花蛋。经初步观察，松花蛋上的松花大致有两种类型，一种“花根”贴近蛋壳表皮，另一种花根贴近蛋黄外表皮。两种松花由蛋清的两个表皮向内生长，其他地方绝无松花，这表明松花是金属离子在蛋白胶体中通过膜渗透、扩散、聚集而成的。有关资料表明，松花生成于碱性环境，松花富含镁元素 。[汤炯吾，《巧制松花蛋和盐蛋》，四川科学技术出版社 1992 年。另有如下文献：刘仪初 、李树青、罗烈武，松花皮蛋形成机理探讨，《商业科技》，1984 年第 8 期；李树青、梁丹 ，松花蛋的“松花”晶体的分离与结构分析，《商业科技》，1988 年第 1 期；李树青、王庆 玉、谢盛良，溏心皮蛋蛋白胶体中松花晶体的多少与其镁离子含量的相关性，《食品科学》 ，1991 年第 10 期。]如果研究清楚松花的化 学成分、生成理化条件，就可以加入无毒(甚至有利于健康)的微量元素，人为控制松花的生长。

§7.1 复数运算与点列迭代

7.1.1 复数四则运算

迪万内在《混沌动力系统引论》中说：“复解析函数动力学是一个比较专门的课题。……20 世纪 20 年代，这一领域在数学家 Fatou 和 Julia 的指导下曾经极度繁荣,……但在 70 年代的新繁荣时期以前有一段休眠期。后来，主要由 Mandelbrot 的美丽动人的计算机作图和道阿

弟、哈伯德和沙利文(D.Sullivan)等人引人入胜的数学工作，又一次把人们的注意力引向复平面丰富多彩的初等映射的动力学行为。”

对于我们而言，“解析函数”、“复变函数”等概念都可暂且不管，知道在复平面上对简单的多项式进行迭代能够生成美妙的分形图形，就足够了。在计算机十分普及以前，要做这种练习几乎是不可能的。当年朱丽亚和法图是靠抽象的数学思维，艰难地研究复迭代的。进入80年代，计算机产业大发展，正巧分形几何学又到来了，一切条件都已成熟。芒德勃罗在干柴上点了一把火，复迭代复兴了，这把火一直烧到现在，而且火势仍不减当年。

复数迭代比实迭代稍稍麻烦一些，进行复迭代需要知道复数的四则运算。我们先复习一下这些以前学过的内容。

任意复数可以写成三种形式:

1) 代数形式: $z=x+yi$;

2) 三角形式: $z=r\ (\cos\theta+i\ \sin\theta)$;

3) 指数形式: $z=r\ e^{i\theta}$。

其中 i=SQRT(-1)是虚数单位，x=Re(z)是复数的实部，y=Im(z)是复数的虚部，r=|z|=|$x+y$i|=SQRT(x^2+y^2)叫做复数的模，θ 叫做复数的辐角。在复平面上，实部一般用横坐标表示， 虚部一般用纵坐标表示。

设 z_1，z_2，z_3都是复数，复数基本运算如下:

恒等式: $e^{i\theta}=\cos\theta+i\sin\theta$,

加法: $Z_1+Z_2=(X_1+X_2)+(y_1+y_2)i$,

减法: $Z_1-Z_2=(X_1-X_2)+(y_1-y_2)i$,

乘法: $Z_1 . Z_2=(X_1X_2-y_1y_2)+(X_1y_2+X_2y_1)i$,

除法: $Z_1/Z_2=(X_1X_2+y_1y_2)/(X_2X_2+y_2y_2)+(X_2y_1-X_1y_2)/(X_2X_2+y_2y_2)i$,

余弦: $\cos(x+iy)=\cos x\cosh y-i\sin x\sinh y$,

正弦: $\sin(x+iy)=\sin x\cosh y+i\cos x\sinh y$,

指数: $e^{x+iy}=e^x\cos y+ie^x\sin y$,

其中

双曲余弦: $\cosh x=(e^x+e^{-x})/2$,

双曲正弦: $\sinh x=(e^x-e^{-x})/2$.

在复迭代程序中,这些基本运算经常用到,最好将它们都编制成函数(过程),存到一个单元中去,使用时可以把它们当成“标准函数(过程)”调用。设单元为 COMPLEX.TPU, COMPLEX.PAS 的内容如下:

```
unit complex; {PASCAL 单元: 复数四则运算}

interface

procedure add (x1, y1, x2, y2 : real; var x3, y3 : real);
{复数加法}
```

```
procedure sub (x1, y1, x2, y2 : real; var x3, y3 : real);
{复数减法}
procedure mult(x1, y1, x2, y2 : real; var x3, y3 : real);
{复数乘法}
procedure cdiv(x1, y1, x2, y2  : real; var x3, y3 : real);
{复数除法}
function cosh(x : real) : real; {双曲余弦函数}
function sinh(x : real) : real; {双曲正弦函数}
procedure csin(x,y : real; var x1, y1 :real);{复数正弦}
procedure ccos(x,y : real; var x1, y1 :real);{复数余弦}
procedure cexp(x,y : real; var x1, y1 :real);{复数指数}
implementation
procedure add (x1, y1, x2, y2 : real; var x3, y3 : real);
{复数加法}
     {计算 z_3 = z_1 + z_2,
      其中:z_1 = x_1 + iy_1,z_2 = x_2 + iy_2,z_3 = x_3 + iy_3}
begin
     x3 := x1+x2;y3 := y1+y2
end;
```

```
procedure sub (x1, y1, x2, y2 : real; var x3, y3 : real);
{复数减法}
{计算 z_3 = z_1 - z_2}
begin
     x3 := x1-x2;y3 := y1-y2
end;

procedure mult(x1, y1, x2, y2 : real; var x3, y3 : real);
{复数乘法}
{计算 z_3 = z_1 * z_2}
begin
     x3 := x1*x2-y1*y2;y3 := y1*x2 + x1*y2
end;

procedure cdiv(x1, y1, x2, y2 : real; var x3, y3 : real);
{复数除法}
{计算 z_3 = z_1 / z_2}
```

```
var
denom : real;                          {分母}
begin
     denom := x2*x2 + y2*y2;
     x3 := (x1*x2 + y1*y2)/ denom;     {实部}
     y3 := (x2*y1 - x1*y2)/ denom      {虚部}
end;



function cosh(x: real): real;        {计算双曲余弦 cosh(x)}
begin
     cosh := (exp(x) + exp(-x))/2.0
end;

function sinh(x: real): real;        {计算双曲正弦 sinh(x)}
begin
     sinh := (exp(x) - exp(-x))/2.0
end;
```

```
procedure ccos(x, y : real; var x1, y1 : real);
{计算复数余弦 z = cos(x + iy)，其中:x1 是 z 的实部，y1 是 z 的虚部}
begin
    x1 := cos(x)*cosh(y);y1 := -sin(x)*sinh(y)
end;

procedure csin(x, y : real; var x1, y1 : real);
{计算复数正弦 z =sin(x + iy)，其中:x_1 是 z 的实部，y_1 是 z 的虚部}
begin
    x1 := sin(x)*cosh(y);y1 := cos(x)*sinh(y)
end;

procedure cexp(x, y : real; var x1, y1 : real);
{计算复数指数 z = e^(x + iy)，其中: x_1 是 z 的实部，y_1 是 z 的虚部}
```

```
begin
    x1 := exp(x)*cos(y); y1 := exp(x)*sin(y)
end;
end.   {复数四则运算单元结束}
```

7.1.2 复平面上的点列

我们都知道，任一个复数 z=x+yi 可以对应着 xy 平面上的点(x, y)。复数数列$\{z_n\}$如果定义为 $z_n=x_n+y_ni$ ，则 z_n表示了点 $p_n(x_n, y_n)$，从而可以说$\{z_n\}$在 xy 平面上生成了点列$\{p_n\}$。

假定这个复数数列由初值z_0和迭代公式

$$z_n=f(z_{n-1}),\ n=1, 2, 3, \ldots$$

确定，即从 n=1 起顺序代入数值，则对任意 n，均可得到对应的z_n。

一般来说，当 n 趋于无穷时，点列$\{z_n\}$的趋势可以分成以下四种情况:

1. 收敛到一点;
2. 有限个点作周期性振动;
3. 某区域作不规则的运动(无秩序);
4. 发散。

一个复数点列如何运动取决于迭代公式 f 和初值z_0。

下面来看一个简单的复数数列 $z_n=z_{n-1}^2+z_c$，其中 z_c 为复数常量，现在我们先编写一个程序来描画上述点列。设 $z_c=x_c+y_ci$，$z_0=x_0+y_0i$，复数平面上描画区域由外部变量 x_min，x_max，y_min，y_max 指定。设

x_min<=x<=x_max，y_min<=y<=y_max

区域以及画面中央(320，200)为原点，在 x 方向+(-)dx 点，y 方向+(-)dy 点的范围内描画点列，其中 dx，dy 也是外部变量。

Main 主函数中用数组 z[ ],y[ ]表示复数数列的实部和虚部，参加计算的项目个数用符号常量 k1 定义，从代入 x[0]=x0,y[0]=y0 起直到 x[k1],y[k1] 止共计算 k1+1 个项目，但计算中若平方和 x[k]*x[k]+y[k]*y[k]>=4 时则停止计算转向 BR:，这时数列是发散的。

若计算一直到最后未被中断的话，那么这个数列的趋势就是收敛、振动或无秩序之一。下面对此进行分类：振动的场合可以求出其周期点的数目，符号常量 BOX 用于这时的周期点，以最后一项为中心作出纵横+-BOX 的小正方形，从最后数起第 k 个项目若进入其中的话，则周期点的数目判断为 k。

点列的描画在收敛时用 1 号蓝色，在振动时则用到其周期点数目为止的颜色，如周期点数目为 4 时，可得到 4 个收敛的部分数列，分别用 1 号蓝色，2 号绿色，3 号浅蓝色，4 号红色来分类。调色板 8 号深灰色可用函数 setpalette 预先变更为 14 号淡黄色，因为深灰色不易与背景黑色区别。

无秩序和发散场合用 15 号白色描画，对振动场合的周期点数目达到

14 以上时也用白色描画，另外，收敛、振动、无秩序、发散无任哪种场合。其初值均用黄色正方形框住，便于和其它点相区别。

Main 函数定义的 k_max 为项目数，prd 为代入周期点数目的变量，发散以外时，k_max=k1。在程序的最后显示出 text 画面，描画区域和周期点的坐标等。

颜色和颜色号对应关系如下表:

0－黑，1－蓝，2－绿，3－浅蓝，4－红，5－紫，6－黄，7－浅灰，8－深灰，9－浅黄，10－浅绿，11－浅水蓝，12－淡红，13－淡紫，14－淡黄，15－白

注意，当yc=0.0 时，复数zc实际上是一个实数。现在我们就以x_min=-2.0, x_max=2.0, y_min=-0.2, y_max=0.2 为描画区域，以xc=-1.35, yc=0.0, x0=0.2, y0=0.05 为复数常量和初值在画面上表示出点列的动向，其它外部变量和符号常量是dx=224, dy=112, KL=100, BOX=0.01, x轴上会出现 4 个周期点，如图 7.1.1 所示，其源程序见p7-1-1.htm.

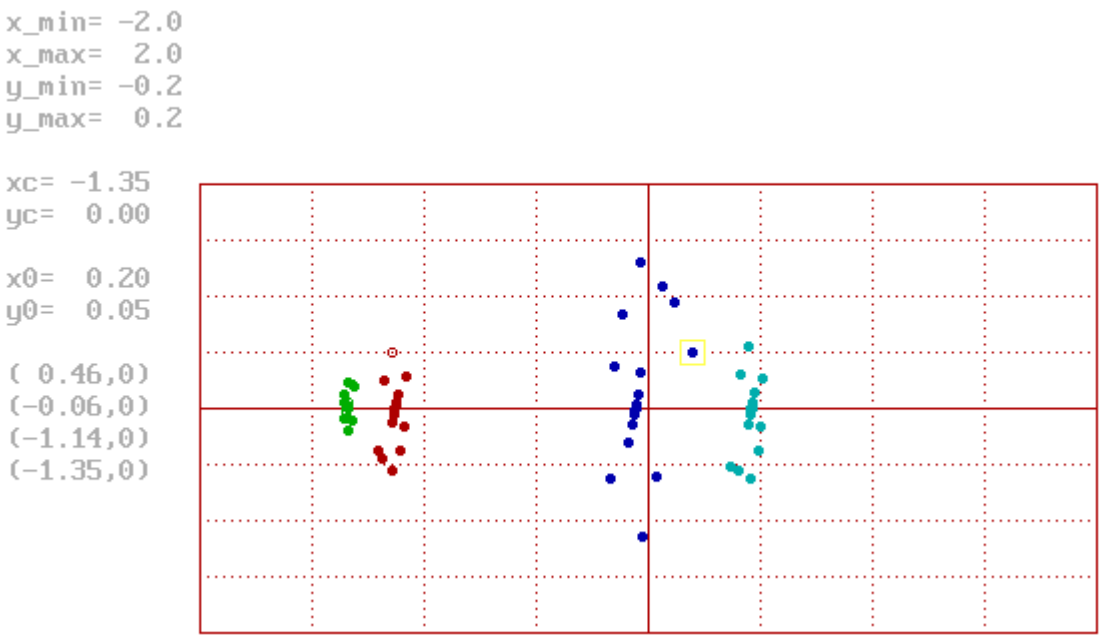


图 7.1.1　复数数列 zn=zn-1^2+zc 中 4 周期的振动

zc=-1.35, z0=0.2+0.05I

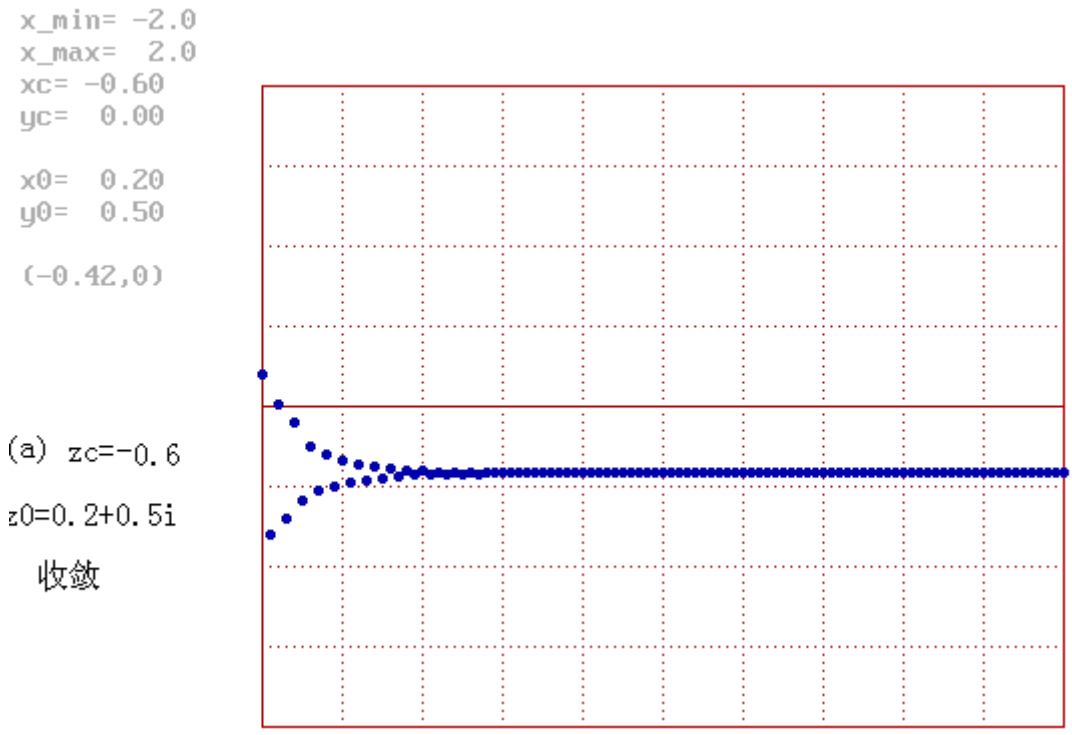
描画区域 x=-2.0~2.0, y=-0.2~0.2

4 个周期点：(0.46, 0), (-0.06, 0), (-1.14, 0), (-1.35, 0)

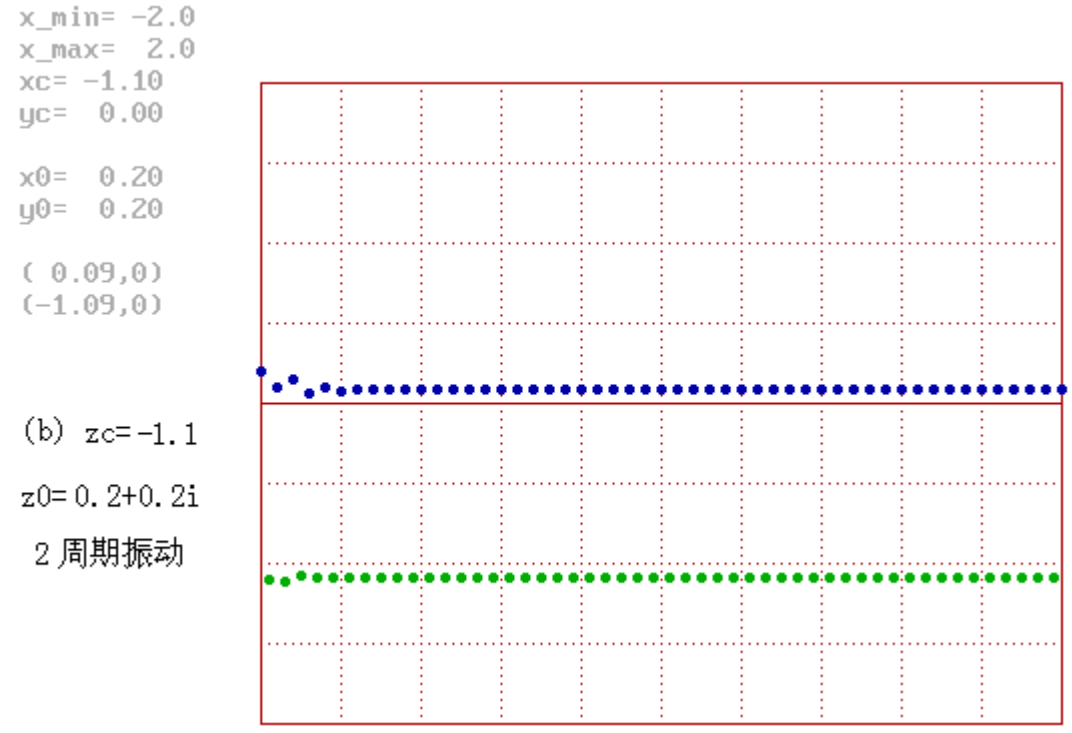
7.1.3 预料不到的变动

代入若干适当的值就可以知道，当 zc 为实数时，复数系列 zn=zn-1^2+zc 不任收敛或振动，其收敛点和周期点都一定在 X 轴上。将横坐标为 X，纵坐标为 zn 的点的横坐标记为 xn，并用来描画图形，则可从其它角度来看复数点列的走向。

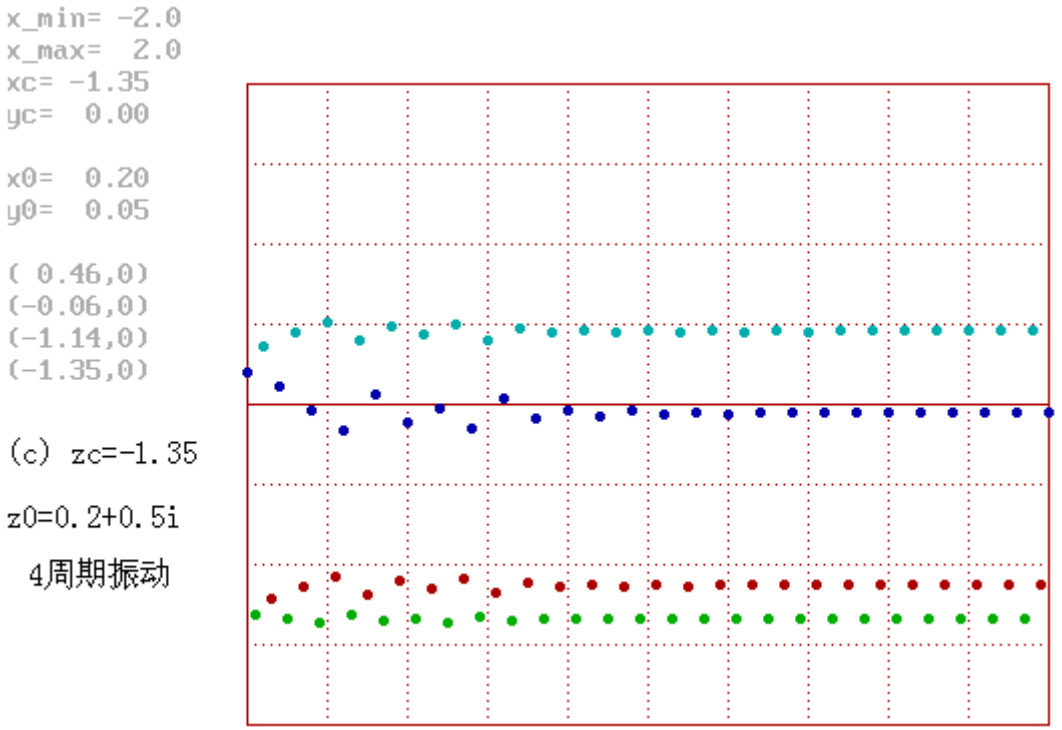
下面将第 1 节的程序改动一下，dx 变成 d，增加表示横轴上单位长度的符号常量 UX，程序为：



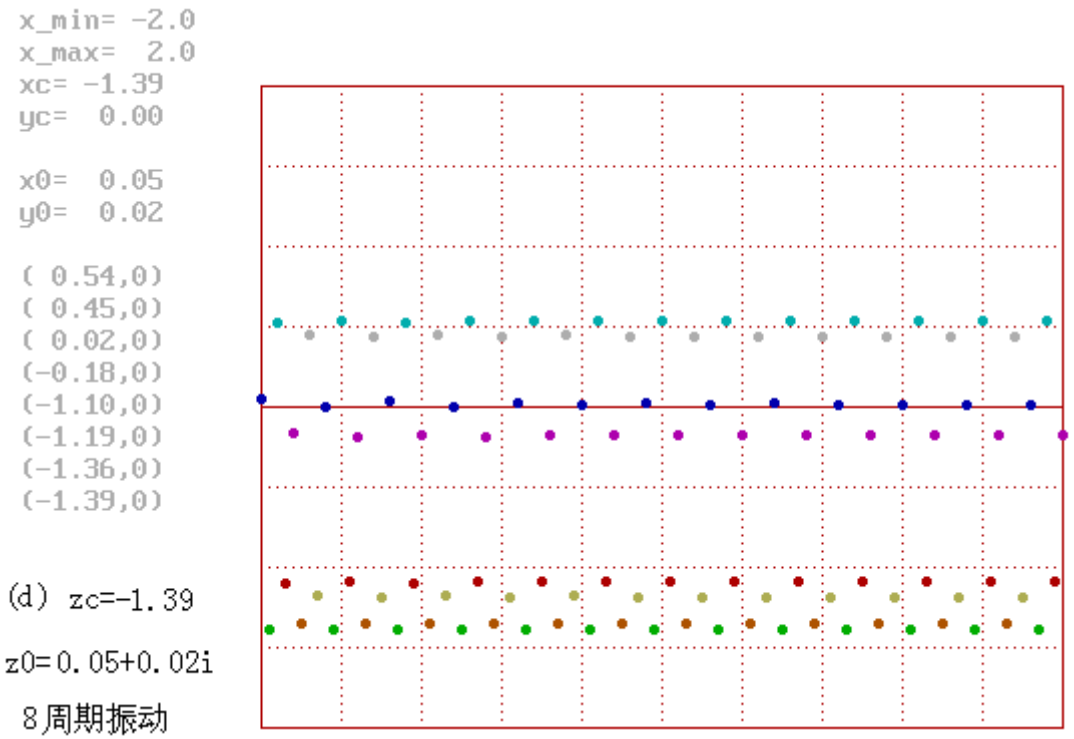
(a)--zc=-0.6, z0=0.2+0.5I, 数列收敛，收敛点为(0.42,0);



(b)--zc=-1.1, z0=0.2+0.2I, 2 周期振动，周期点为(0.09,0), (-1.09,0);

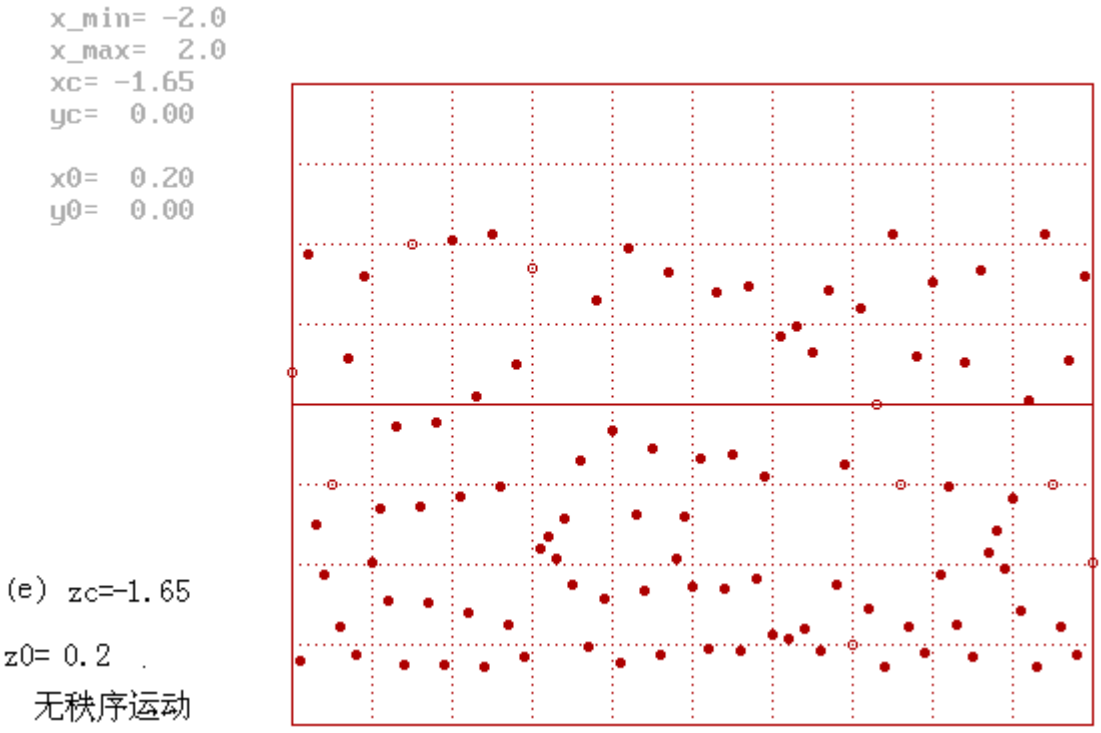


(c)--zc=-1.35, z0=0.2+0.5I, 4 周期振动，周期点为(0.46,0), (-0.06,0),(-1.14,0),(-1.35,0);

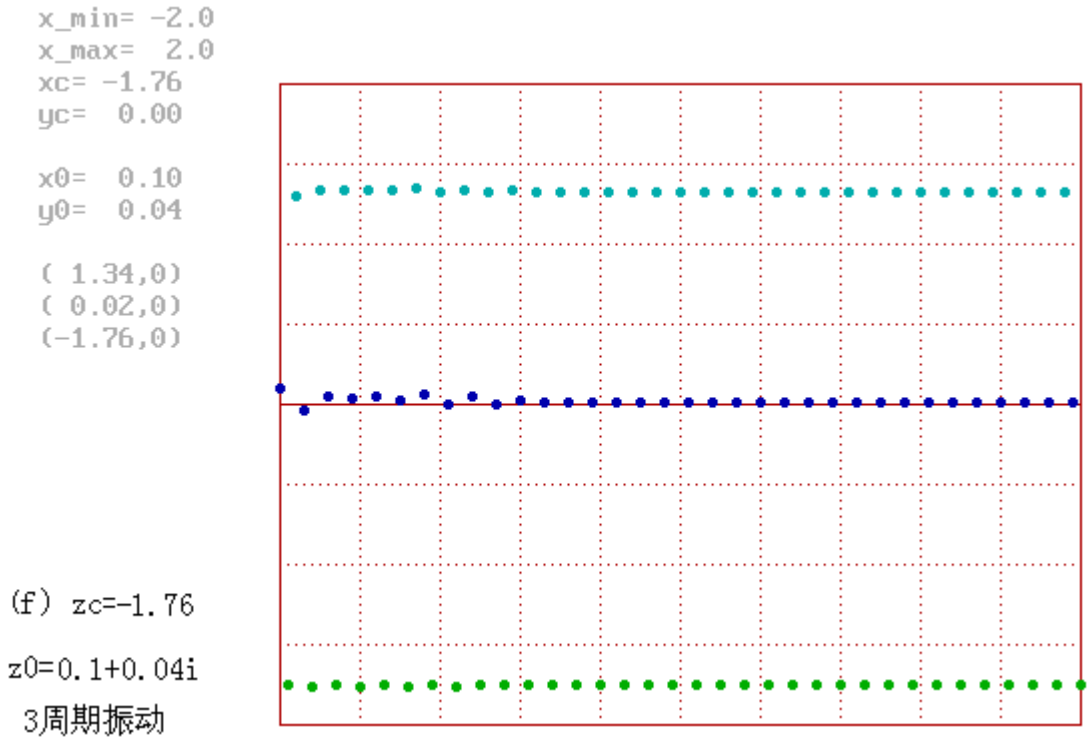


(d)--zc=-1.39, z0=0.05+0.02I, 8 周期振动，周期点为(0.54,0),

(0.45,0),(0.02,0),(-0.18,0), (-1.10,0), (-1.19,0), (-1.36,0), (-1.39,0)



(e)--zc=-1.65, z0=0.2, 无秩序运动



(b)--zc=-1.76, z0=0.1+0.04I, 3 周期振动，周期点为(1.34,0), (0.02,0),

(-1.76,0)

图 7.1.2 复数系列中 x 坐标的变动，横方向 n=1~100，纵方向 x=-2.0~2.0

§7.2 Julia集合

我们知道，复数数列 zn=zn-1^2+zc 生成的点列，因为复数常量 zc 和初值 z0 的不同取法，可以有 4 种动向，即收敛、振动、无秩序和发散，其中振动的场合，还可以根据周期点的数目再细分，虽然前面的讨论中我们取周期点数目为 2、3、4、8，实际上可以取任意值。

对于某个常量 zc,使得数列{zn}不发散的初值 z0 的集合称为 Julia 集合,不发散即意味着必为收敛、振动、无秩序之一。

我们作如下规定：固定某 zc 后，定下复数平面上某个区域，对于该区域内的所有点，看看将这些点作为初值的点列的动向，对于各个初值，根据动向的不同分别用下列颜色表示：

- 收敛－－1 号蓝色
- n 周期振动(2<=n<=13)——n 号颜色，但 8 周期时用 14 号淡黄色
- 14 以上周期振动－－15 号白色
- 无秩序－－15 号白色
- 发散－－不画

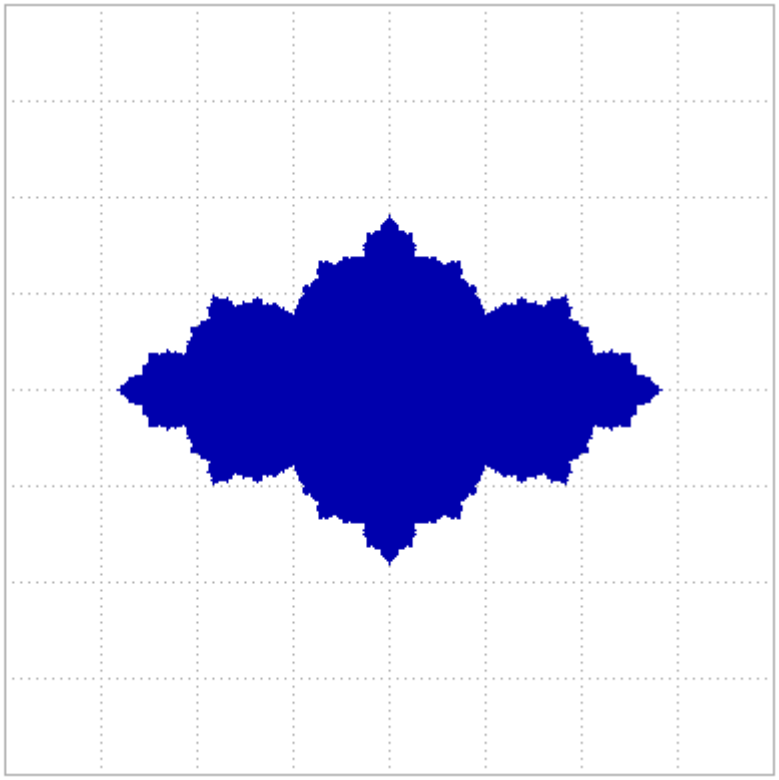
这样，用所赋予的颜色描画点时，有色部分即为 Julia 集合。这时，我们可以看到，由于要对每点进行处理和描画，这是相当费时间的。

我们设定用于处理较长描画时间的规定于符号常量 STEP，在主函数 main 中，对 STEP 设定某个值，则上下左右均按此间隔在格子上描画这些点，当然，当 STEP 为 1 以外时，只能描画不完全的图，但可缩短描画时间，一般用于测试。首先设定 STEP 为 4～10 左右看看大致情形，然后直至为 1 描画出完全图形。

指定数列项数的符号常量 KL 也是影响描画时间的因素，该值越大时间越长，但能描出精密图形。下面我们规定 KL=200。

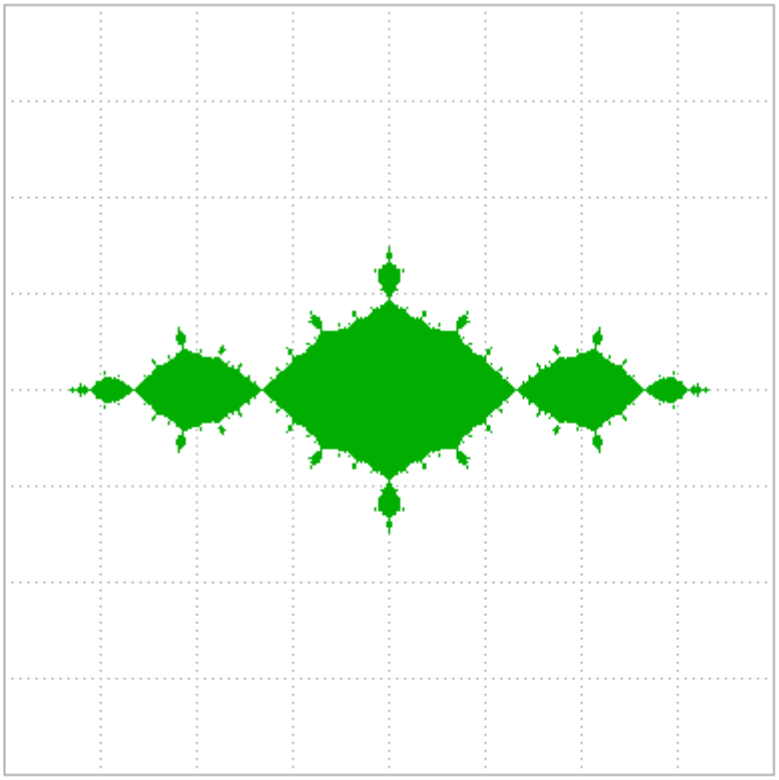
我们先看复数常量zc为实数的情形，描画的Julia集合，如图 7.2.1 所示，其源程序如下：

```
x_min= -2.0
x_max=  2.0
y_min= -2.0
y_max=  2.0

xc= -0.60
yc=  0.00
```



(a)　　zc=-0.6，收敛(蓝色)

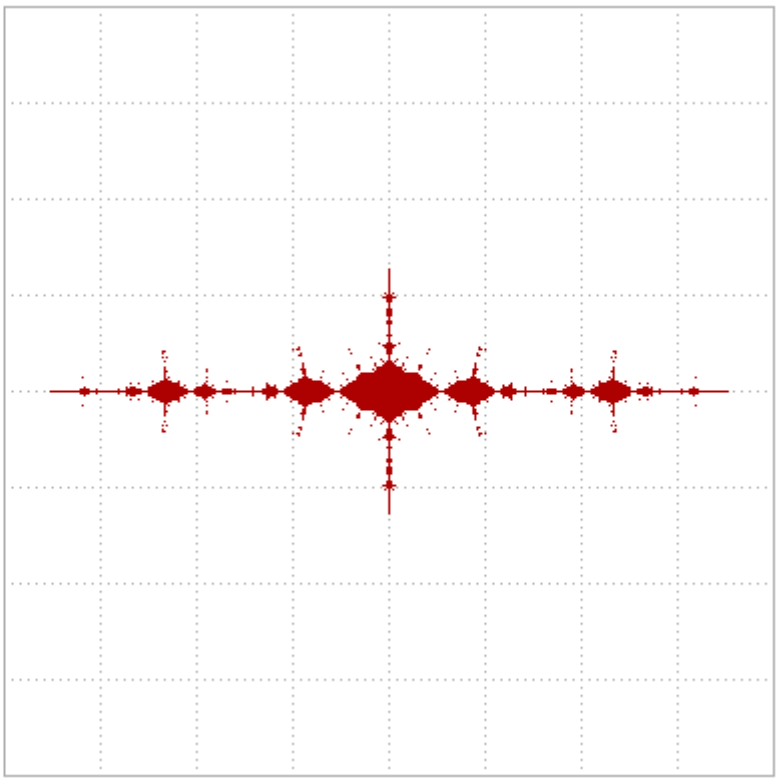
```
x_min= -2.0
x_max=  2.0
y_min= -2.0
y_max=  2.0

xc= -1.10
yc=  0.00
```



(b)　　zc=-1.1，2 周期振动(绿色)

x_min= -2.0
x_max= 2.0
y_min= -2.0
y_max= 2.0

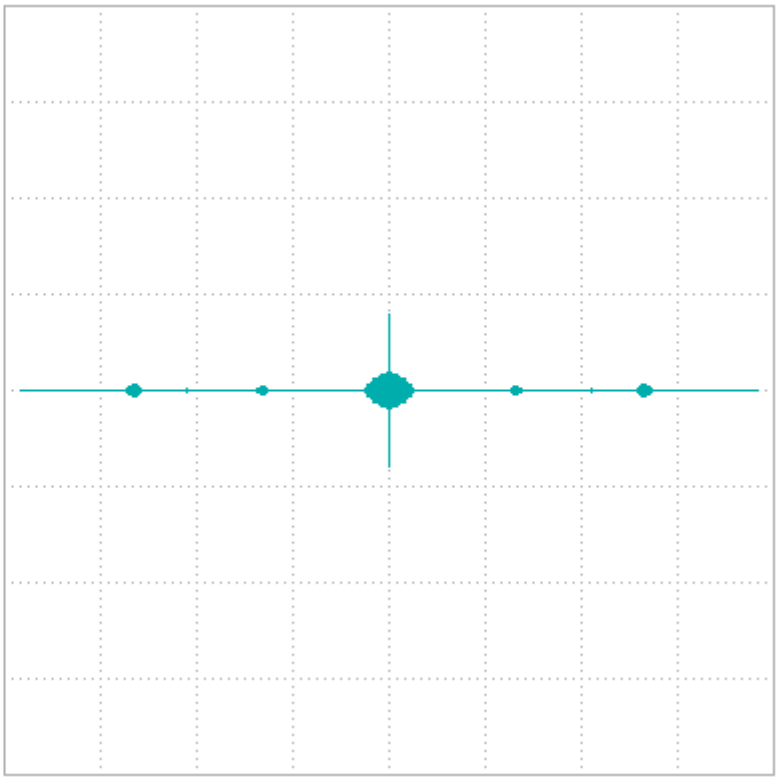
xc= -1.35
yc= 0.00



(c)　　zc=-1.35，4 周期振动(红色)

x_min= -2.0
x_max= 2.0
y_min= -2.0
y_max= 2.0

xc= -1.76
yc= 0.00



(d)　　zc=-1.76，3 周期振动(浅蓝色)

图 7.2.1　zn=zn-1^2+zc(zc 为实数)的 Julia 集合，描画区域:

x=-2.0¯2.0，y=-2.0¯2.0

从上图可以看出，当 zc 为实数时，Julia 集合是关于 X 轴和 Y 轴对称的图形，但是当 zc 为复数时，这种对称性就不存在了，而只有绕原点旋转 180 度的对称性，即所谓点对称图形。

如果要描画 zc 为复数的 Julia 集合，对前面的程序要作小的改动:首先是点描画函数要改为

```
void plot(int col)
{  putpixel(sx,-sy,col); putpixel(-sx,sy,col);}
```

在主函数 main 中，第 4 个 for 循环变量 sy 的初值要改成-dy:

```
for(sy=-dy; sy<=dy; sy+=STEP)
```

这样只要对外部变量 xc,yc 代入适当的值后执行该程序就可以描画 Julia 集合。

要注意的是，Julia 集合只用一种颜色着色，这意味着，对于某个 zc，如描画 Julia 集合的话，那么该集合上所有的点以及以这些点为初值的数列 zn=zn-1^2+zc 的趋势是同一类型的，若收敛则都收敛，若 2 周期振动，则都是 2 周期振动。在一个集合上不存在收敛和 2 周期振动混合的情形，若按点列的走向对 Julia 集合进行分类的话，则其类型是可以唯一决定的。

代表 Julia 集合的特征的就是其形状，整体形状是关于原点的点对称，另外还具有 2 个主要中心，若干叶子从该中心向周围伸长，其中朝着原点的最大叶子与从另一中心伸长出的叶子相连，呈现出共有一张叶子 2 株玫瑰花结的形状，各叶的顶端又形成次中心，其周围也伸长着叶子，从该中心伸长出的叶子的配置与围绕主要中心的叶子的配置是同一结构，如此反复形成分形结构，照直伸长的叶子的 Julia 集合就如水面上的浮草或培养容器上的繁殖的酵母一样(如图 7.2.2(c) (e) (g))。

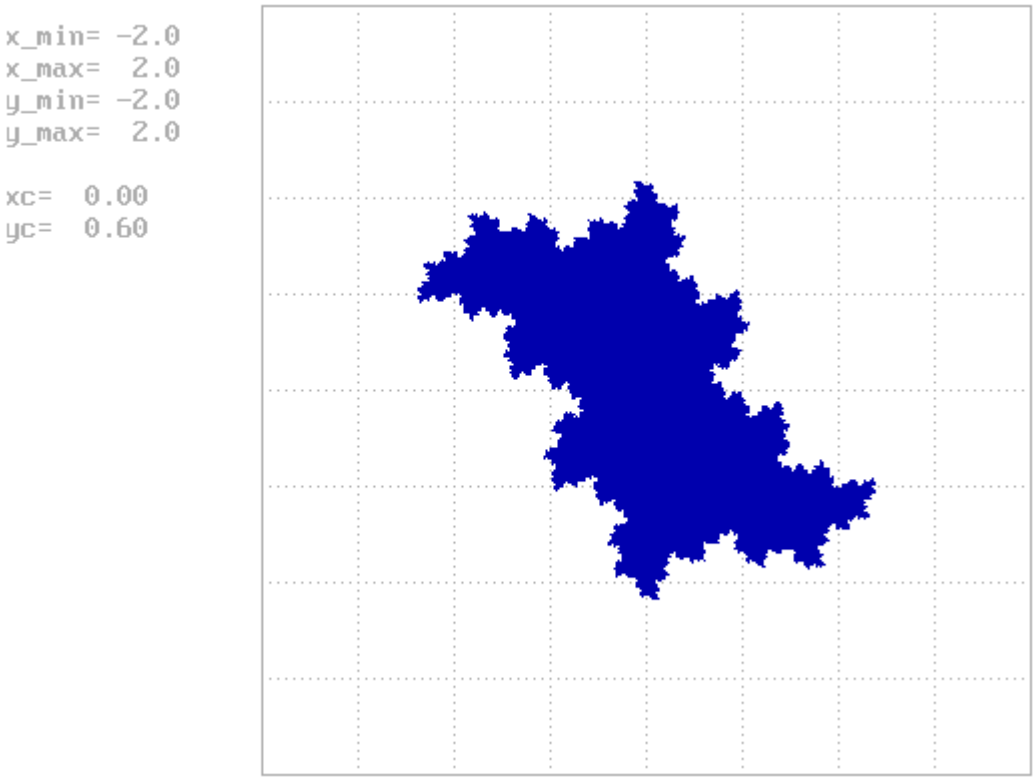
除了上面的正统形状，还有各种变形，如有时其周围伸长的叶子成涡旋状的扭曲，这时与其说叶子不如说是火焰，且扭曲的方向左右都有(如图 7.2.2(b) (d) (h))。再变形是叶子中间变细，像动物的腿脚(如图 7.2.2(f))。

围绕中心旋转的叶子或火焰的数目与 Julia 集合的成因密切相关，以 Julia 集合上的点为初值的数列的周期点数目在几乎所有的场合均和叶子的数目一致，如图 7.2.2(c)，浅蓝色的 Julia 集合，以它的点为初值的数列导致 3 周期振动，这时集合的叶子数也是 3。

特殊情况是叶子像节足动物的腿脚那样的情形，如图 7.2.2(f)，集合为 6 周期振动，然而却没有 6 枚叶子的中心。可是仔细观察可以发现 2 枚叶子和 3 枚叶子的两个中心，实际上此时是变成了两重结构，从根基上是(0.7, -0.1)附近的 2 枚叶中心，(0.15,-0.2)附近的 3 枚叶的中心是勒掉一枚叶子重新生成的。

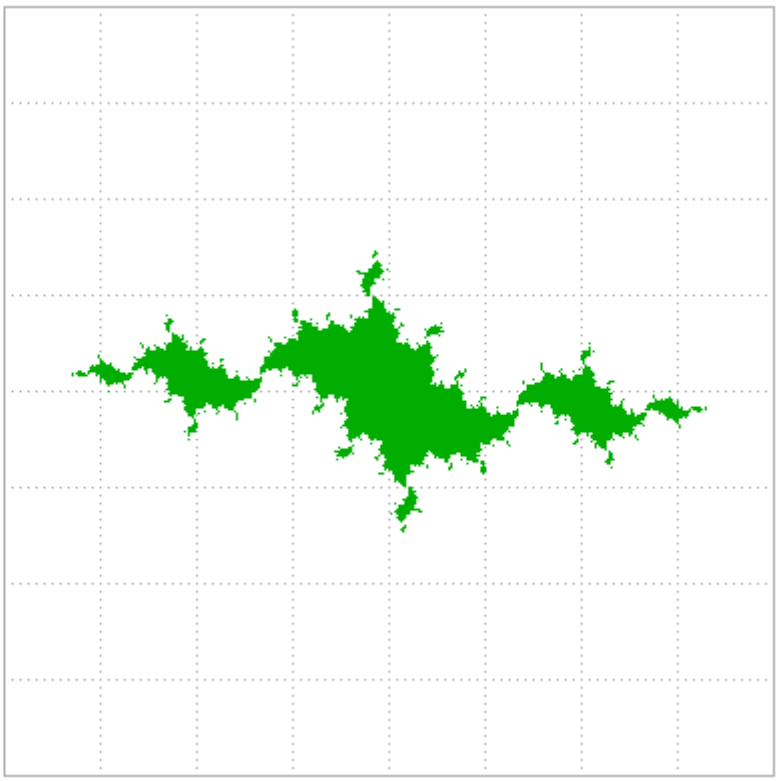
另外，图 7.2.2 是 8 个最具分形特征的 Julia 集合，随便代入 zc 的

值是无法描画出这些图形的，要描画特征明显的 Julia 集合必须知道某些规则，关于这些规则，下面会说明。



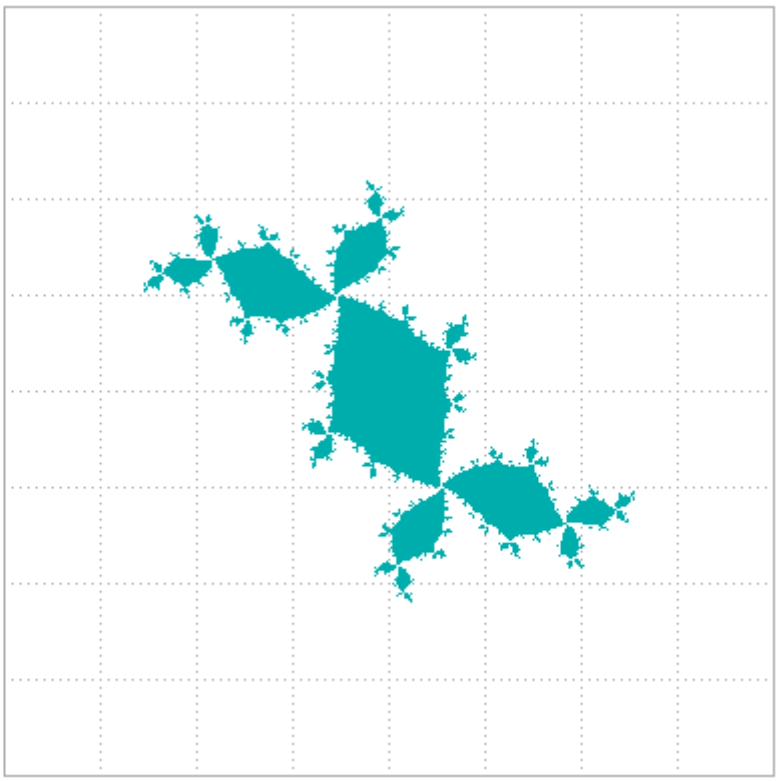
(a)　zc=0.6i，收敛(蓝色)

```
x_min= -2.0
x_max=  2.0
y_min= -2.0
y_max=  2.0

xc= -1.10
yc=  0.20
```



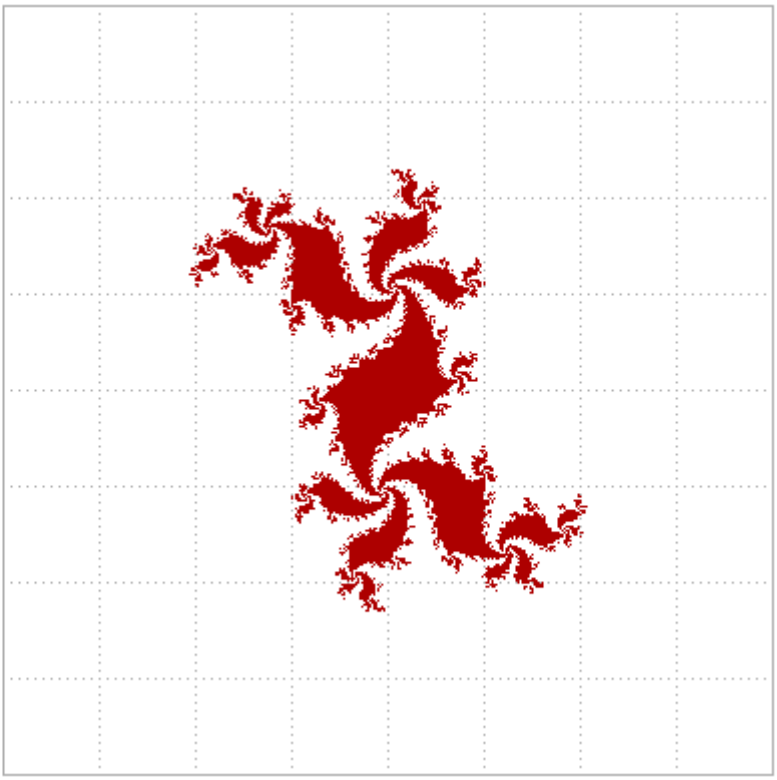
(b)　zc=-1.1+0.2i，2 周期振动(绿色)

```
x_min= -2.0
x_max=  2.0
y_min= -2.0
y_max=  2.0

xc= -0.10
yc=  0.77
```



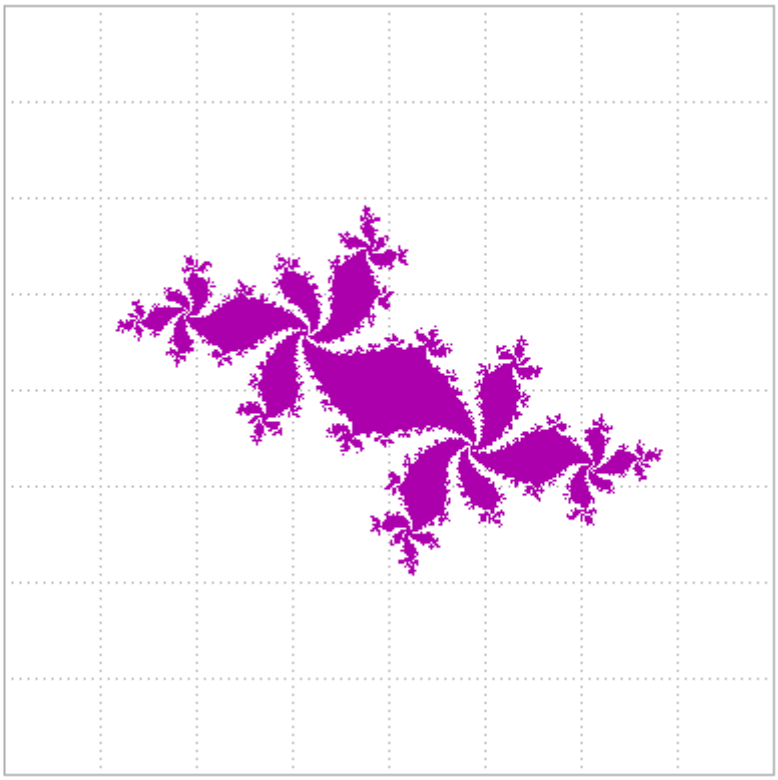
(c)　zc=-0.1+0.77i，3 周期振动(浅蓝色)

```
x_min= -2.0
x_max=  2.0
y_min= -2.0
y_max=  2.0

xc=  0.30
yc=  0.50
```



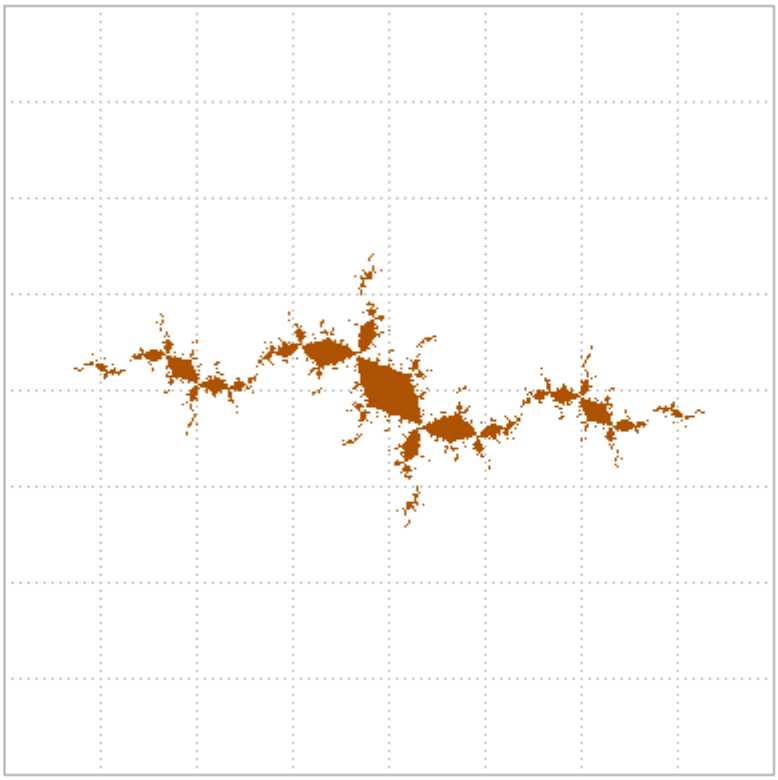
(d)　zc=0.3+0.5i，4 周期振动(红色)

```
x_min= -2.0
x_max=  2.0
y_min= -2.0
y_max=  2.0

xc= -0.52
yc=  0.55
```



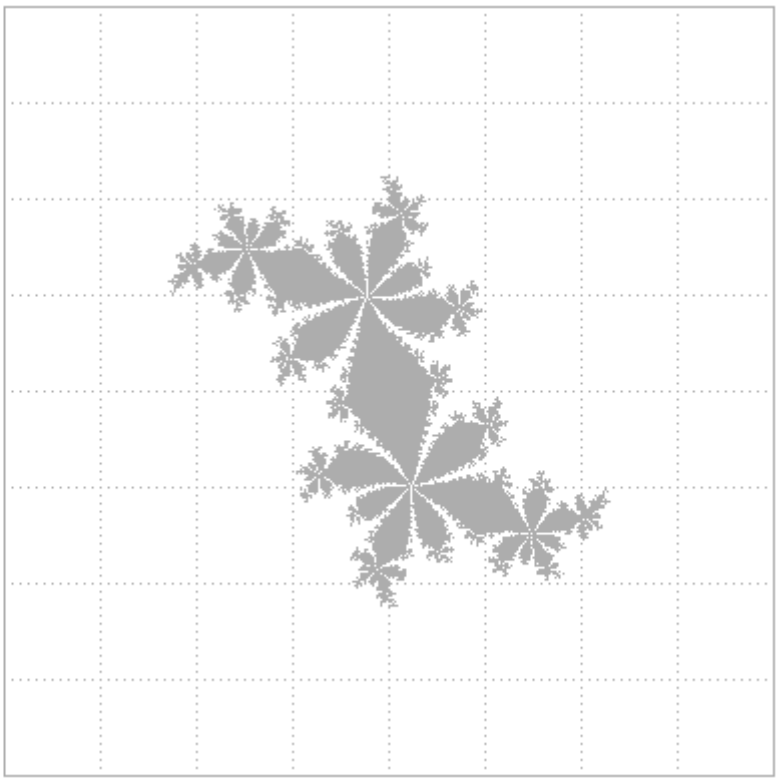
(e)　zc=-0.52+0.55I，5 周期振动(紫色)

```
x_min= -2.0
x_max=  2.0
y_min= -2.0
y_max=  2.0

xc= -1.15
yc=  0.25
```

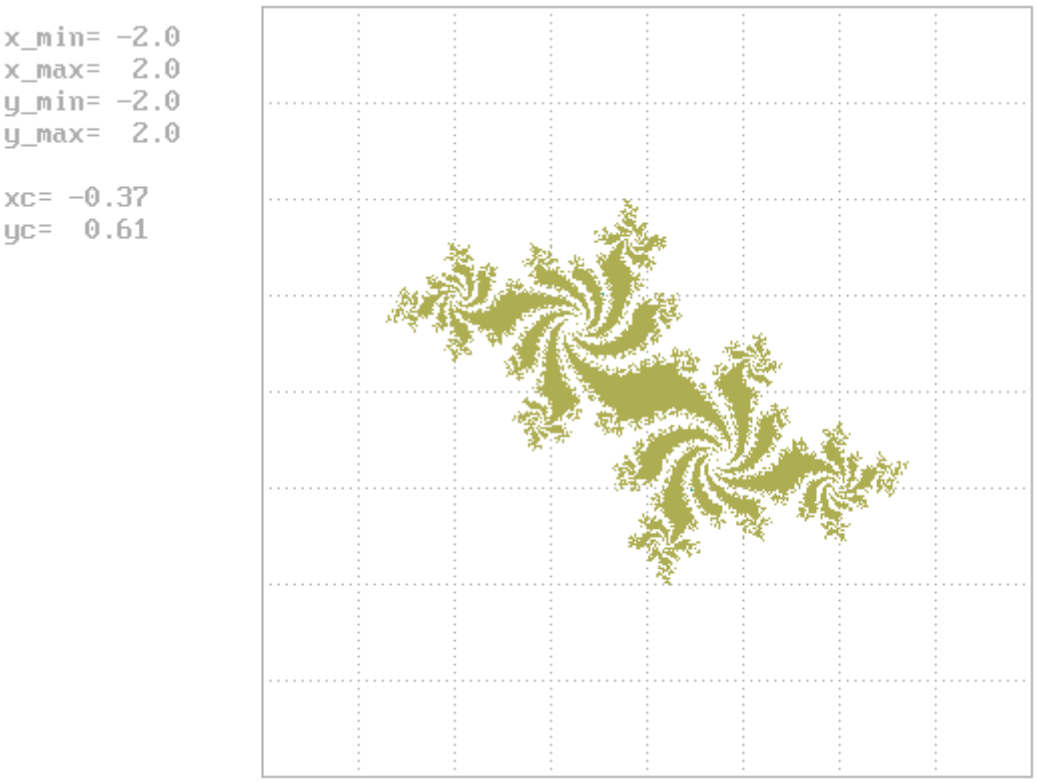


(f)　zc=-1.15+0.25I，6 周期振动(黄色)

```
x_min= -2.0
x_max=  2.0
y_min= -2.0
y_max=  2.0

xc=  0.12
yc=  0.61
```



(g)　zc=0.12+0.61I，7 周期振动(浅灰色)



(h)　zc=-0.37+0.61I，8 周期振动(浅黄色)

图 7.2.2　zn=zn-1^2+zc(zc 为复数)的 Julia 集合，描画区域:

x=-2.0¯2.0，y=-2.0¯2.0

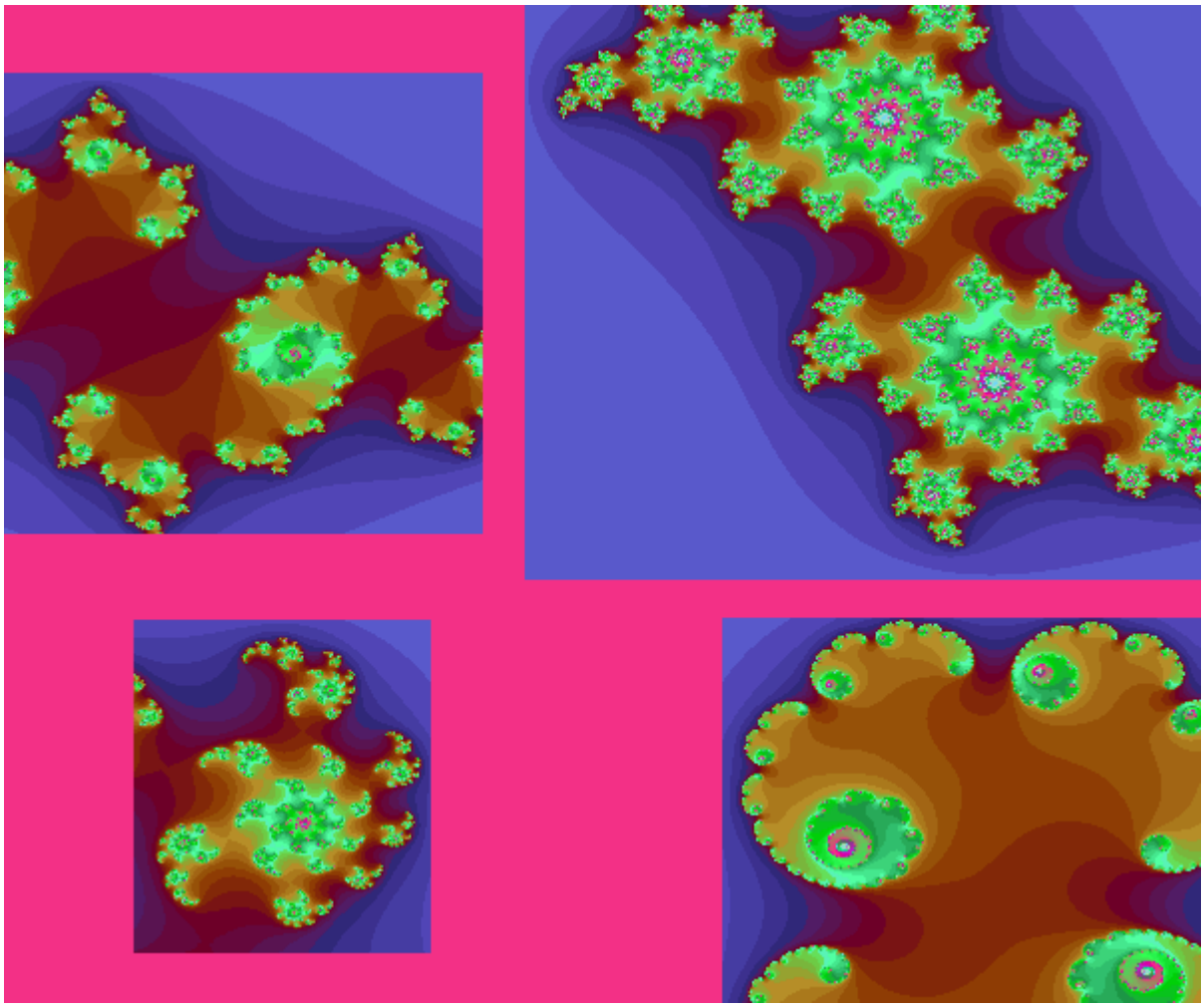


图 7.2.3 朱丽亚集图谱选

§7.3 Mandelbrot集合

特别地，对复数系列 zn=zn-1^2+zc，赋予初值 z0 且 zc=z0 并使得 zn 不发散的 z0 的集合称为 Mandelbrot 集合。这个数列 z0 兼有 zc 的作用，因而也决定了集合的形状，而且，这个形状也因次数不同而不同。

显然，Mandelbrot 集合是关于 X 轴对称的，也无需指定 zc，例如我们取 x_min=-2.25，x_max=0.75，y_min=-1.5，y_max=1.5，为提高精度，取 KL=300.

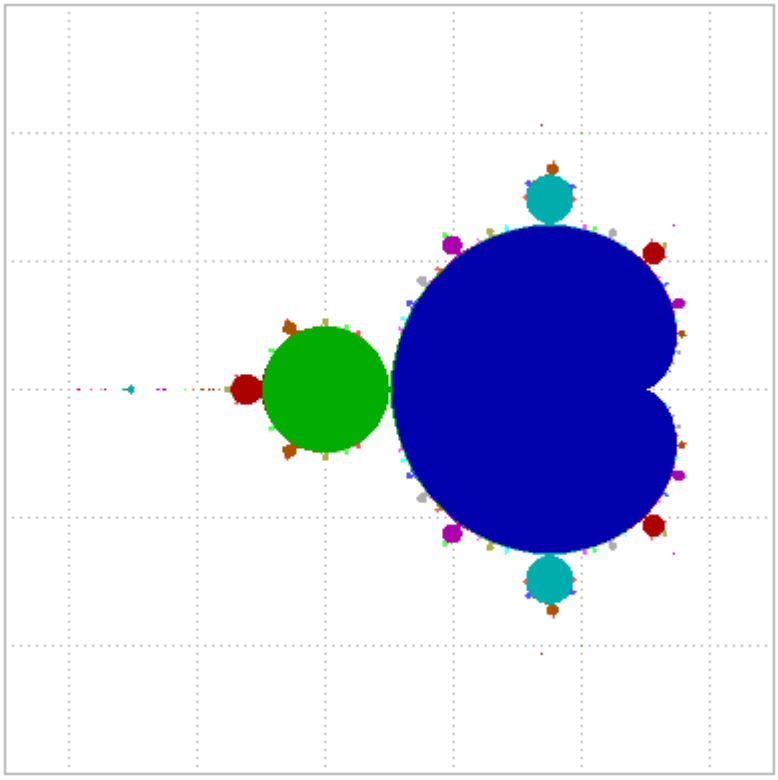


图 7.3.1　zn=zn_1^2+z0 的 Mandelbrot 集合

描画区域：x=-2.25~0.75，y=-1.5~1.5

这样描画出来的Mandelbrot集合与Julia集合不同，它用各种各样的颜色来着色，心形部分的主体为蓝色，左侧相邻部分的圆瘤为绿色，再左边的瘤为红色。另外，上下对称的耳朵样的瘤为浅蓝。其它大大小小的瘤也有其特定的颜色，这些颜色表明了以这些点为初值的数列的走向。从图中可以看到，Mandelbrot集合中存在着收敛、2 周期、3 周期和 4 周期振动等部分。源程序

复解析映射总是将复平面分解为两个互不相交的子集合。一个是稳定的集合，其上的动力学行为是平庸的；另一个是朱丽亚集 *J*，其上的映射是混沌的。数学家已经证明朱丽亚集 *J* 是完备集(即 *J* 与它的导集 *J′* 是一回事)，并且是不变的。

在复迭代中影响最大的当属迭代 *z*→*z*^2+*c*,实际上它只是形式更一般的复解析迭代 *z*_(*n*+1)=*F*(*z*_*n*)+*c* 的一种， *F* 是一个非线性函数。显然 *z*→*z*^2+*c* 也是最简单的一种，它在复迭代中的地位相当于逻辑斯蒂映射 *x*_(*n*+1)=a *x*_*n*(1− *x*_*n*)在实迭代中的地位。

考虑一般形式的 *F*, 令 *z*=*x*+i*y*, *c*=*c*_(*X*)+i*c*_(*Y*)，其中 i 表示虚数，i=SQRT(−1)。分离实部与虚部，具体化迭 代关系便有:

$$x \to f(x, y) + c_{-}(X),$$

$$y \to g(x, y) + c_{-}(Y).$$

图 7.3.2 芒德勃罗集逐步放大图

图 7.3.3 芒德勃罗集逐步放大图

图 7.3.4 芒德勃罗集逐步放大图

80 年代初芒德勃罗在迭代 $z \to z^2+c$ 时，发现了著名的芒德勃罗集，简称 *M* 集。当时迭代精度和色彩调配均不理想，显现的 *M* 集也不好看。但是过了不久，许多高质量的 *M* 集图片纷至沓来，尤以德国布来梅大学动力系统图形室所作的图片最为精美，受到举世赞誉。

随之而来的是，各大学的教师、研究生以及本科生纷纷利用自己的计算机试算复迭代， *M* 集一时泛滥高等院校。据说有一段时间校方被迫作出规定，不允许利用实验室的公用计算机玩芒德勃罗集。

在当今时代，您自己购买了一台微机，如果不在上面玩一玩 *M* 集和 *J* 集，实在太遗憾了。看了本书后，读者一定要亲自试一试。编写计算 M

集和 J 集的程序并不复杂，您可以参照本书给出的 PASCAL 程序，编制适合您自己使用的更好的程序，您可以用 PASCAL，C， C++，Java,甚至 BASIC 语言。

图 7.3.5 芒德勃罗集“峡谷地带”放大图

通常所说的 M 集是迭代二次函数 $z\to z^2+c$ 产生的，此函数具体化就是

$$x\to x^2-y^2+c_x\ (X),$$

$$y\to 2xy+c_y\ (Y).$$

其中 $z=x+iy$, $c=c\_(X)+ic\_(Y)$, 以横轴 x 记录实数的实部，以纵轴 y 记录实数的虚部。M 集合实际上是常数 $c=(c\_(X), c\_(Y))$ 构成的图象。让 c 从屏幕左上角开始变化，逐行增加，一直变到屏幕右下角。如果取的区域是 200×200，则一共要计算 40,000 个点，把计算的结果用不同的颜色标记下来，就得到一幅图象，这就是 M 集。对于不同的 c 值，如何得到表征迭代性质的不同的结果呢?

容易知道，无穷远处肯定是迭代的一个吸引子，即对于复平面上相当多的初始条件，迭代最终都跑到无穷远处。但研究发现，在原点附近还存在一个奇特的区域，在迭代过程中此区域永远不会跑掉。在非严格的意义上，这个不变的集合就是 M 集，我们的主要任务就是画出这个集合的边界——实际上边界是分形曲线，极其复杂，M 集图象的全部魅力就在这里。

在复平面上，我们以原点 (0, 0) 作为参考点，观察迭代过程是否远离原点，以及逃离原点的速度如何。为此规定一个距离函数

$D=x\^2+y\^2$,

其实 D 有许多不同的取法，以上取法是最普通的。可以看出，如果 D 较大，表 明迭代点离原点较远，如果 D 较小，表明迭代点离原点较近。假设对于任何一个 c，迭代都从 $(x\_0, y\_0)=(0, 0)$ 开始，我们观察迭代点列

(x_1, y_1), (x_2, y_2), (x_3, y_3), … , $(x\_(100), y\_(100))$, …

的变化状况。每一次都计算一下 D值，即该点与原点(0，0)的距离(为方便计，这里 实际上计算的是距离的平方)。取一个参考距离 R，不妨取得大一些，比如 R=40(实际上取 20 就足够了)。现在想知道迭代多少次后实际的距离 D大于 R 。在迭代过程中如果 D小于 R，则继续让计算机迭代，要规定一个上限， 比如 300 次。如果迭代了 300 次后结果仍然不跑掉(即 D仍然小于 R)，则可以近 似认为此点属于 M集合。

对于迭代次数小于 300 次的情况，如果迭代 10 次 D就大于 R，则标记 c点为白色；如果迭代 35 次 D开始大于 R，则标记 c点为红色；如果迭代 110 次 D开始大于 R，则标记 c点为蓝色，等等。

标色有很多技巧，表面看来好像属于计算机技术，但实际上这属于传统的美术。懂得传统美术色彩理论的人，在此大有用武之地。一幅分形图形标上不同的颜色，就有完全不同的视觉效果，虽然本质上具有同样的数学结构。因此，从这种意义上说，分形图形艺术是传统美术与计算机的结合。如果掌握了计算机这个工具，甚至完全可以说，这与传统的美术没有本质区别。

§7.4 Julia集合解密

前面我们说到，只有取特定的 zc 才能描画出具有明显特征的 Julia 集合，那么这些 zc 该如何取呢？我们将前面 Julia 集合的 8 个 zc 值重叠到 Mandelbrot 集合上可以得到图 7.4.1，那么除了图 7.4.1 外，其余的 zc 都在 Mandelbrot 集合周边瘤的位置上，而且可以发现 Mandelbrot 集合上这些点的颜色和以这些点作为 zc 的 Julia 集合的颜

色是一致的。例如，Mandelbrot 集合主体的右上侧有个红瘤，而以此为 zc 的 Julia 集合也有 4 枚红叶，因此，若以 Mandelbrot 集合周边的瘤作为 zc，则可以描画出具有这些场所的颜色所表示的枚数的 Julia 集合。

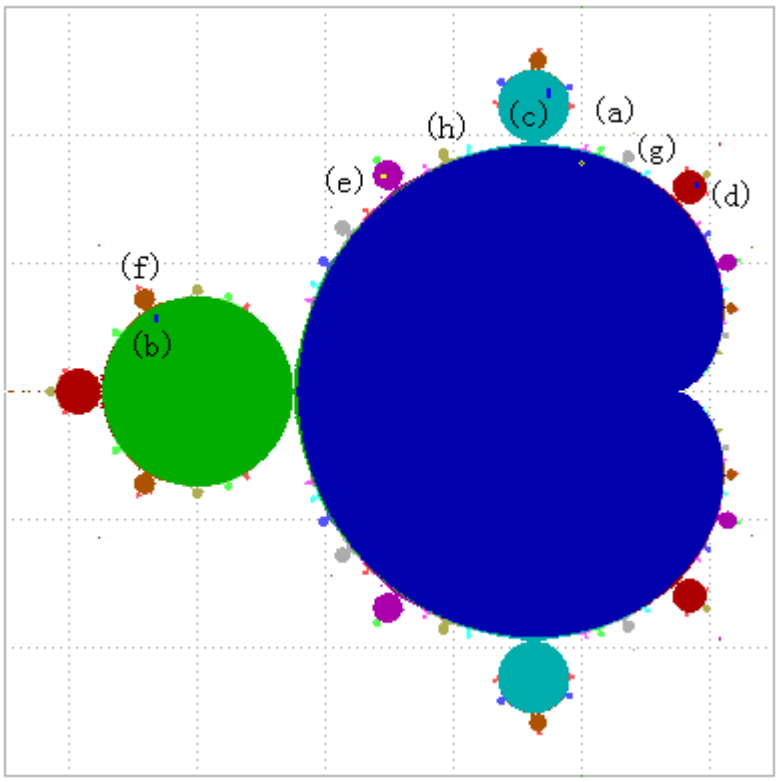


图 7.4.1 8 个 Julia 集合 zc 值在 Mandelbrot 集合上的分布

描画区域：x=−1.5¯0.5，y=−1.0¯1.0

下面我们更加细致地看看这些瘤和 Julia 集合之间的关系。取一个瘤，选若干个点时，以这些点作为 zc 的 Julia 集合会有怎么样的变形呢？图 7.4.2 是图 7.4.1 主体右上侧的 4 周期的红瘤放大图，其顶部连接着淡黄色的 2 次瘤，以图中的 10 个点作为 zc 的 4 叶 Julia 集合如图 7.4.3 所示。

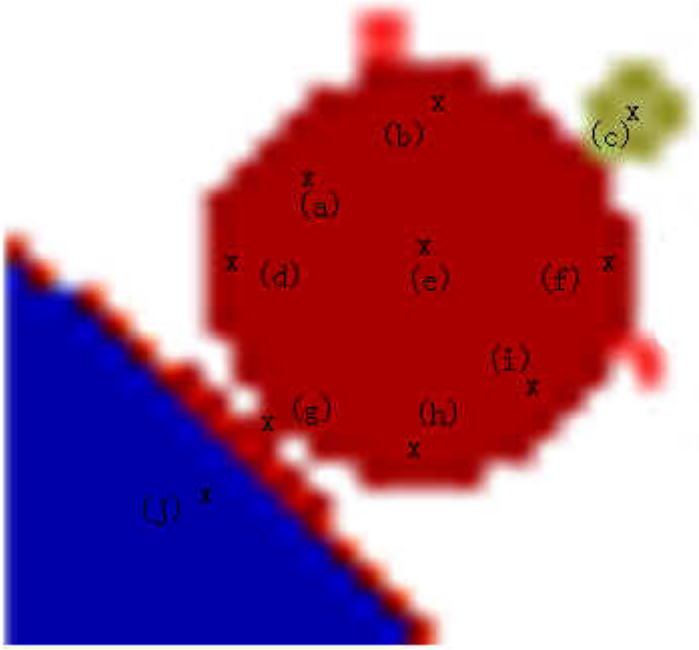


图 7.4.2 图 7.4.3 的 10 个 zc 值在 Mandelbrot 集合上的分布

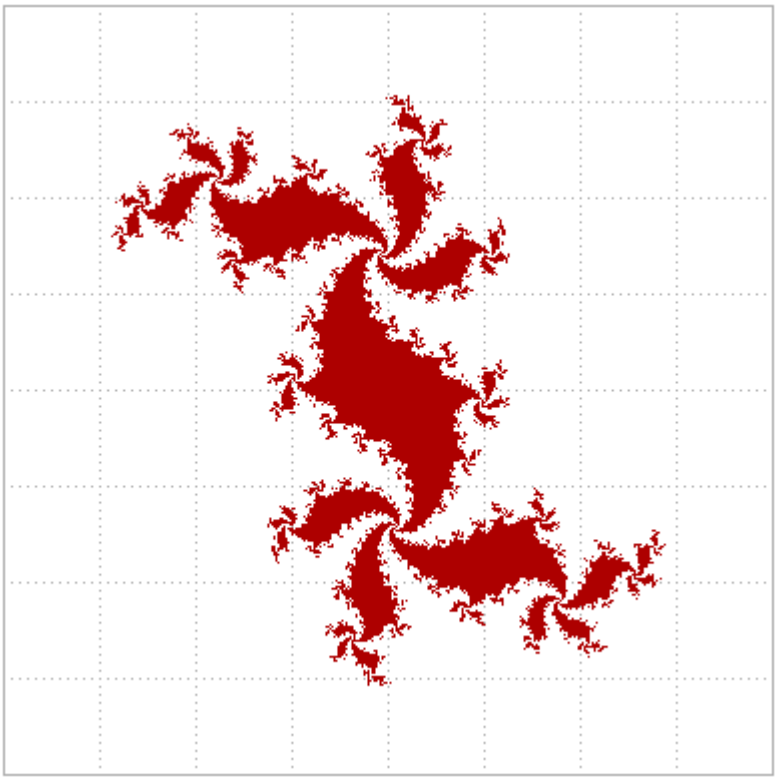
描画区域：x=0.2¯0.4，y=0.45¯0.65

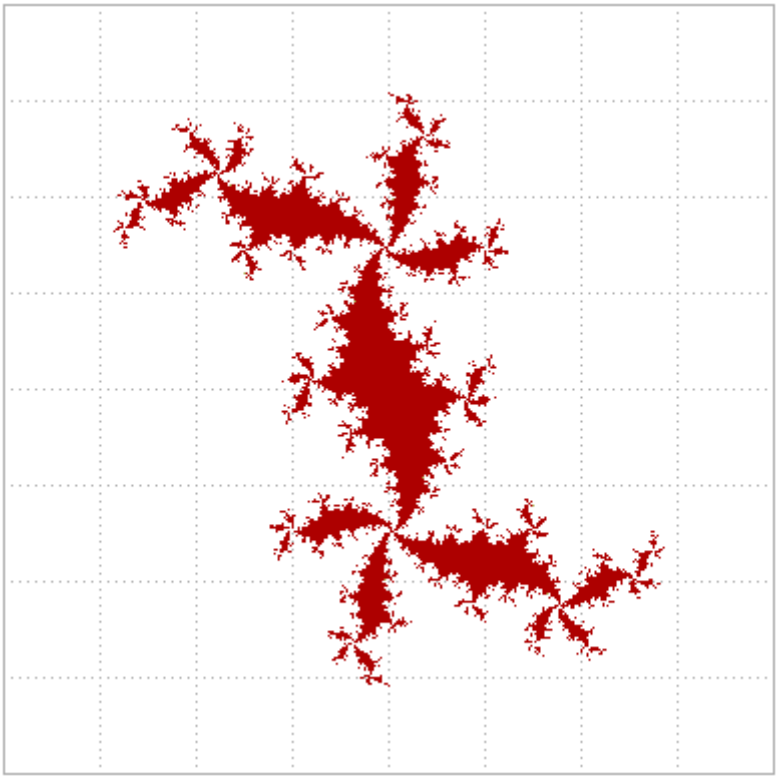
从瘤的根到顶部的中心点上看不到涡旋状的弯曲(图 7.4.3(e) (g))，但从根走到顶部，叶变得越来越细，近似于节足动物的腿脚，离根部最远的图 7.4.3(c)的颜色为淡黄色，以此作为 zc 的数列将是 8 周期振动。反过来，离根部越近叶越粗，从而与相邻叶连成一片变得不能识别了，这就是描画色为蓝色的收敛区域(见图 7.4.3(j))。

火焰样的漩涡将在中心线的两侧发生，以中心线为界，其左右的旋转方向相反。向外的在中心线左侧的点为 zc 的 Julia 集合将是左漩涡(图 7.4.3(a) (b) (d))，反之中心线右侧的点为 zc 的 Julia 集合将产生右漩涡(图 7.4.3(f) (h) (i))。无论在哪一侧，离根越近漩涡越大。

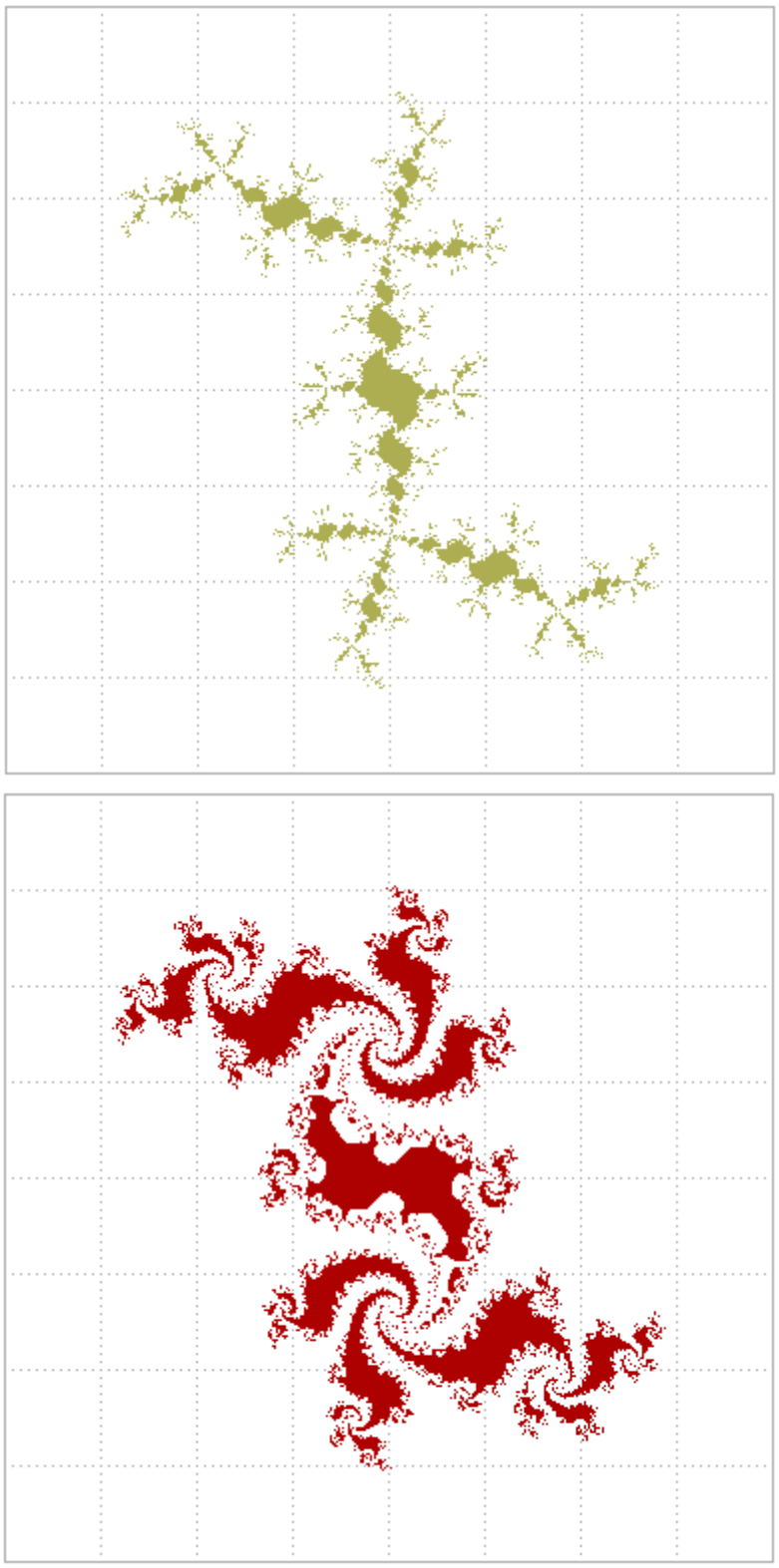
Mandelbrot 集合是 Julia 集合的缩图，这里总结了 Julia 集合的所

有知识，知道了两者的关系，那么某 zc 的 Julia 集合会是什么样的形状就可以容易地预想到了。

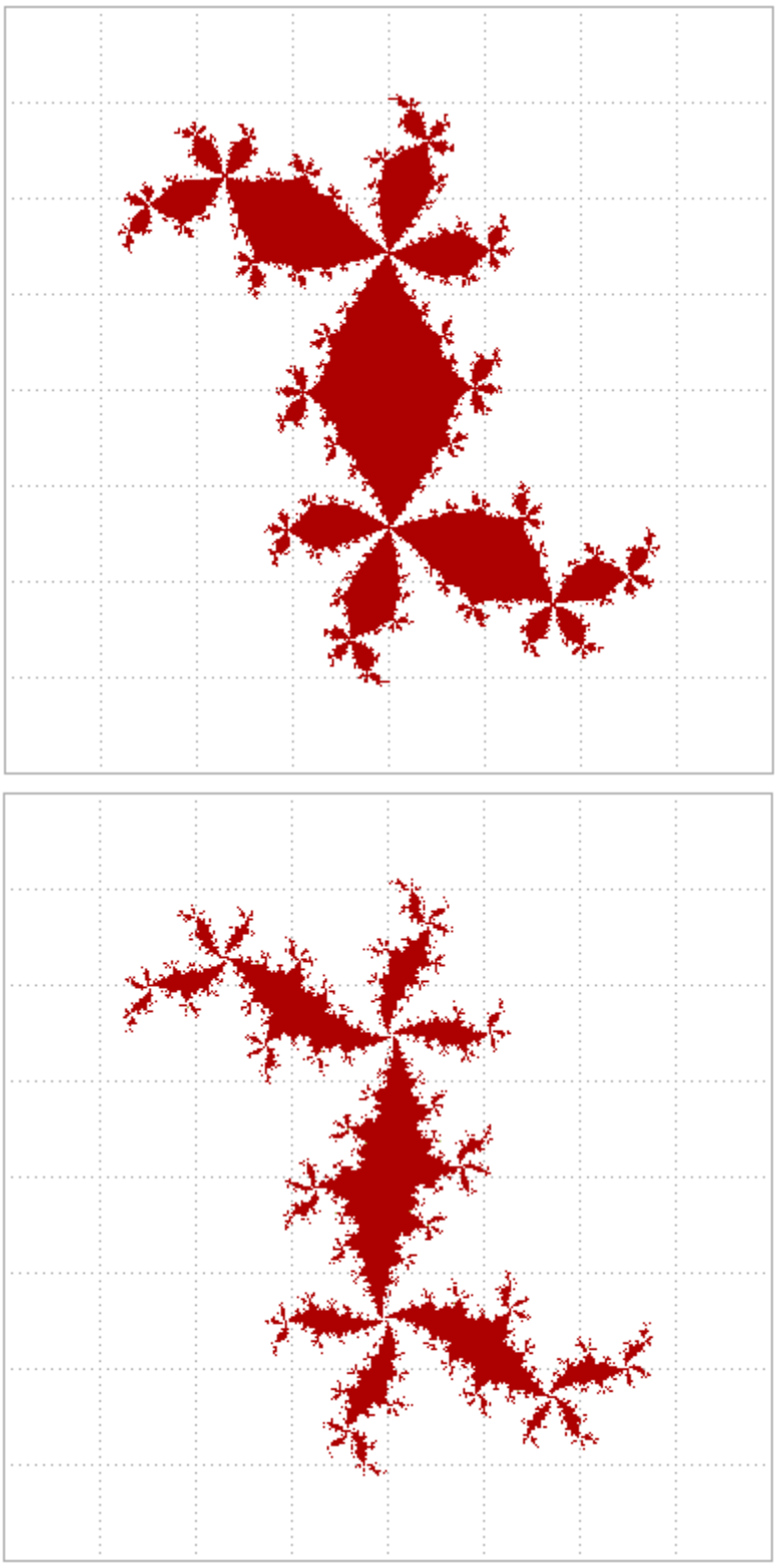




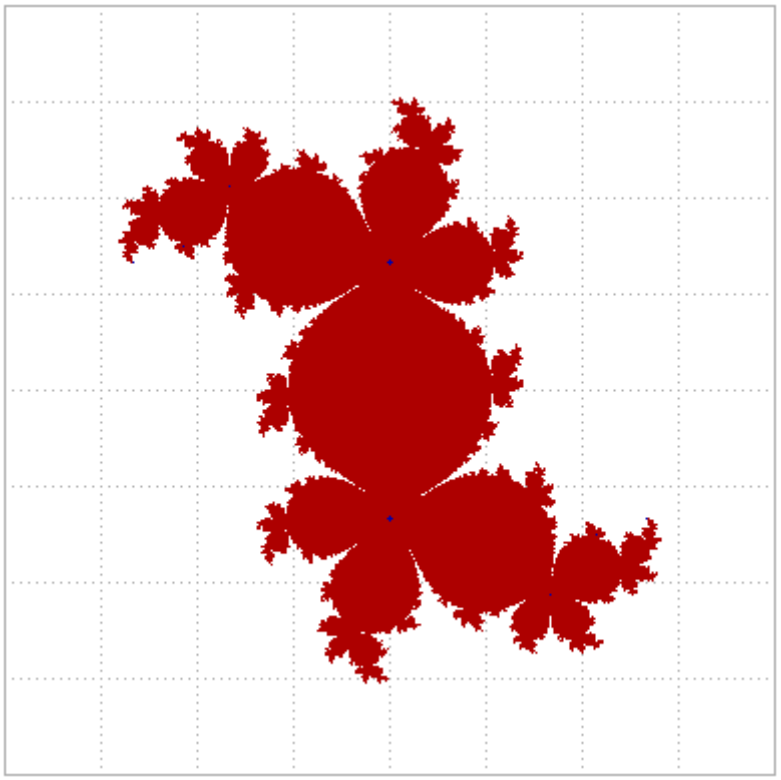
(a) (b)

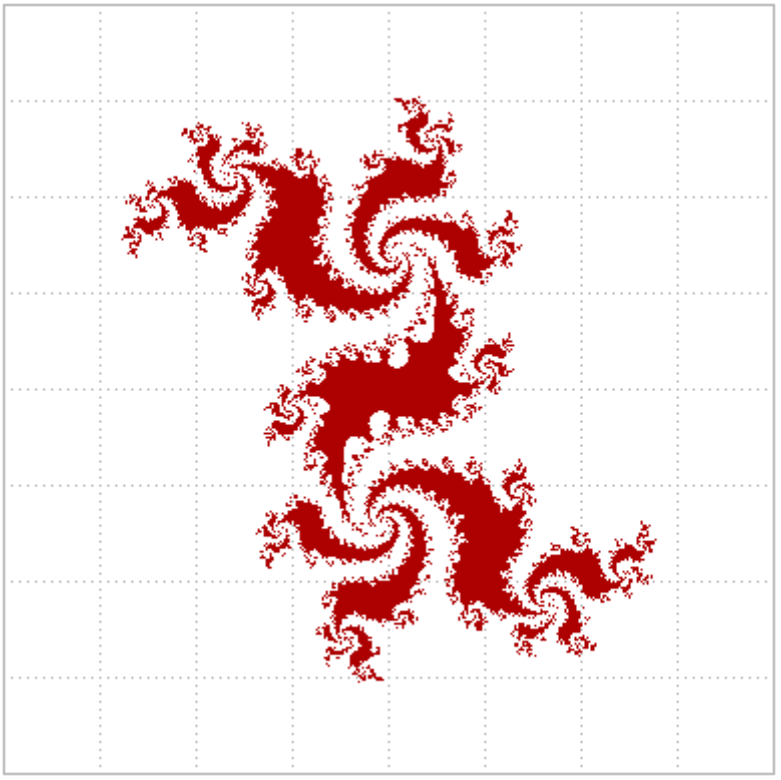


(c) (d)

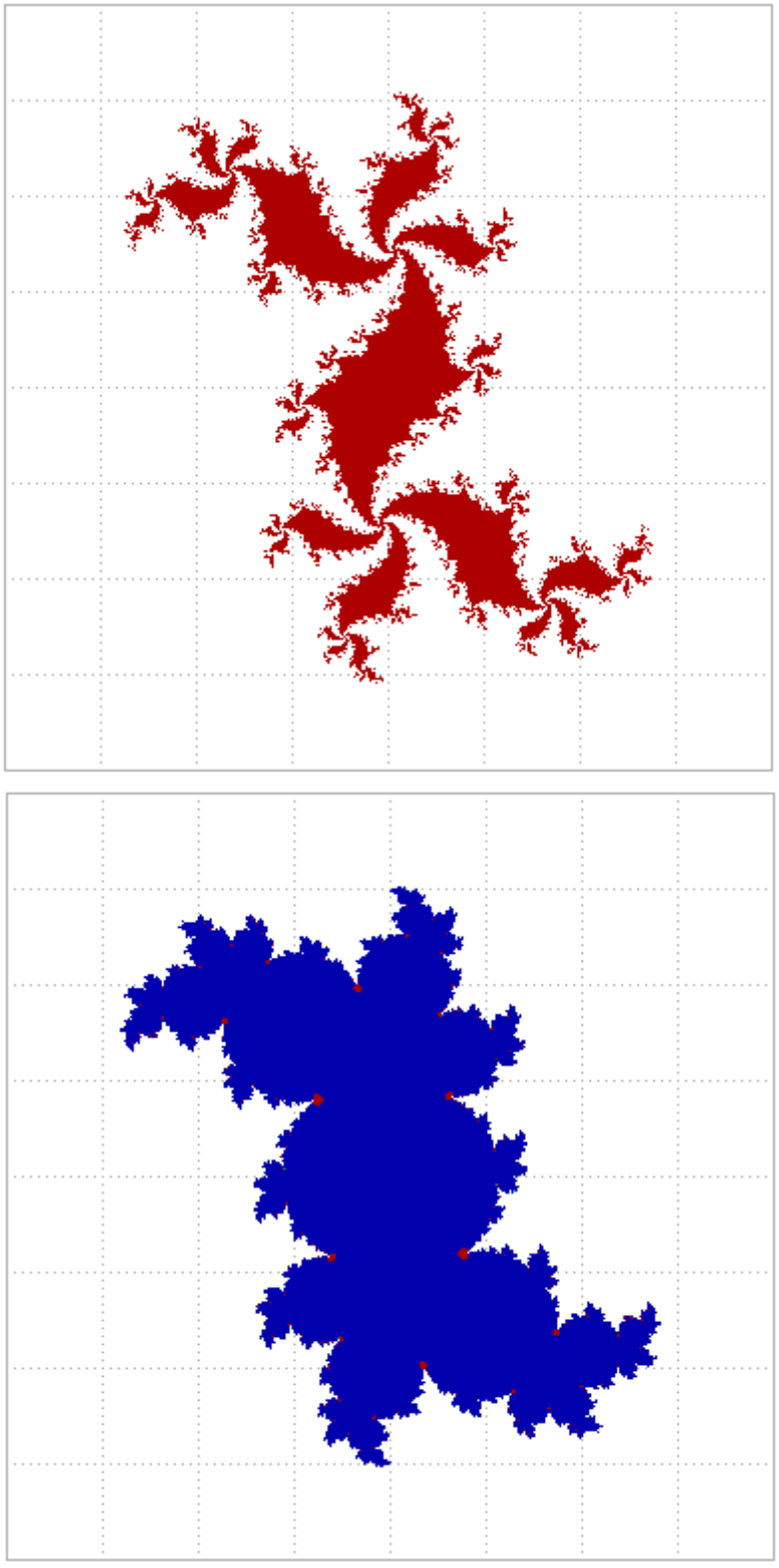


(e) (f)





(g) (h)



(i) (j)

图 7.4.3 根据不同的 zc 值，Julia 集合形状的变化

描画区域：x=-1.5~1.5, y=-1.5~1.5

(a) zc=0.26+0.56I; (b) zc=0.29+0.57I; (c) zc=0.32+0.57I; (d) zc=0.24+0.54I;

(e) zc=0.29+0.54I; (f) zc=0.32+0.54I; (g) zc=0.25+0.5I; (h) zc=0.29+0.49I;

(i) zc=0.31+0.51I; (j) zc=0.24+0.49i.

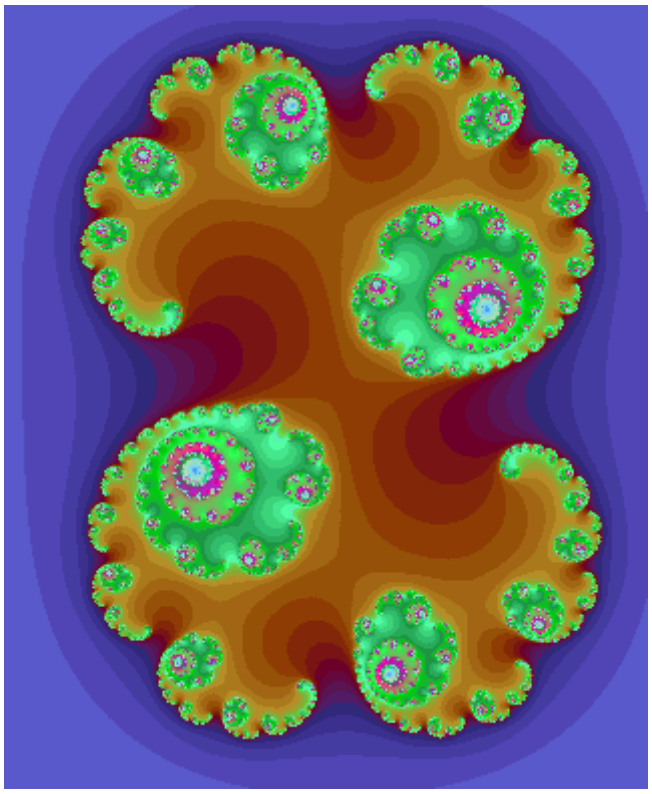


图 7.2.4 对应于右侧芒德勃罗集的朱丽亚集的形状

§7.5 高维和高次情形

7.5.1 高维情形

通常我们是在二维复平面上研究广义的 *M* 集和 *J* 集，也可以通过“四元数”(quaternions)将它们推广到高维空间中去。在二维复平面中表示复数只用两个基向量：1 和 *i*。在四维空间中讨论超复数，现在有四个基向量：1，*i*，*j* 和 *k*。 任一复数可以表示为

$$q=x+yi+zj+qk.$$

超复数基向量之间的运算关系(注意，不同于传统上四元数基向量之间的运算关系)为：

$$ij=ji=k,\quad jk=kj=-i,\quad ki=ik=-j,\quad ii=jj=-kk=-1,\quad ijk=1.$$

注意，运算关系的规定多少有些任意性，也可以规定 *i*^2=*j*^2=*k*^2=+1。在四维空间 *H* 中也可以研究迭代 $x\rightarrow x^2+c$ 下的超朱丽亚集 *J*， 选一个截面，将超朱丽亚集投影到三维空间中，可以得到立体的 *J* 集图象。

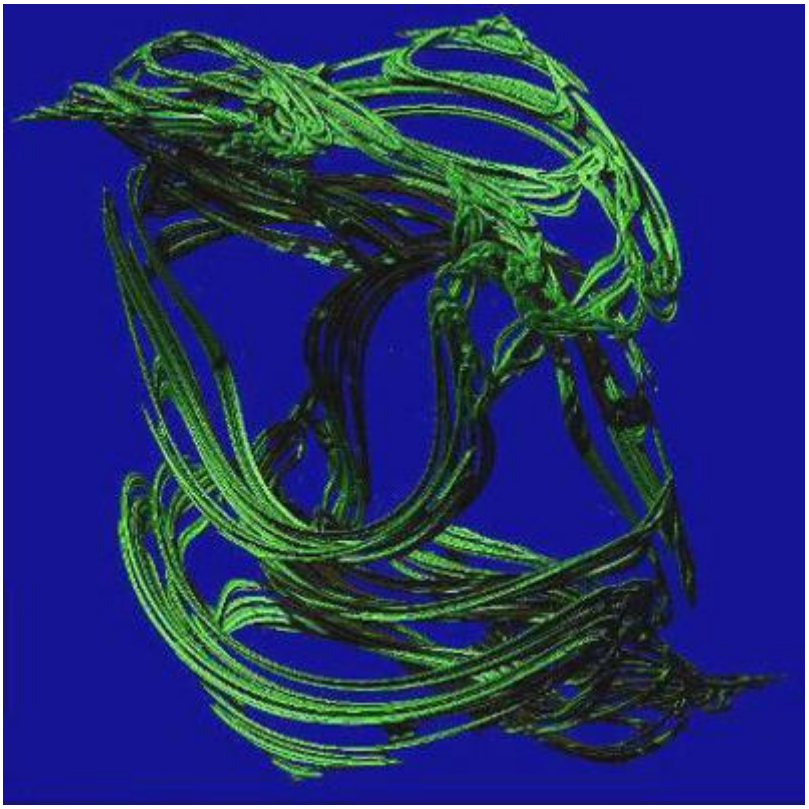


图 7.5.1　用四元数法得到的高维朱丽亚集的投影图

7.5.2 高次情形

在上面的讨论中，复数数列都限于 2 次函数，这是非线性映射中最简单的，可是其描画出的图形就已经是非常复杂了。

这儿，我们试着将 2 次函数改为 3 次函数，看看描画的 Julia 集合、Mandelbrot 集合有什么特征。这时复数数列的迭代式为 zn=zn-1^3+zc，程序中相应的实部和虚部的计算为：

x[k]=x[k-1]*x[k-1]*x[k-1]-3x[k-1]*y[k-1]+xc;

y[k]=3*x[k-1]*x[k-1]*y[k-1]-y[k-1]*y[k-1]*y[k-1]+yc。

伴随着形状的变化，函数 plot 和 main 中的变量的变域均必须改变。可以看出，3 次函数的 Julia 集合是绕原点的 3 次对称，而 3 次函数的

Mandelbrot 集合是关于 X 轴和 Y 轴对称的，如图 7.5.2.¯7.5.2 所示。

图 4.6.1　红瘤右上例的小 Mandelbrot 集合,描画区域：x=0.35¯0.366, y=0.636¯0.652

图 4.6.2　zc=0.359+0.643i 的 Julia 集合，描画区域：　x=-1.5¯1.5, y=-1.5¯1.5

图 4.6.3　3 次函数的 Mandelbrot 集合，描画区域：x=-1.5¯1.5, y=-1.5¯1.5

图 4.6.3　3 次函数的 Julia 集合，描画区域：x=-1.5¯1.5, y=-1.5¯1.5

(a)　zc=0.57+0.24I，4 周期振动(红色)

(b)　zc=0.35+0.69I，5 周期振动(紫色)

(c)　zc=0.22+0.96I，6 周期振动(黄色)

7.5.3 广义芒德勃罗集和朱丽亚集

在科学史上，非线性科学家首先研究了形如 $z \to z^2+c$ 之类的复映射 ，通过等势面着色法，得到了令科学界为之激动不已的 M 集，以及各式对应的 J 集。

除了二次映射，是否可以考虑任意多项式映射，也可以包含三角函数、指数函数，甚至对数函数呢?

没有人说不可以。考虑了那些函数后，迭代是否很复杂呢?当然很复杂，特别是想给出解析结果，就比较困难。但是，在这里我们关心的不是数学严格性，而是结果是否具有美学价值。

有了这样的认识，事情就好办多了，你可以根本不管函数的性质，拿过来，装进机器，算它一算再说。你明白了!有点儿头脑的人立即可以做许多尝试，用不了多久，保证有惊人的发现。我说的不是科学上的发现，而是美学上的发现。你能够做出其他任何人未曾想到的美妙图形。

这样得到的两类图形可以粗略地叫做广义 M 集和广义 J 集，它们是分形艺术图形创作 的主要内容所在。

你可以试一试下述复映射:

$z\to z\hat{}3+c$,

$z\to z\hat{}4+c$,

$z\to z\hat{}5+c$,

$z\to z\hat{}6+c$,

$z\to z\hat{}7+c$,

等等。这些迭代看似很简单，但化成实数形式还是有些复杂性，不过并未超出中学数学知识。如果读者中有谁对复数乘法不熟悉，可以借机复习一遍。这些函数化成实数迭代，分别有如下具体形式:

$x\to x\hat{}3-3xy\hat{}2+c\_(X)$

$y\to 3x\hat{}2y-y\hat{}3+c\_(Y)$,

x→x^4−6x^2y^2+y^4+c_(X)

y→3x^3y+x^3y−4xy^3+c _(Y)，

x→x^5−10x^3y^2+5xy^4+c_ (X)

y→y^5−10x^2y^3+5x^4y+c_ (Y)，

x→6x^5y−20x^3y^3+6xy^5+ c_(X)

y→x^6−y^6−15x^4y^2+15x^2y^4 +c_(Y)，

x→y^7+5x^6y−5x^4y^3−9x^2 y^5+c_(X)

y→x^7−9x^5y^2−5x^3y^4+5x y^6+c_(Y).

事实上,对于任意不小于二次的复多项式 $Q(z)$,在迭代过程中都有性质: $Q(\infty)=\infty$，并且 $Q'(\infty)=0$，于是对于单参量族 $Q_c(z)$ (比如 $Q_c(z)=z^n+c$)，我们总可以定 义一种类似芒德勃罗集的集合。以复二次映射为例,令 $|c|>2$, $|z|>|c|$,则当 $n\to\infty$时, $Q^n_c(z)\to\infty$。设 $A_c(\infty)$ 代表∞的吸引域，C代表复平面，广义芒德勃罗集合可 写作

$M_(Q_c)=\{c\in C:$ Q_c 不属于 A_ $c(\infty)$ 的所有有限临界点$\}$.

因为∞是吸引的，所以总可以定义

$J_(Q_c)$=偏 $A_c(\infty)$

为 Q_c 的朱丽亚集。对于任意 $c\in M_(Q_c)$， 朱丽亚集 $J_(Q_c)$ 是连通的。

最后提一个小问题:如何生成用光滑曲线表示的 M 集或者 J 集的等势线?

基本想法是以黑白两色循环标色，然后用 PhotoStyler 处理。具体步骤是:1)用 Fart1995 得到黑白交替的等势面图形;2)调入 PhotoStyler 2.0 将它转成灰度图; 3)打开“效果”选单，选“突出”一项，再选“确定边界”子项; 4)将图形反转,把图形由灰度图变成 1 位的黑白图,用 GIF 格式存起来。

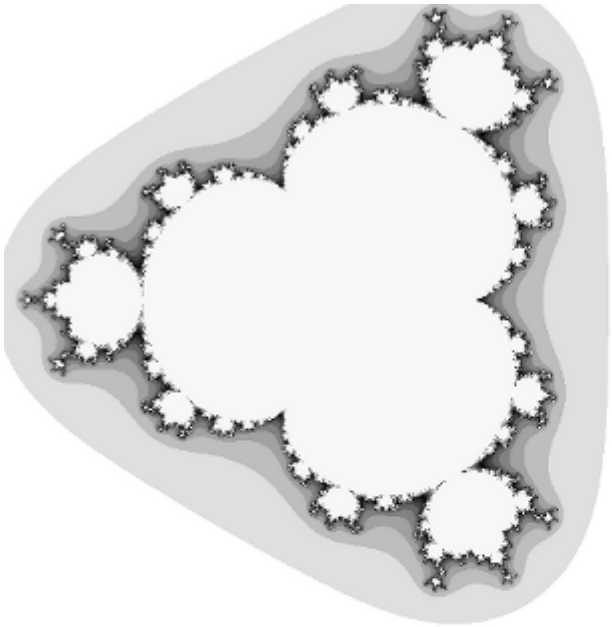
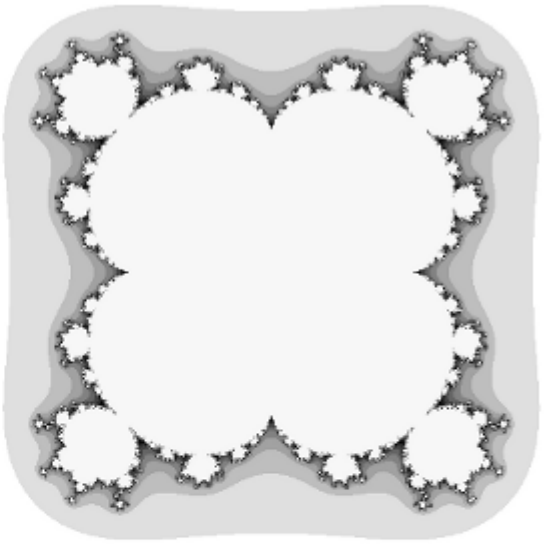
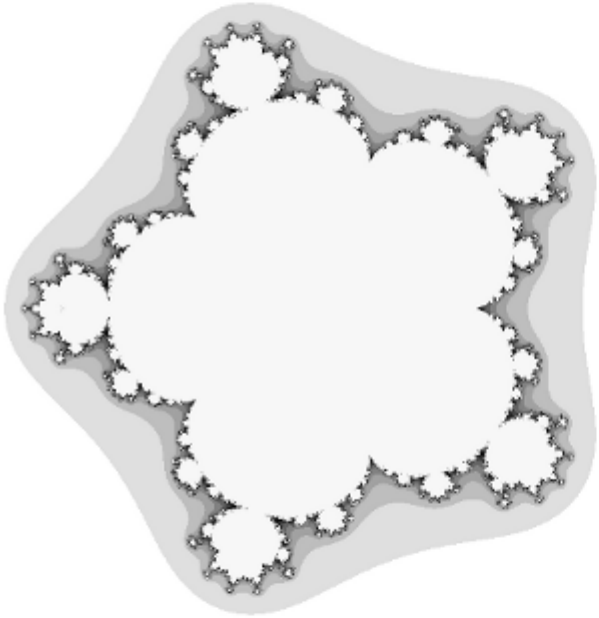
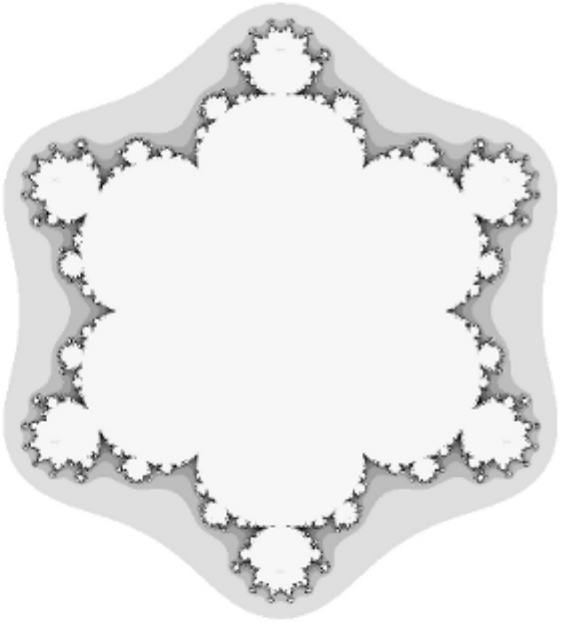
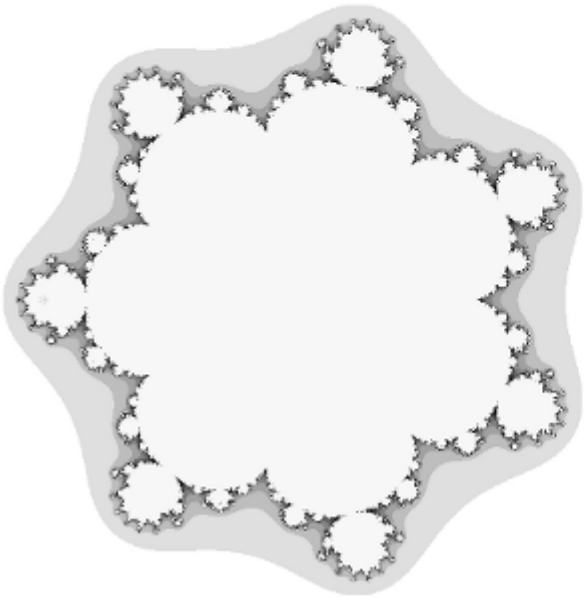
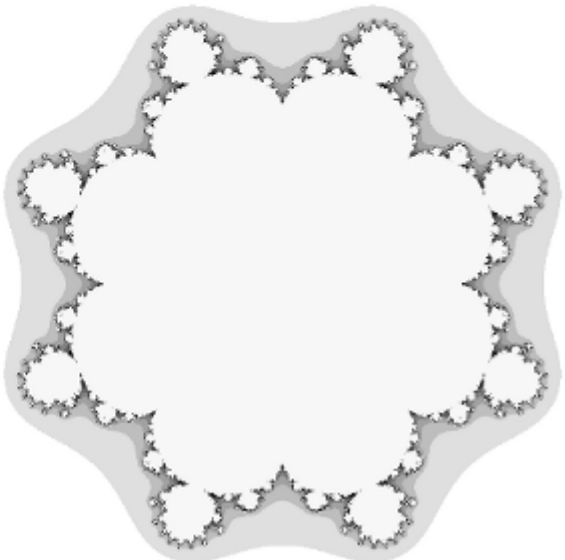







图 7.5.6 广义芒德勃罗集，$z\to z^m+c$,其中 m分别为 4， 5，6，7，8，9.

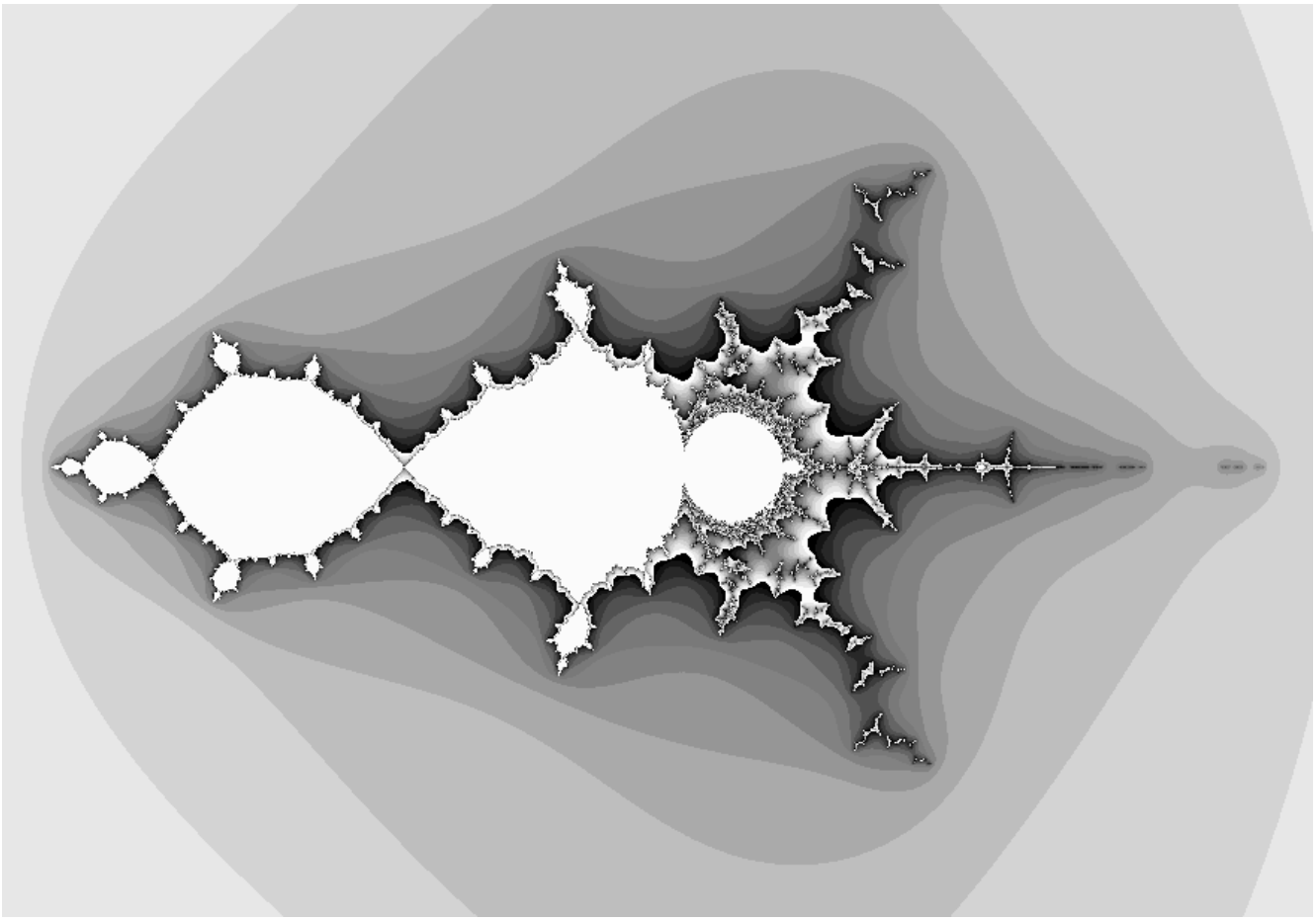


图 7.5.7 一个广义芒德勃罗集

§7.6 牛顿法求根

求代数方程 $f(x)=0$ 的精确解是很难的事情，特别地当 $f(x)$是高于5次的多项式时,不能通过多项式系数的有限次运算得到根的表达式。在这种情况下求方程的近似解却是可以的，牛顿法就是一种比较好的逐次逼近法。牛顿法在求根过程中逼近很快,用计算机计算是十分方便的。

牛顿法的本质仍然是“以直代曲”，首先猜测一个值 x_1，用它近似方程的根 c，用过 $(x_1, f(x_1))$ 点的切线

$$y=f(x_1)+f'(x_1)(x-x_1)$$

近似代替曲线 $f(x)$，然后用切线方程

$$y=f(x_1)+f'(x_1)(x-x_1)=0$$

的根

$$x=x_2=x_1-f(x_1)/f'(x_1)$$

近似代替曲线方程的根 c，这样就得到 c 的第二个近似值。依此类推可得到迭代公式

$$x_{n+1}=x_n-f(x_n)/f'(x_n).$$

在复平面上选定一个区域，对于任意初始点(除去(0，0)点)，讨论它在牛顿法迭代过程中的行为。一般选 $f(x)=x^p-1$，其中 p 是大于 2 的正整数。这样，迭代公式还可以改写为

$$x_{n+1}=[(p-1)x^p_{n+1}]/px^{p-1}_n.$$

对于 $x^3-1=0$，有三个根：

$x_1=1$，

$x_2=[-1+\mathrm{SQRT}(3)\,i]/2$，

$x_3=[-1-\mathrm{SQRT}(3)\,i]/2$，

三个根均匀地分布在单位圆上。这三个根周围构成三个“吸引盆”(attractor basin)，初始点迅速被吸引到盆内，最后停止在三点之一。用计算机迭代，以当前点到三个终点的距离远近为标准，标上不

同的颜色，就能得到美丽的分形图，特别是在120°线、240°线附近有复杂的“项链”结构。

迭代过程照例要先将复数分解为实部和虚部:

$x \to 2x/3+(x\text{^}2-y\text{^}2)/[3(x\text{^}2+y\text{^}2)\text{^}2]$,

$y \to 2y/3-2xy/[3(x\text{^}2+y\text{^}2)\text{^}2]$

以 $f(x)=x\text{^}3-1$ 为例,用牛顿法生成分形图形的一个简单的PASCAL源 程序如下:

```
{NEWTON.PAS, Newton Method 1994-02-02}
Uses Graph,Crt;
Label 30;
Var
i,np,nq,gd,gm,k: integer;
x1,y1,x,y,nx,ny,cm:real;
Function  dist(x,y:real):real; {定义一个距离函数}
BEGIN
dist:=x*x+y*y;
END;
BEGIN
gd:=VGA;   gm:=VGAHI;
InitGraph(gd,gm,'d:\pascal');
```

```
FOR nx:=-160 TO 160  do
FOR ny:=-160 TO 160 do
BEGIN
      x:=nx/80;    y:=ny/80;
      for k:=1 to 30 do
      BEGIN
        if (nx=0) and (ny=0) then goto 30; {排除(0, 0)点}
        cm:=3*dist(x,y)*dist(x,y);
        newx:=2*x/3+(x*x-y*y)/cm;
        newy:=2*y/3-2*x*y/cm;
        x:=newx;  y:=newy;
      END;
      if dist(x-1,y) < 0.03 then putpixel(nx+200,200-ny,2);
      if dist(x+1/2,y-sqrt(3)/2) < 0.03 then
putpixel(nx+200,200-ny,3);
      if dist(x+1/2,y+sqrt(3)/2) <0.03 then
putpixel(nx+200,200-ny,4);
30:
      if KeyPressed then exit;
END;
```

```
CloseGraph;

readln;

END.
```

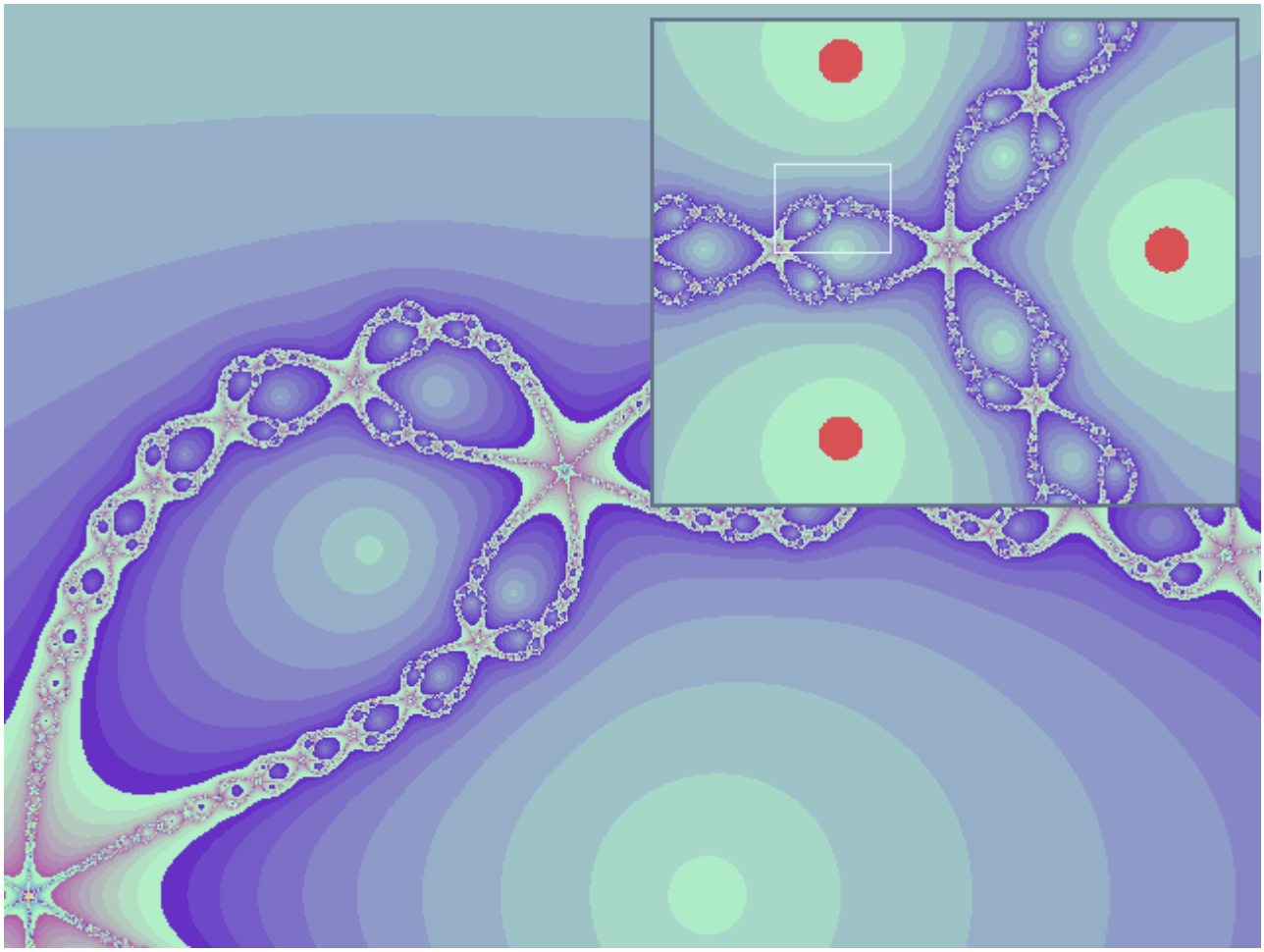
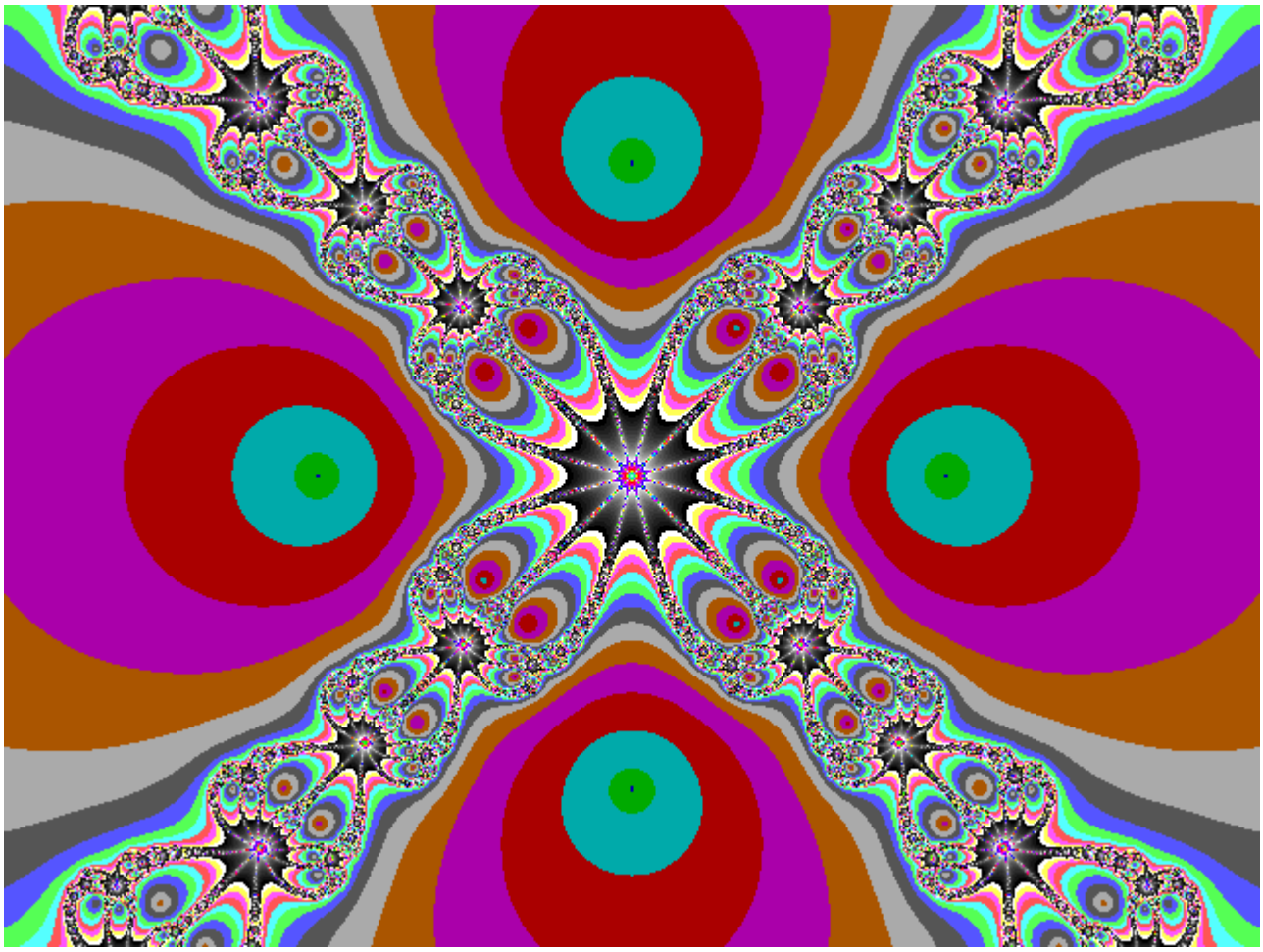


图 7.6.1 用牛顿法求方程 z^3-1=0 的根所得到的“项链”



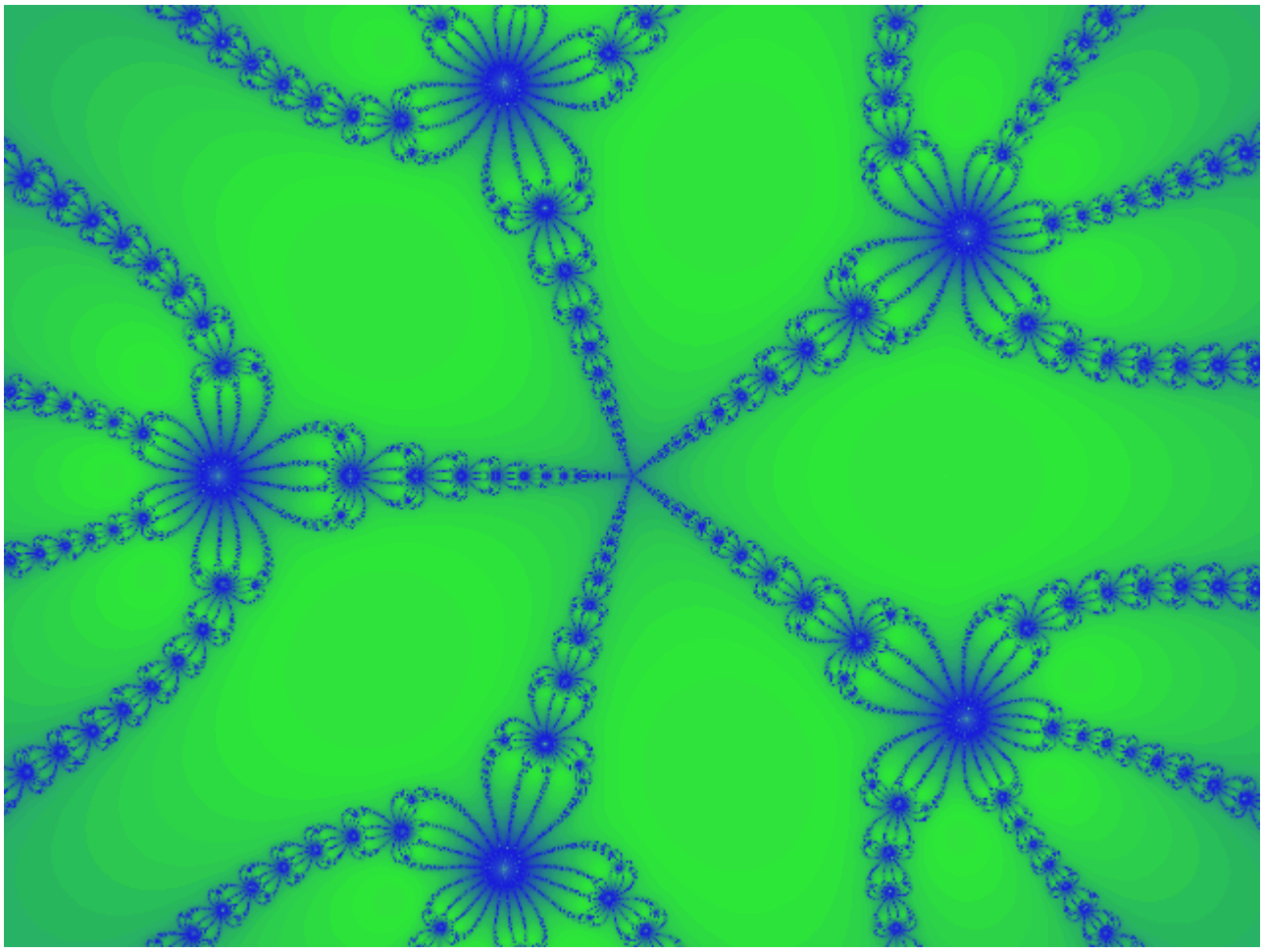


图 7.6.2 用牛顿法求 z^4-1=0 和 z^5-1=0 的根得到的分形图

以上程序采用浮点运算，速度是比较慢的。如果每迭代一次就计算一次距离，则速度会更慢。为了使不同吸引区域有不同的颜色，并且能够表现收敛速度，必须做更多的距离计算，但得到的图形也更美丽。特别是将分界处的“项链”放大来看，会得到精致的结构。

实际上可以把牛顿法扩展到形式为 x^y-r=0 的复迭代，其中 x 和 y 均为复数，r 为实数，也可以为复数。特别应当考虑 y 为纯虚数时的迭代，这时有极漂亮的“编织结构”。

§7.7 发散区域的分类

从前面众多描画的 Julia 集合、Mandelbrot 集合的图形可以看到，在集合外部有许多复杂的条纹花样，这部分区域就是发散区域，这些条纹花样可以根据发散的速度来分类。以本章开始的迭代复数数列为例：$z_n=z_{n-1}^2+z_c$，当 z_n 的绝对值为 2 以上(即 z_n 的绝对值的平方 4 以上)时，该数列是发散的，也就是说，在描画以原点为中心、半径为 2 的圆时，对于某个 n，如果 z_n 跳到圆外，那么它就不会再回到圆内。

以 $z_0=x_0+y_0i$ 为初值，按 $z_1, z_2, \ldots$ 顺序进行计算，使 z_n 跳到圆外，在下面的程序中，对于点(x0,y0)，将赋予由二维数组 col[ ][2]和函数 getcol 指定的颜色号码。数组为

col[ ][2]={{2,1},{4,2},{8,4},{16,3},{32,5},{KL+1,6}}

时，n 和描画颜色之间的关系如下：

1<=n<=2- - 颜色为 1 号蓝色

2<n<=4--- - 颜色为 2 号绿色

4<n<=8 --- 颜色为 4 号红色

8<n<=16- - 颜色为 3 号浅蓝色

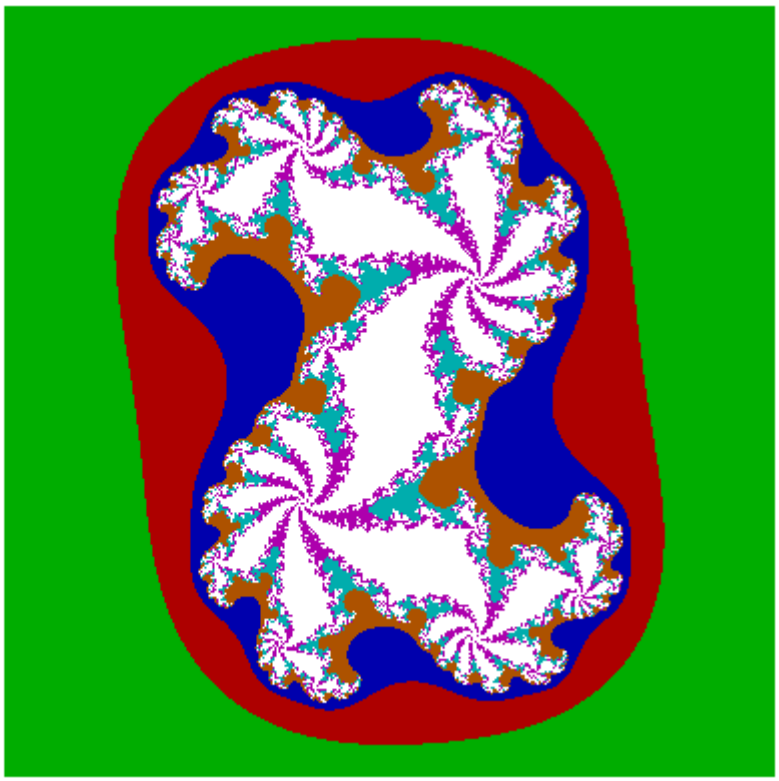
16<n<=32- 颜色为 5 号紫色

32<n<=KL+1 颜色为 6 号黄色

我们规定，不发散时，与前面程序相反，什么也不描画，这样一来就可描画出周围有条纹花样的黑色分形图形。下面我们将前面的描画

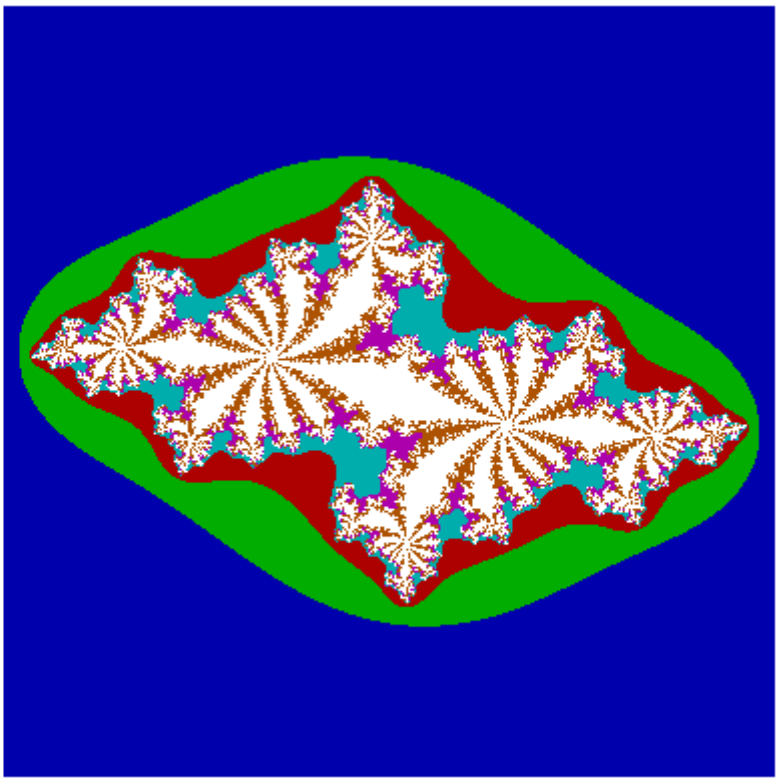
Julia 集合和 Mandelbrot 集合的程序作一个修改，这时，分形集合本身变成了黑色的。图 7.7.1 是按发散区域分类的 Julia 集合，图 7.7.2 是按发散区域分类的 Mandelbrot 集合，其源程序如下：

```
x_min= -1.4
x_max=  1.4
y_min= -1.4
y_max=  1.4

xc= 0.380
yc= 0.150
```



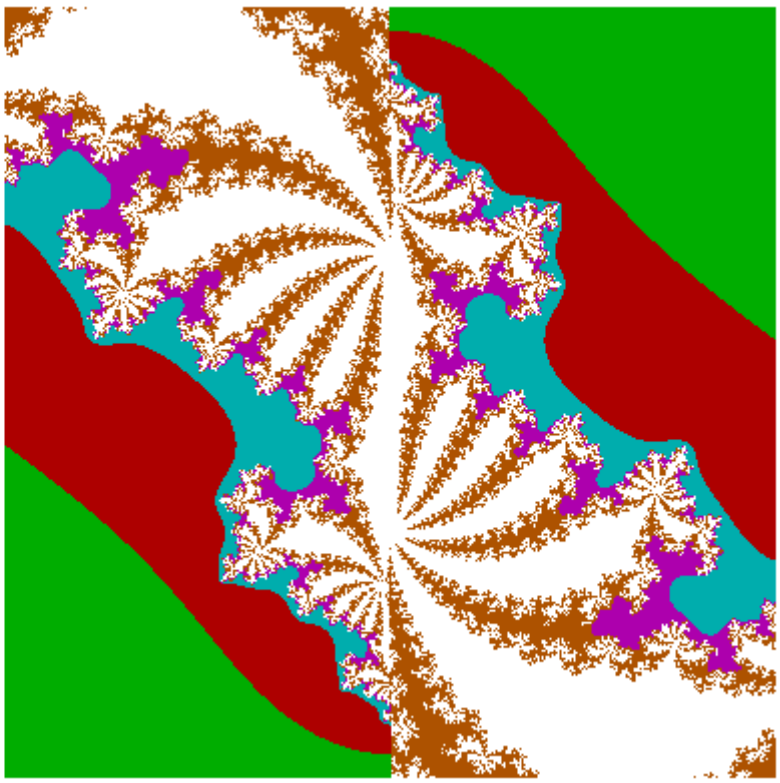
(a)

```
x_min= -1.6
x_max=  1.6
y_min= -1.6
y_max=  1.6

xc=-0.704
yc= 0.280
```



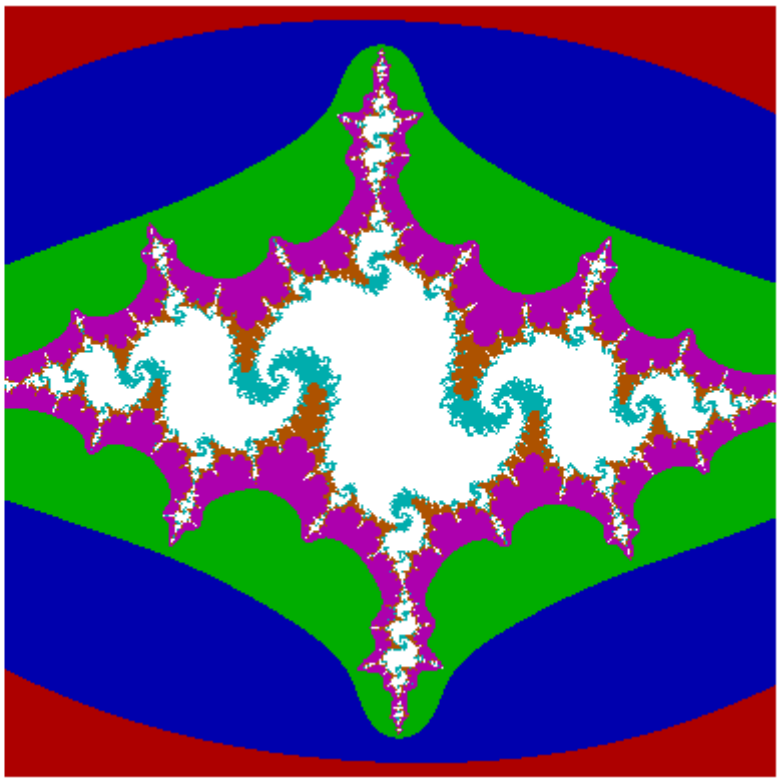
(b)

```
x_min=-0.27
x_max= 0.13
y_min= 0.51
y_max= 0.91

xc=-0.704
yc= 0.280
```



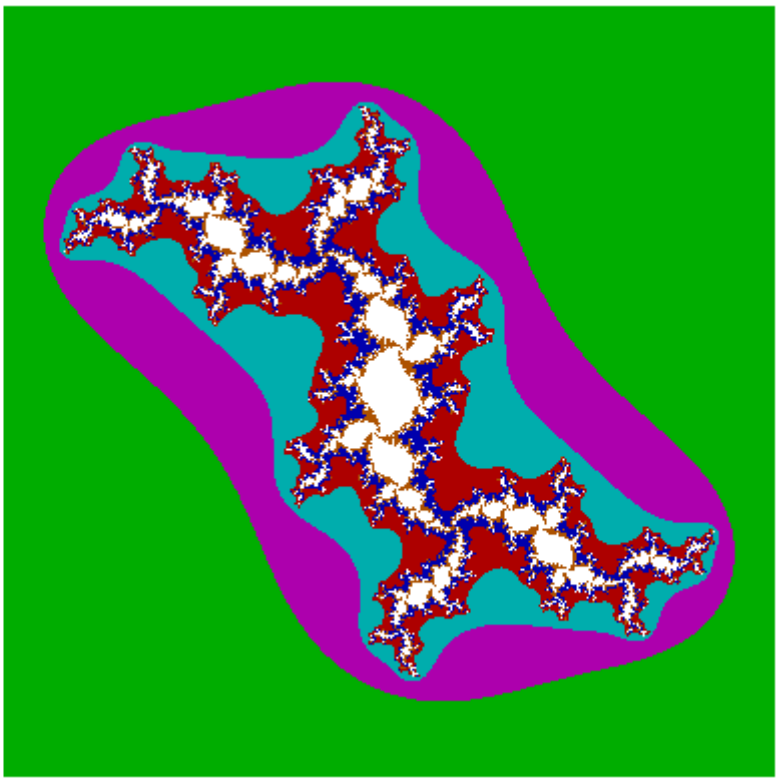
(b')

```
x_min= -0.8
x_max=  0.8
y_min= -0.8
y_max=  0.8

xc=-1.270
yc= 0.040
```



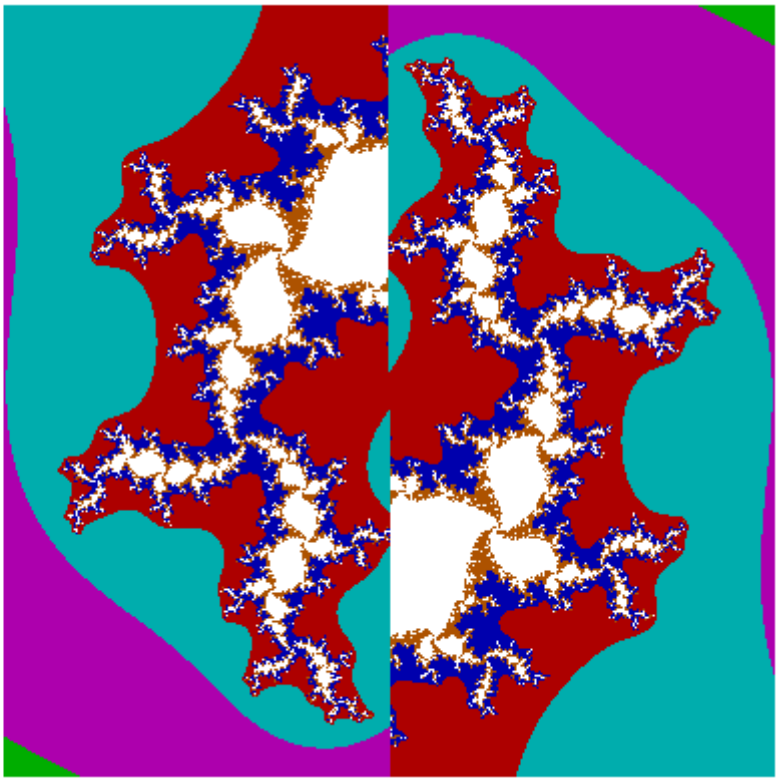
(c)

```
x_min=-1.50
x_max= 1.50
y_min=-1.50
y_max= 1.50

xc=-0.035
yc= 0.795
```



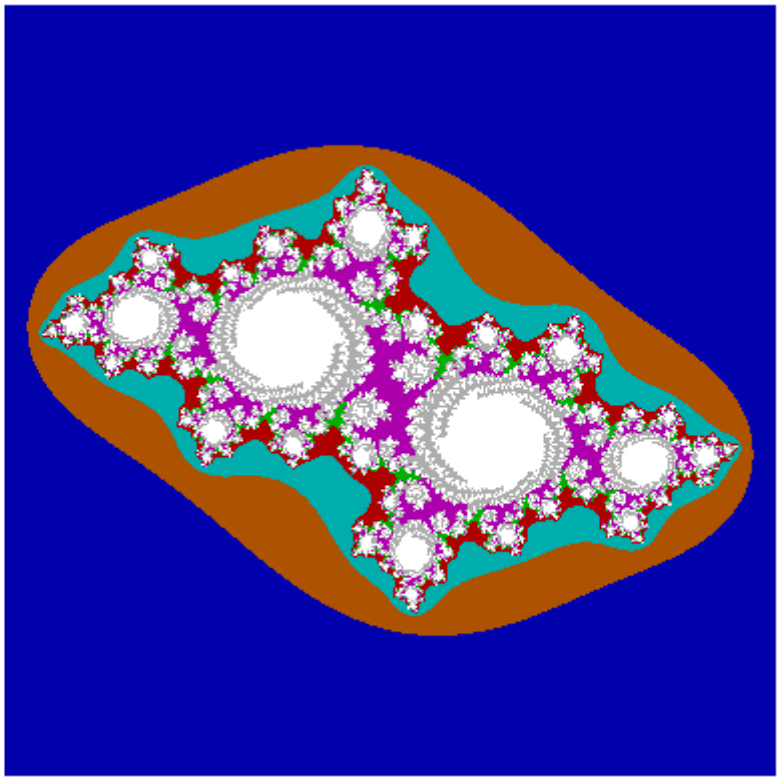
(d)

```
x_min=-0.38
x_max= 0.12
y_min= 0.65
y_max= 1.15

xc=-0.035
yc= 0.795
```



(d')

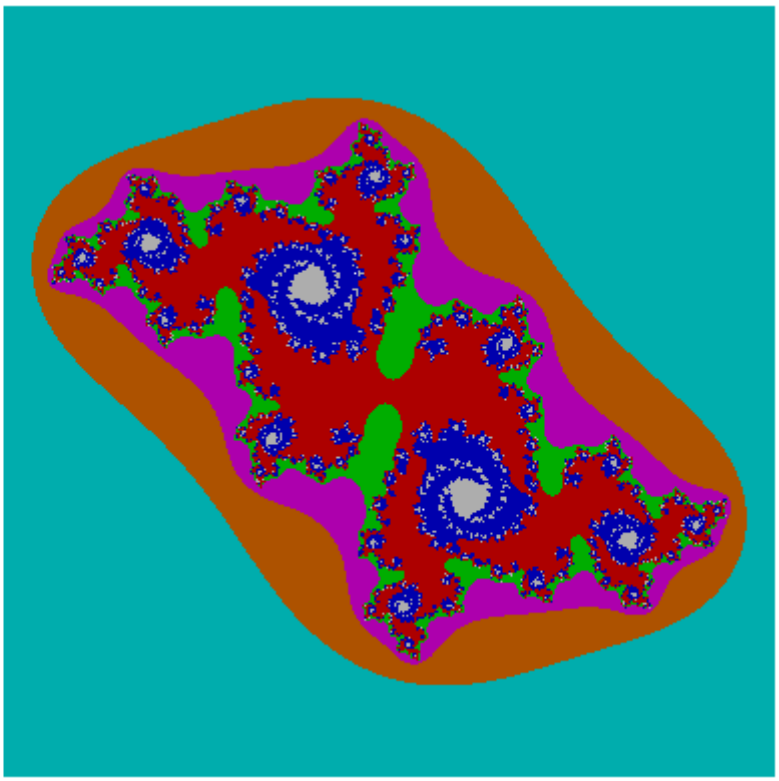
```
x_min=-1.60
x_max= 1.60
y_min=-1.60
y_max= 1.60

xc=-0.600
yc= 0.425
```



(e)

x_min=-1.50
x_max= 1.50
y_min=-1.50
y_max= 1.50

xc=-0.230
yc= 0.670



(f)

图 7.7.1　按发散区域分类的 Julia 集合

(a) /* 描画区域 */

```
double x_min=-1.4,x_max=1.4,y_min=-1.4,y_max=1.4;
double xc=0.38,yc=0.15;/*complex constants*/
double col[][2]={{2,2},{4,4},{8,1},{16,6},{32,3},{KL+1,5}};
```

(b) /* 描画区域 */

```
double x_min=-1.6,x_max=1.6,y_min=-1.6,y_max=1.6;
double xc=-0.704,yc=0.28;/*complex constants*/
double col[][2]={{2,1},{4,2},{8,4},{16,3},{32,5},{KL+1,6}};
```

(b′) /* 描画区域 */

```
double x_min=-0.27,x_max=0.13,y_min=0.51,y_max=0.91;
double xc=-1.27,yc=0.04;/*complex constants*/
double col[][2]={{2,4},{4,1},{8,2},{16,5},{32,6},{KL+1,3}};
(c) /* 描画区域 */
double x_min=-0.76,x_max=0.76,y_min=-0.76,y_max=0.76;
double xc=-1.27,yc=0.04;/*complex constants*/
double col[][2]={{2,4},{4,1},{8,2},{16,5},{32,6},{KL+1,3}};
(d) /* 描画区域 */
double x_min=-1.5,x_max=1.5,y_min=-1.5,y_max=1.5;
double xc=-0.035,yc=0.795;/*complex constants*/
double col[][2]={{2,2},{4,5},{8,3},{16,4},{32,1},{KL+1,6}};
(d′) /* 描画区域 */
double x_min=-0.38,x_max=0.12,y_min=-0.65,y_max=1.15;
(e) /* 描画区域 */
double x_min=-1.6,x_max=1.6,y_min=-1.6,y_max=1.6;
double xc=-0.6,yc=0.425;/*complex constants*/
double
col[][2]={{2,1},{4,6},{8,3},{16,4},{36,2},{81,5},{KL+1,7}};
(f) /* 描画区域 */
double x_min=-1.5,x_max=1.5,y_min=-1.5,y_max=1.5;
```

```
double xc=-0.23,yc=0.67;/*complex constants*/
double
col[][2]={{2,3},{4,6},{8,5},{16,2},{32,4},{64,1},{KL+1,7}};
```

图 7.7.2　按发散区域分类的 Mandelbrot 集合

```
(a) /* 描画区域 */
double x_min=-2.2,x_max=0.6,y_min=-1.4,y_max=1.4;
double col[][2]={{6,4},{8,1},{12,2},{18,5},{KL+1,6}};
(b) /* 描画区域 */
double x_min=-1.8,x_max=-1.74,y_min=-0.03,y_max=0.03;
double col[][2]={{18,4},{24,1},{36,2},{54,5},{KL+1,6}};
©/* 描画区域 */
double x_min=-1.49,x_max=-1.47,y_min=-0.01,y_max=0.01;
double col[][2]={{30,4},{36,1},{48,2},{66,5},{KL+1,6}};
```

§8.1 一维逻辑斯蒂映射

一维逻辑斯蒂(logistic)映射是混沌和分形研究中地地道道的“麻雀”，许多著作中都要讲解，最精彩的还是北大朱照宣教授 1984 年写的讲义《混沌》(非线性力学讲义第五章)。关于逻辑斯蒂映射的数学研究可参见考雷特(P.Collet)和艾克曼(J.-P.Eckmann)1980 年出版

的《区间上作为动力系统的迭代映射》，张景中(1936-)、熊金诚(1938-)1992 年合著的《函数迭代与一维动力系统》，郝柏林著《从抛物线谈起》，以及郑伟谋(1946-)和郝柏林 1994 年合著的《实用符号动力学》。

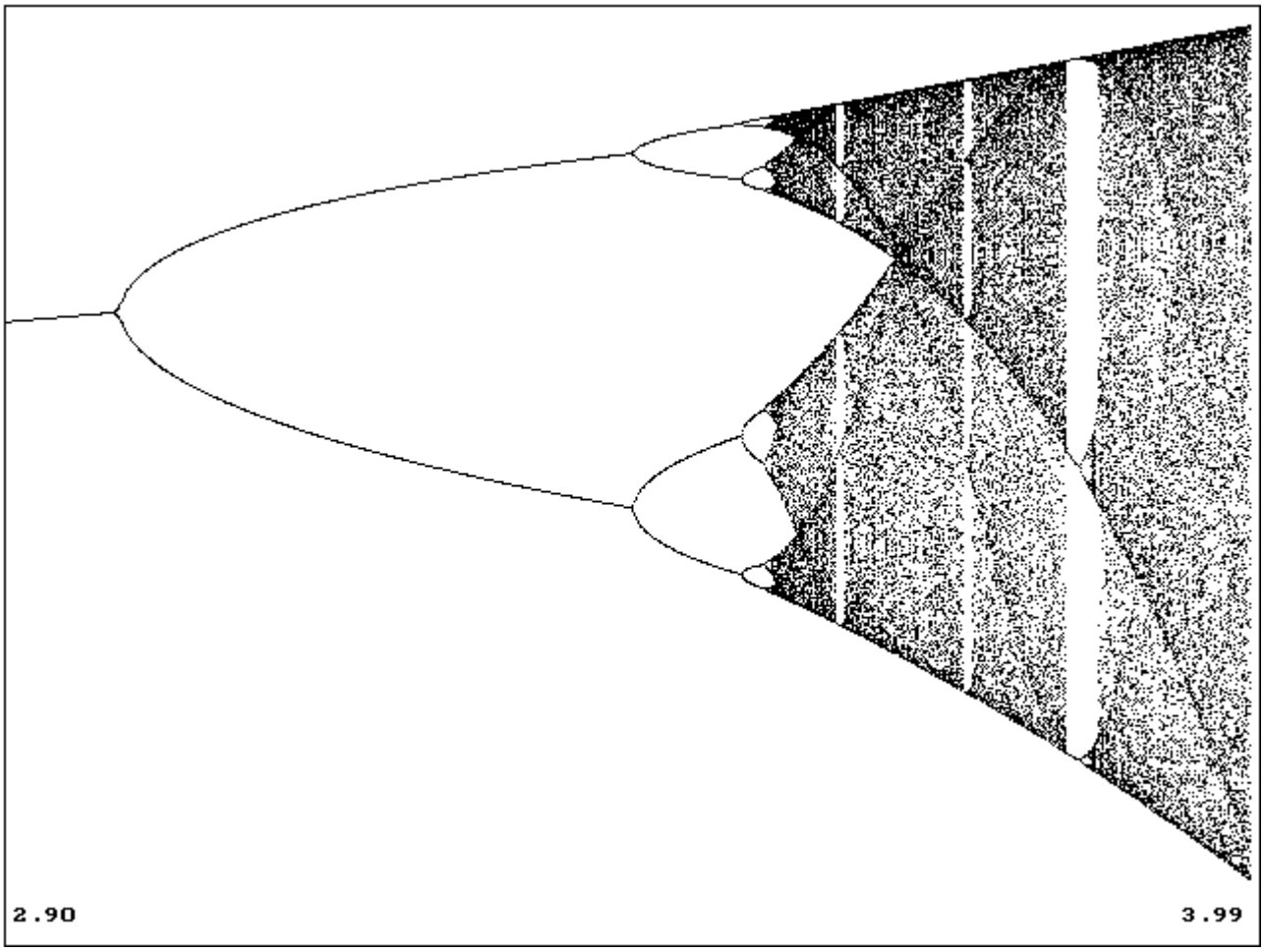


图 8.1.1 逻辑斯蒂映射 $x \to ax(1-x)$ 的分岔图

非线性科学史中，关于一维逻辑斯蒂映射的研究有一系列有趣的典故，涉及一长串著名科学家的名字，如马尔萨斯(T.R.Malthus,1766-1834)、乌拉姆(S.M.Ulam,1909-1984)、萨可夫斯基(A.N.Sarkovskii)、MSS(指三个人)、DGP(指三个人)、费根鲍姆、梅(R.May,1936- ，

robert.may@zoo.ox.ac.uk)、约克 (J.Yorke, 1941-)、李天岩、辛格(D.Singer)等等，限于篇幅不再叙述。下面仅从图形的角度非常粗浅地介绍逻辑斯蒂映射的分岔过程。

映射(mapping)也叫迭代(iteration)，比如 $x_{n+1}=2x_n$，若 $x_1=3$,则 $x_2=6$，$x_3=12$ 等等。从控制系统的角度看，这也叫反馈(feedback)，把输出当作输入，不断滚动。很容易想到，反馈的结果有若干种：发散的、收敛的、周期的等等。但是我们要问一下，一共有多少种可能的运动类型?是否存在既不收敛也不发散，也不周期循环的迭代过程?

回答是肯定的。这一点至关重要，但可惜的是人们最近才普遍认识到有这种运动类型。这说的就是有界非周期运动，它与混沌有关。

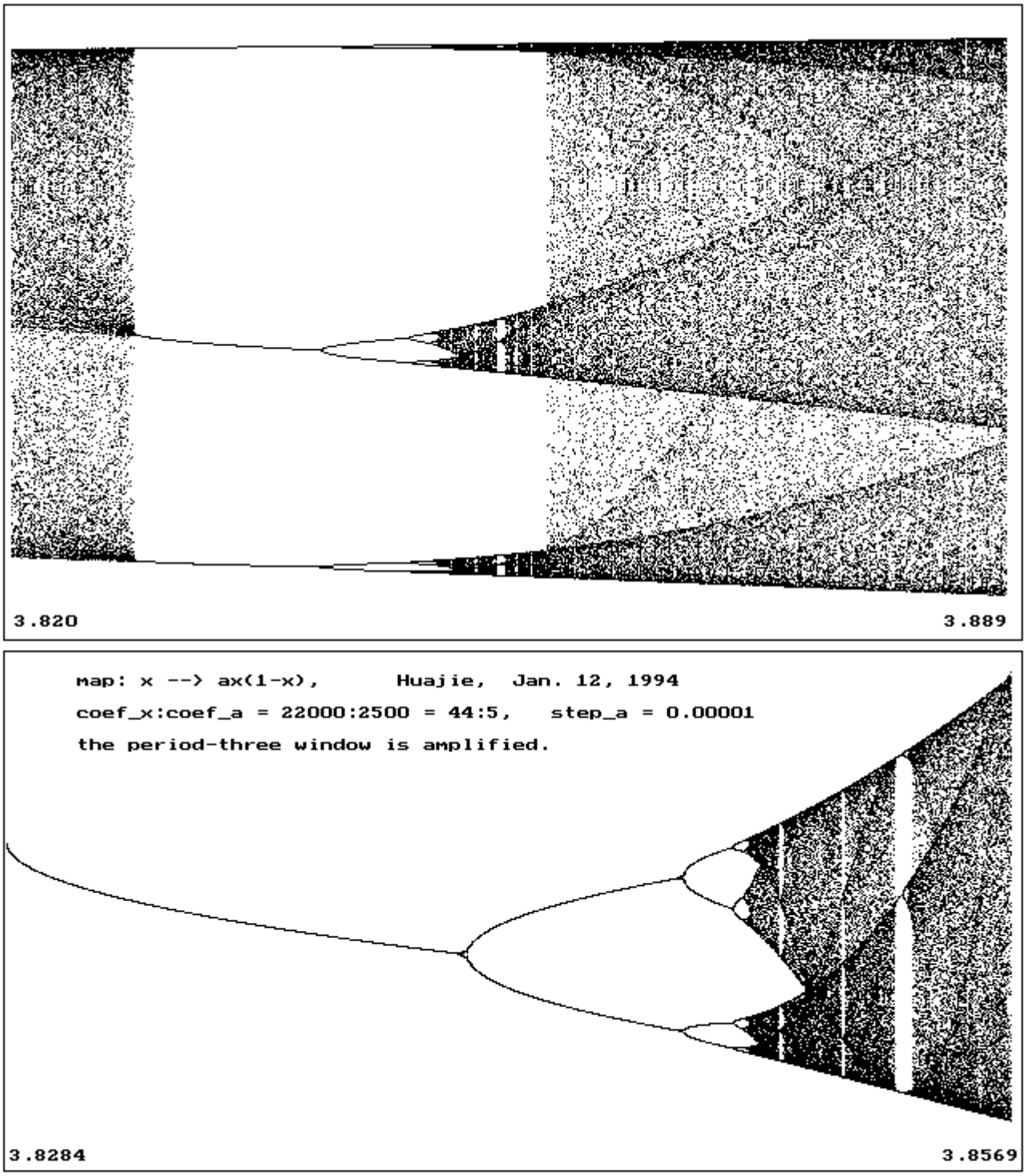


图 8.1.2 周期三窗口的放大图，注意横、纵坐标的比例不同

逻辑斯蒂映射的形式为

$$x\_(n+1) = a x\_n (1 - x\_n),$$

其中 a 是参数，取值范围是 [-2, 4]，通常人们只注意 [0, 4] 这一半，其实另一半 [-2, 0] 也一样有趣。x 的取值为 [0, 1]。映射的不动点是指满足关系 $\xi = a\xi(1-\xi)$ 的点 ξ，解得 $\xi_1=0$，$\xi_2=1-1/a$。设映射用 f 表示，f 的 2 次迭代记作 f^2，3 次迭代记作 f^3，等等 。注意，这种记法不表示乘方关系。f 的不动点也叫 f 的周期 1 点。f^2 的不动点实际上是 f 的周期 2 点。同理 f^n 的不动点与 f 的周期 n 点是一回事。映射 f 的周期 m 点的稳定性由乘子

$$\lambda = | df^m/dx | = | f'(x_1) f'(x_2) f' \cdots f'(x_m) | = | \prod^m_{i=1} f'(x_i) |$$

完全决定。映射 f 的周期点(包括不动点，它为周期 1 点)的稳定性可具体定义为:

$|\lambda| < 1$，吸引，稳定;

$|\lambda| > 1$，排斥，不稳定;

$|\lambda| = 1$，中性;

$\lambda = 0$，超稳定。

以参数a为横坐标、以x的稳定定态(stable steady states)为纵坐标作图， 得到图 8.1.1、图 8.1.2 等。从图中可以看出开始是周期加倍分岔(也称周期倍化分岔或周期倍分岔)，然后是混沌，混沌区中又有周期窗口。窗口放大后又可见到同样结构的一套东西。此所谓无穷自相似结构。其相应PAS程序为p8-1-1

更为有趣的是，不但对于上述形式的映射有这种分岔结构，映射取如下形式

$x_{n+1}=1-\lambda x_n^2$,

$x_{n+1}=\mu \sin(\pi x_n)$,

$x_{n+1}=x_n \exp[\delta(1-x_n)]$

时，仍然可以得到相似的结构，这叫做结构普适性。从图中看到，一维映射不断发生周期倍化分岔，比如存在 $2, 2^2, 2^3, 2^4, \cdots, 2^{\infty}$ 及 $3\times 2, 3\times 2^2, 3\times 2^3, \cdots, 3\times 2^{\infty}$ 等周期加倍过程。早在 60 年代苏联就有一位杰出的数学家将所有可能的周期轨道进行了排序，这人便是萨可夫斯基，1995 年他曾到中国进行学术访问。萨可夫斯基序列中第一个是周期 3，然后是周期 5，最后是周期 4、周期 2 和周期 1，中间有无穷多别的周期。

曾在洛斯阿拉莫斯国立实验室任职的费根鲍姆在研究周期倍化过程中，发现相邻分岔间距之比收敛到一个不变的常数:

$$\delta=\lim_{n\to\infty} |[a_n-a_{n-1}]/[a_{n+1}-a_n]|$$

$$=4.669,201,609\cdots$$

不仅仅对于逻辑斯蒂映射有这个常数，对于一维“单峰”映射，都能算出同一个常数 δ 来。这件事很重要，令非线性科学界为之一震，后来曾有人为此提议给费根鲍姆授诺贝尔奖 ，但未成功。原因不详，作者猜大概是: 第一，严格说来这项成果属于数学而非物理科学，而诺贝尔奖从不授予数学学科; 第二, 人们还未彻底搞清 δ 的含义、意义; 第三,

此项建议在讨论过程中首先遭到非线性科学内部权威人士的激烈反对。此次未获奖，不等于以后不能获奖，不过麻烦的是，非线性科学研究是集体奋斗的历史，我们一口气可以说出 10 个、20 个杰出科学家，但找出一个或者两个最杰出者，却让人犯难。如果把非线性科学与相对论、量子力学相比，这也是一个极大的区别。也许一人或者几人独创一门新科学(如相对论)的时代一去不复返了。

一维映射除了 δ 外还有其他度量普适性，如标度因子 α 等，可参看其他著作 。

§8.2 里雅普诺夫指数

确定混沌运动一般有分数维数、里雅普诺夫(A.M.Liapnov，1857-1918)指数、拓扑熵、功率谱等几个定量指标，本节谈里雅普诺夫指数。

一般系统的里雅普诺夫指数并不很容易求出，但对于一维逻辑斯蒂映射，可以给出解析式，便于数值计算。设 λ 是里雅普诺夫指数，对于映射 $x\to ax(1- x)$，

$f'(x)=\mathrm{d}f/\mathrm{d}x=a-2ax$，则

$$\lambda=\lim_{N\to\infty}\{1/N\sum\nolimits_{i=1}^{N-1}\ln|\mathrm{d}f/\mathrm{d}x|\}$$

$$=\lim_{N\to\infty}\{1/N\sum\nolimits_{i=1}^{N-1}\ln|a-2ax|.$$

在实际计算中N取足够大即可，可以取 500 或者 2000。计算里雅普诺夫指数 λ 的源程序为p8-2-1

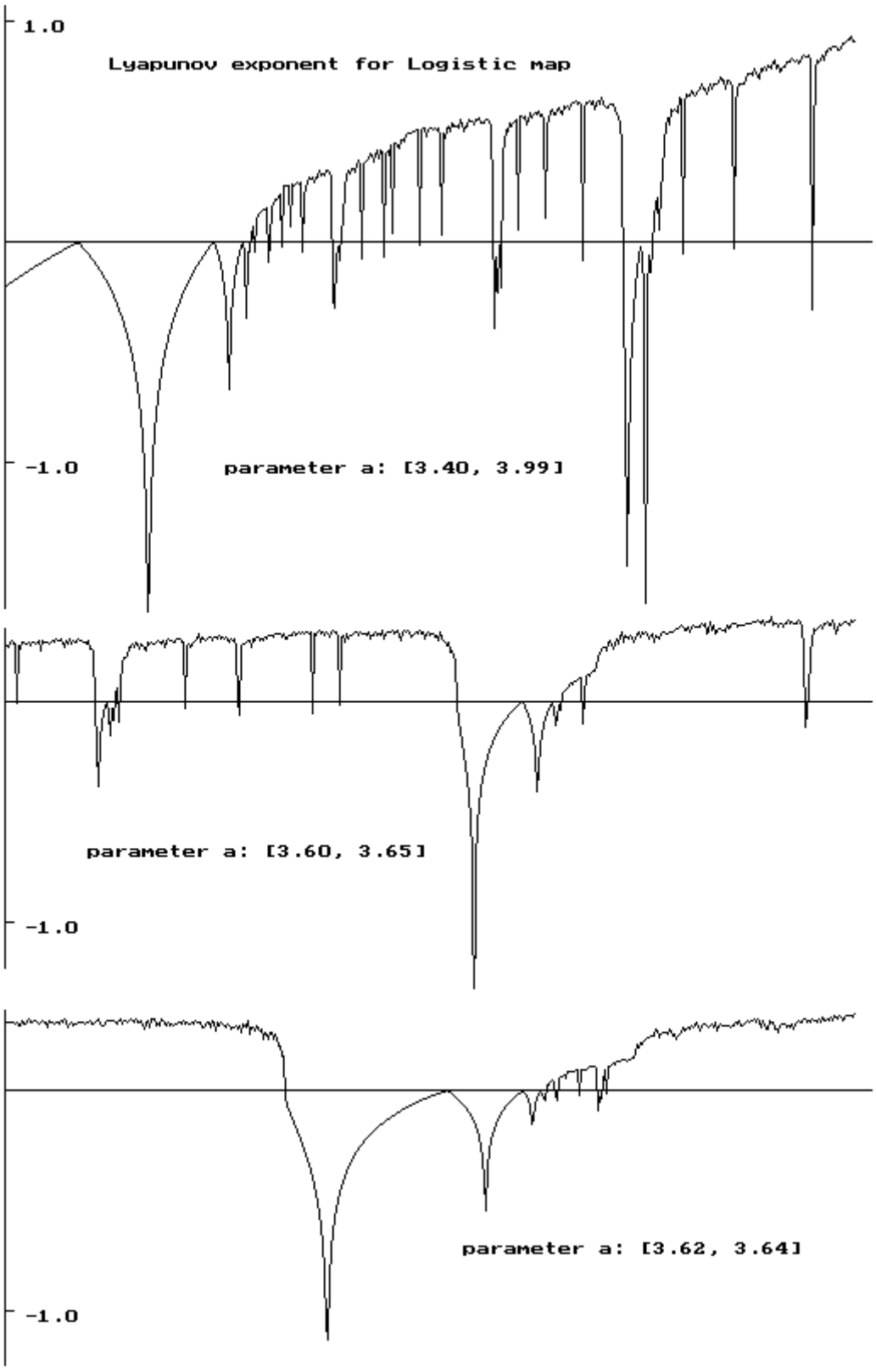


图 8.2.1 一维逻辑斯蒂映射里雅普诺夫指数谱

当 λ 大于零时，一般认为系统处于混沌运动，当 λ 小于零时，认为系统处于周期运动或者准周期运动。图 8.2.1 中展示的是参数 $a\in$ [3.40,3.99] 区间上里雅普诺夫指数的情况，中图和下图示意了参数 $a\in$ [3.60,3.65] 和 $a\in$ [3.62,3.64] 区间的情况，横坐标不断变大，纵坐标未变。

§8.3 双混沌映射

现在我们已经知道，简单的逻辑斯蒂映射竟有极复杂的行为，以参数 a 和定态 x ^*作图(a, x^*)发现了各层次的周期窗口。特别当映射处于混沌区时，相点具有遍历(ergodic)甚至混合(mixing)行为，当 a 接近 4 时，相点差不多可以充满整个相空间(0,1)。

既然逻辑斯蒂映射在迭代过程中具有复杂行为，那么可否把它作为一种“数据源”——类似准随机数发生器——来使用呢?回答是肯定的。在混沌理论和分形理论研究中，特别是混沌控制研究中，人们经常这么做，即将一个方程的输出作为其他系统的输入信号。

中学时我们学过极坐标与直角坐标的转化，设极坐标用(ρ，θ)表示，直角坐标用(x, y)表示，则 $x=\rho\sin\theta$, $y=\rho\cos\theta$。现在考虑用逻辑斯蒂映射的迭代输出作为输入，不断代换其中的 ρ 和 θ，计算出 x 和 y 并将它们画到平面上去。

先作一些定性分析，因为当 $0<a<4$ 时，逻辑斯蒂映射是有界的，值域是（0,1），所以 ρ 也是有界的，值域也是(0，1)。又因为正弦、余弦函数值域是[-1,1]，所 以 x 和 y 一定处于单位圆内。

我们取两组逻辑斯蒂映射: $x\rightarrow ax(1-x)$ 和 $y\rightarrow a\ y(1-y)$,做如下操作

(1) $x\_(n+1)=ax\_n(1-x\_n)$，{a 为参数}

(2) $y\_(n+1)=ay\_n(1-y\_n)$,

(3) $R=x\_(n+1)+y\_(n+1)$,

(4) $T=x\_n+y\_n$,

(5) $M=R\sin(KT\pi)$，{K 为参数}

(6) $N=R\cos(KT\pi)$,

(7) $x\_n=x\_(n+1)$,

(8) $y\_n=y\_(n+1)$

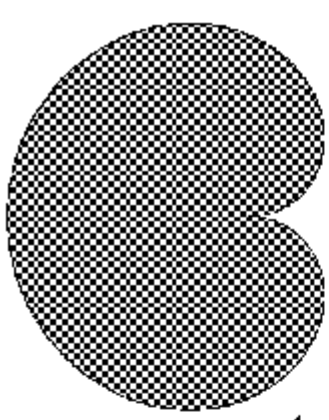
为了避免混乱，符号重新作了安排。上面的操作中的(1)和(2)式提供数据源，(3)和(4)式对数据源的数据作简单的四则运算(加、减、乘、除均可，并且可以乘上某个系数，上面只给出相加的情况)，(5)和(6)式则对数据作最后处理，然后对(M，N)描点作图,(7)和(8)式表示迭代。实际上上述步骤可以简化为

$M=f(x\_(n+1),y\_(n+1))\sin[g(x\_n,y\_n)\ K\pi]$

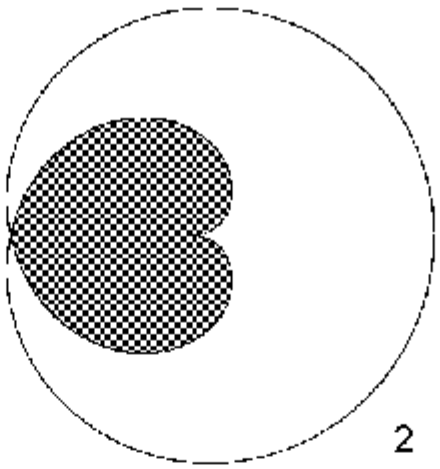
$N=f(x\_(n+1),y\_(n+1))\cos[g(x\_n,y\_n)\ K\pi]$

函数 f 和 g 可以取各种形式，两者的位置也可以交换。以上还只是对一个参数值 a 而言的，可以变化参数 a，把整个计算过程都记录下来，于

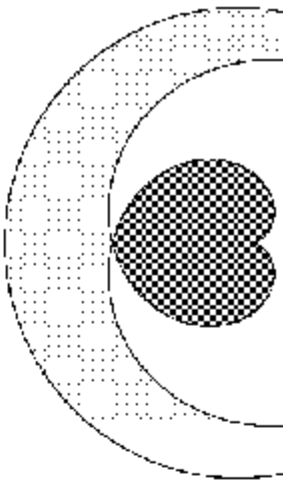
是得到各种各样美丽的图形。本小节的标题称“双混沌映射”(double chaotic maps)，是因为作图的原始数据来源于两个可产生混沌运动的逻辑斯蒂映射。

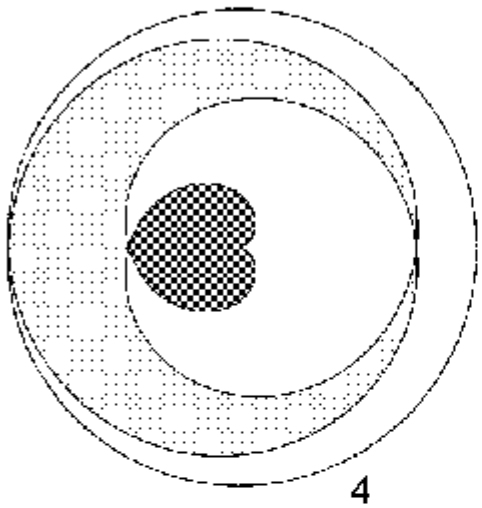


1

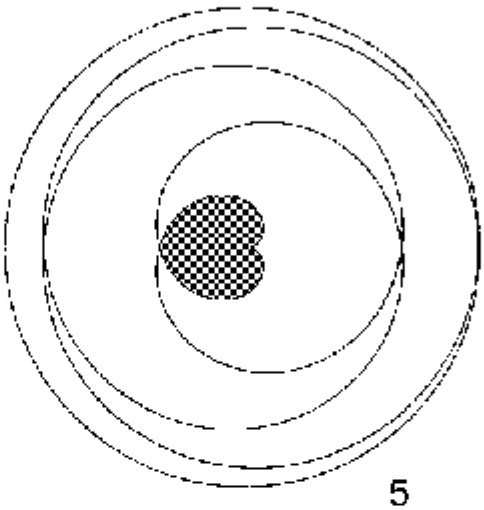


2

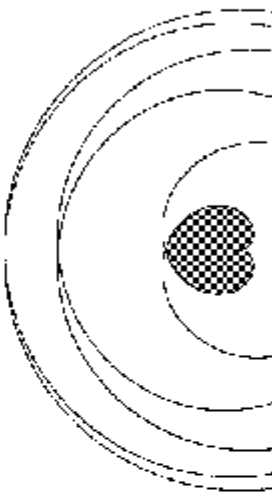


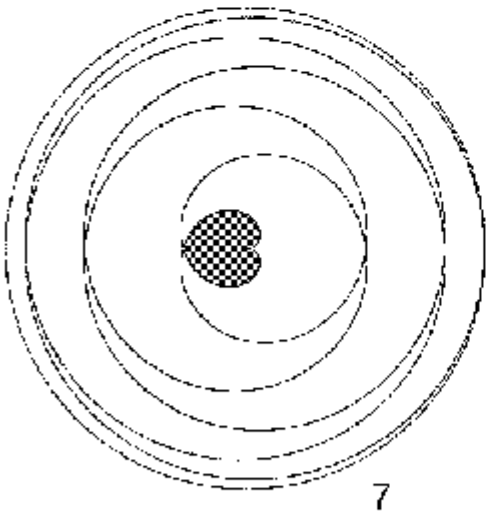


4

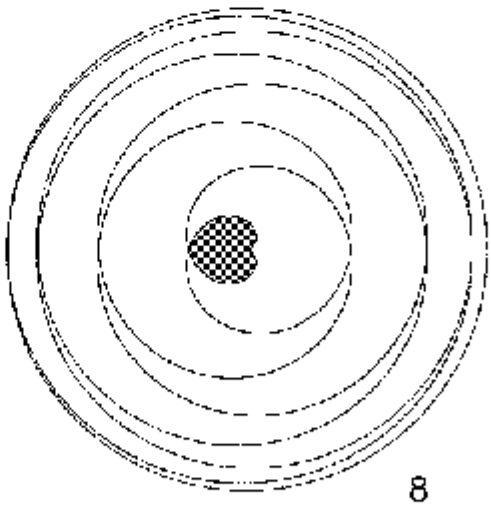


5

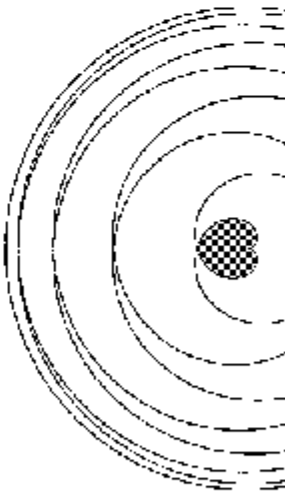




7



8



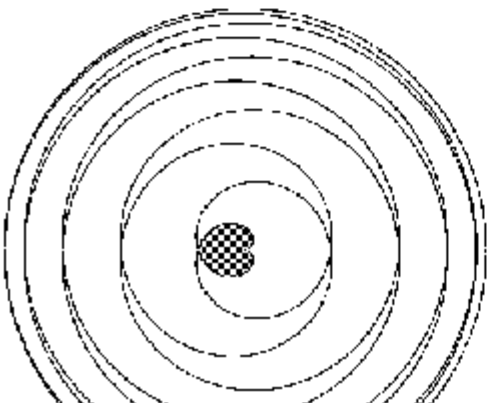


图 8.3.1 参数 K 取不同正整数时得到的图形(试验一)

参数 K 的取值很重要，可以尝试取一般的整数(如 1,2,3,12,23)，也可以取小数甚至负数。

至此，还有几个小问题没有交待。对于(1)和(2)式，迭代初始值 x_0 和 y_0 可以取相同的值，也可以取不同值，效果大不一样。另外，(1)和(2)式中参数也未必都取 a，也可以取不同的值。

现在所有技巧都说完了，能否作出好看的图形，全在于函数、参数的搭配和你的运气了。别急，我们会告诉一些有趣的取值，据此你可以举一反三，作出更美的图形。

```
{SunLiu.PAS}
uses Graph,Dos,Crt;
var
    x,y,x1,x2,xx,yy,y1,y2,bigx,bigy,n,a:real;
Gd,Gm,ErrorCode,i:integer;
begin
  TextColor(RED);
  write('Input parameter K=');
  readln(K);   {1,2,3,4,5,6,7,8,9,17,50,3.25,-3.75}
  Gd:=Detect;InitGraph(Gd,Gm,'D:\PASCAL');
```

```
  ErrorCode:=GraphResult;
  if ErrorCode<>grOK then begin
         Writeln('Graphics
error:',GraphErrorMsg(ErrorCode));
         Writeln('Program aborted...');
         Halt(1);
         end;
  x:=0.2;y:=0.2;a:=3.988;
    {if x equals y, the line is clear.}
  while a<= 3.999 do begin
  for i:=1 to 460 do begin
         xx:=a*x*(1-x);yy:=a*y*(1-y);
         x1:=xx+yy;y1:=x+y;
         bigx:=x1*cos(y1*K*Pi);
         bigy:=x1*sin(y1*K*Pi); {or x1 instead of y1}

PutPixel(Round(bigx*100)+300,225-Round(bigy*100),15);
         x:=xx;y:=yy;
```

```
            end;
        a:=a+0.0001;
        if KeyPressed then Exit;
        end;
    sound(200);delay(200);nosound;readln;
    CloseGraph;
end.
```

我们利用上面的算法和程序做几种试验:

1)试验一固定参数 $a\in[3.988,3.999]$，改变 n 的值，使 K 取不同的 正整数 $n\in\{1,2,3,\cdots\}$。参见图 8.3.1，除一个奇点外，从图形内部到外部要 K 次经过弧线。当 K 取正整数时，中心的“心形”的尖端一直指向正左侧。

2)试验二 K 取小数和负数时的情况，a 取值同上。在图 8.3.2 中，K 分别 取 3.25，3.50，3.75，-3.25，-3.50 和-3.75，可以明显看到中心的“心形”的尖端不断转 动。为清楚起见，有的区域涂上了黑色。

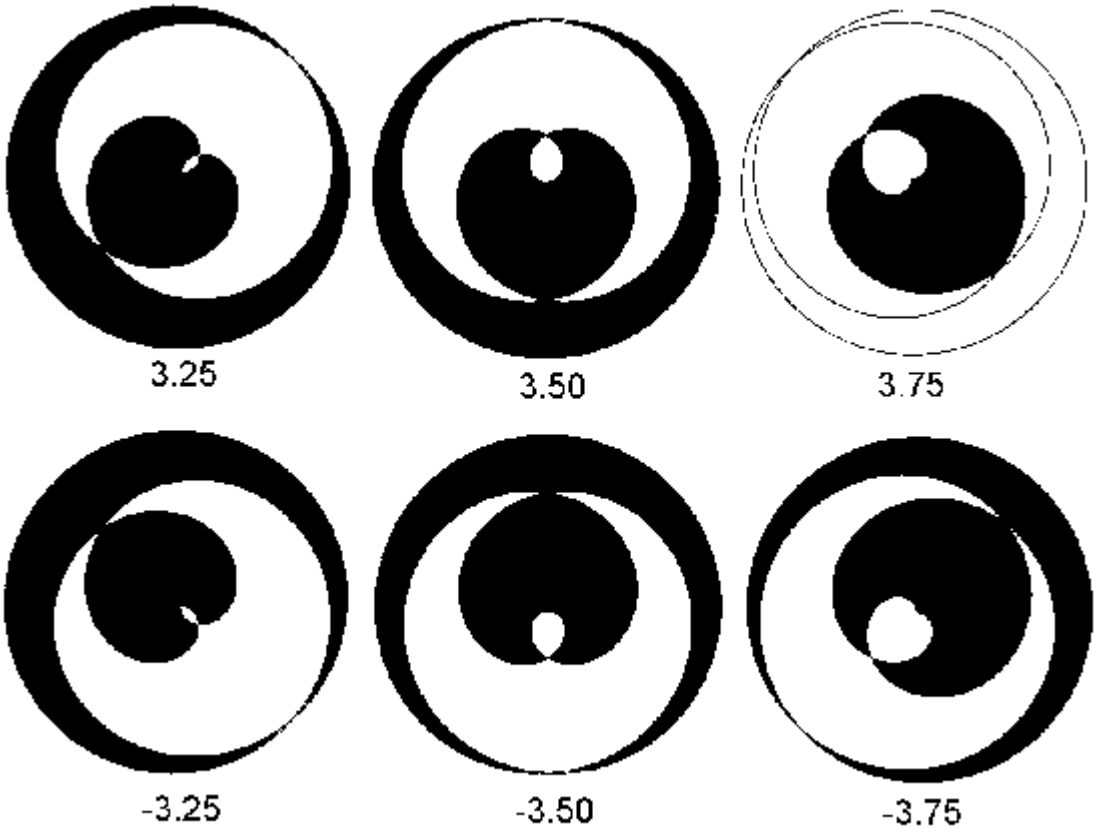


图 8.3.2 参数 K 取小数和负数时得到的图形(试验二)

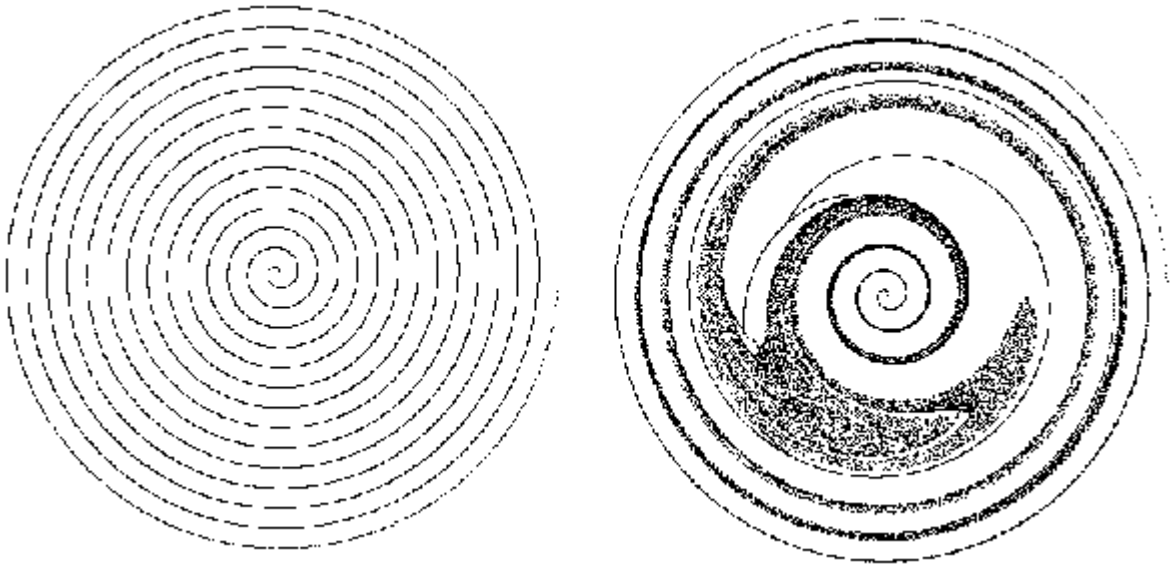


图 8.3.3 左图为试验三、右图为试验四得到的图形，细心观察会发现右图与图 8.1 是一样的， 只是表示方式不同而已。

3) 试验三将 *M*和 *N*形式改变一下：*M*=(*x*_n+*y*_n) sin [(*x*_n+*y*_n) *K* π]，*N*=(*x*_n+*y*_n) cos [(*x*_n+*y*_n) *K* π]，*K*取值范围为 [2.3, 3.999]，这样会得到等距螺线, 见图 8.6 左。

4) 试验四将试验三中 *M*和 *N*表达式三角函数括号里的(*x*_n+ *y*_n)换成(*x*_(n+1)+*y*_(n+1))，得到图 8.3.3 右。

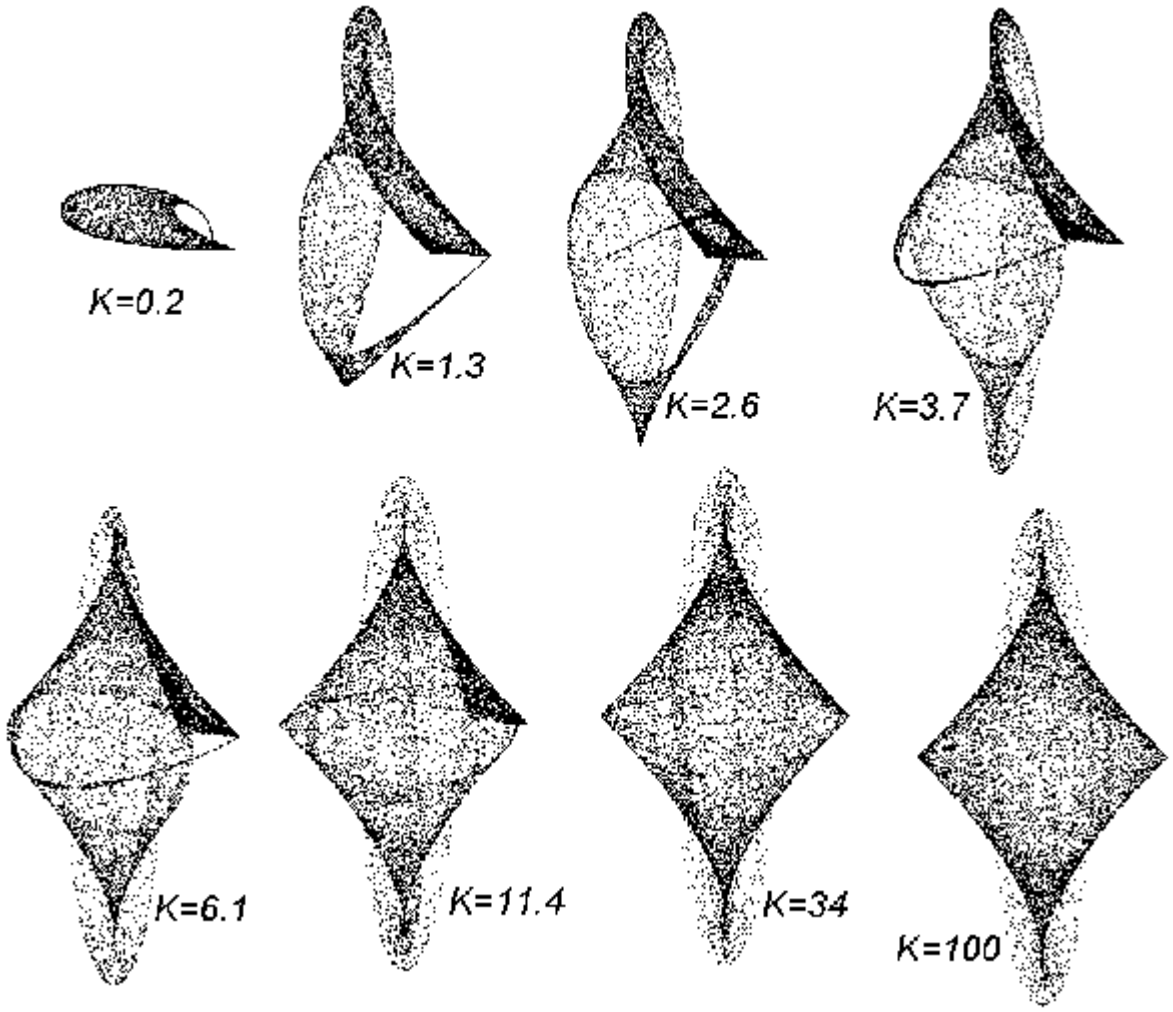


图 8.3.4 具有空间深度的双混沌迭代(试验五)

5)试验五将上面程序中有关语句改为“y1=x*y，

bigx:=cos(x1)cos(y1*K*π)”，其他不变 ，变化参数 K 值，将得到图 8.3.4。

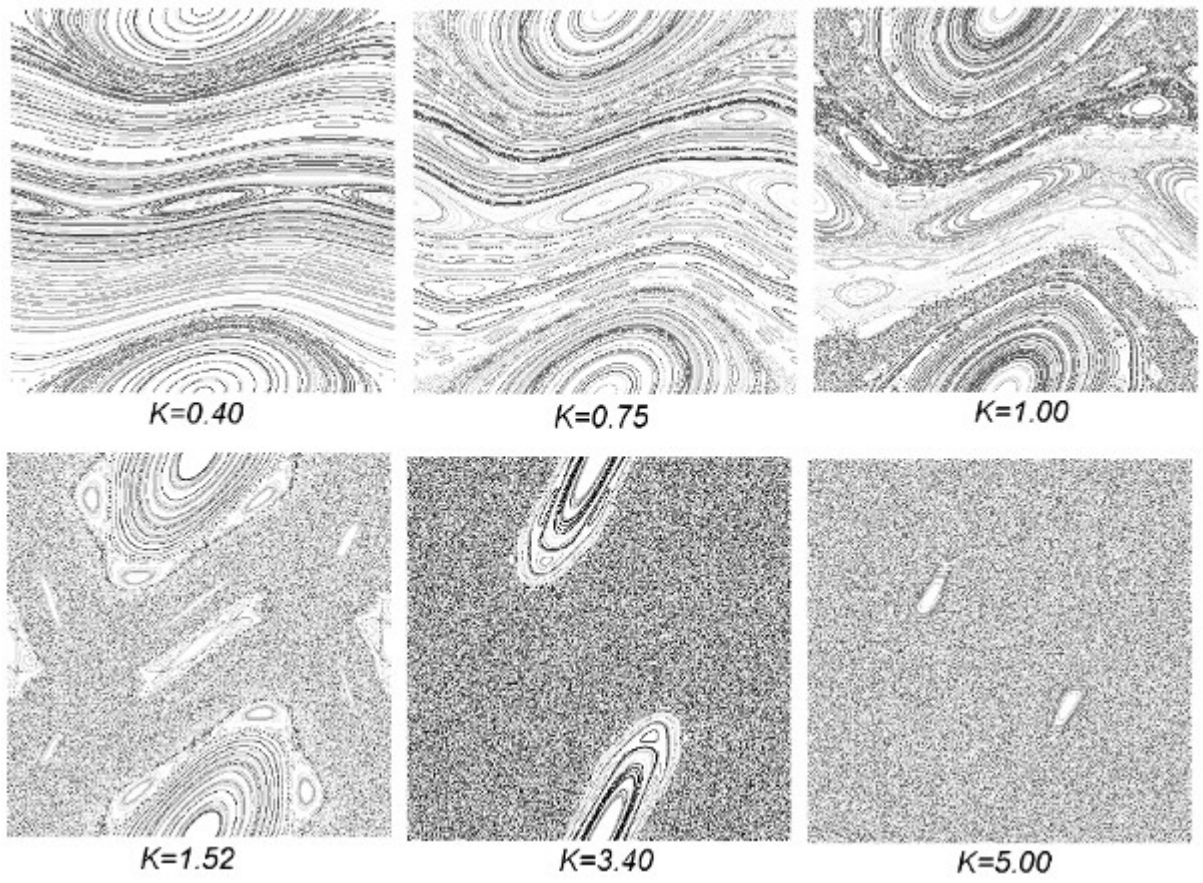
§8.4 标准映射

这里将介绍一种最简单的二维映射——标准映射(standard map)。二维标准映射是一种保守的离散系统，它对于理解三维连续系统的复杂行为很有帮助。可以形象地说，N-1 维的离散映射简单地刻画了相应的 N 维连续流(flow)。

标准映射的一般形式为

$$x\_(n+1)=x\_n+K\sin y\_n,$$

$$y\_(n+1)=y\_n+x\_(n+1),$$


K=0.40
K=0.75
K=1.00
K=1.52
K=3.40
K=5.00

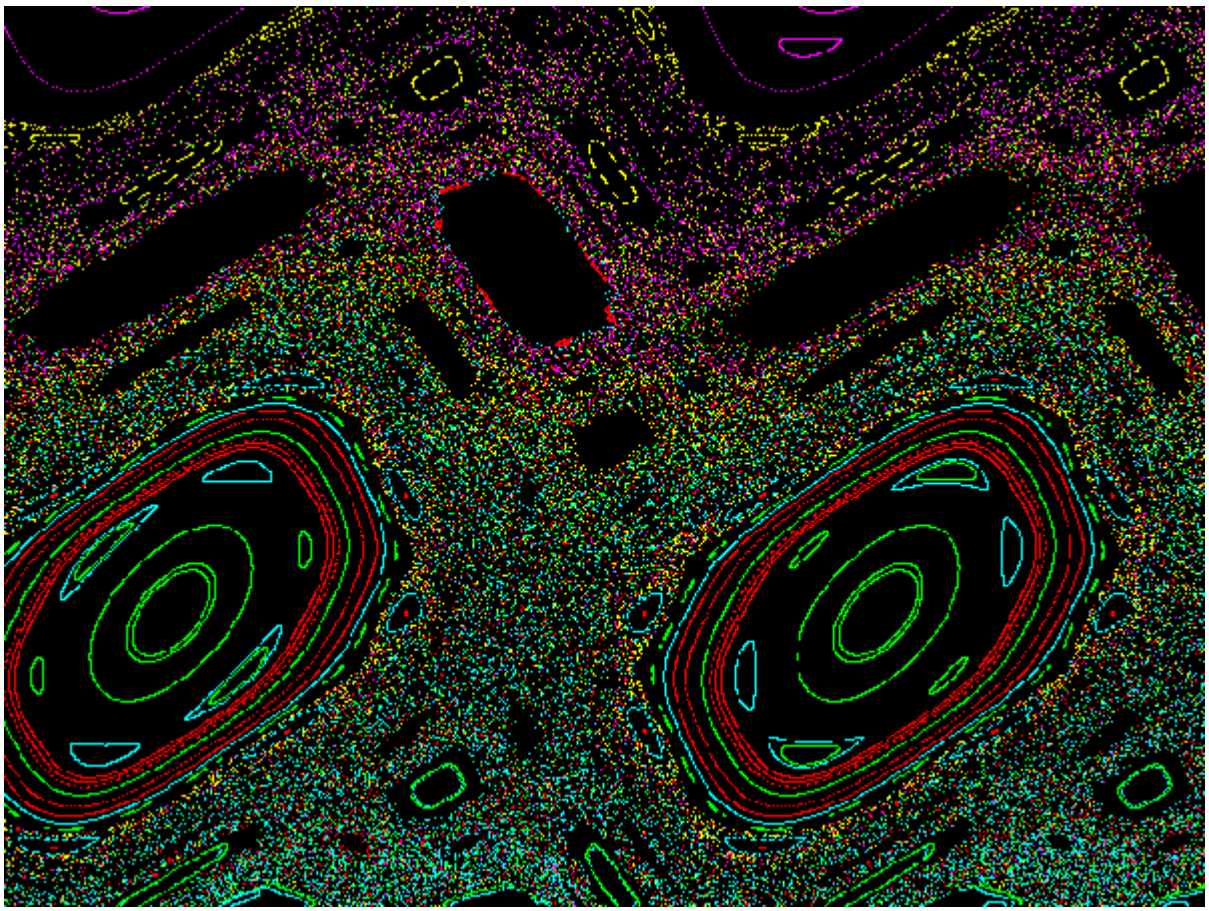


图 8.4.1 *R*取定为 2，改变 *K*得到的标准映射图

此映射的雅可比(C.G.J.Jacobi,1804-1851)矩阵为

J=(偏 *f*/偏 *x*,偏 *f*/偏 *y*;偏 *g*/偏 *x*,偏 g/偏 *y*)_(*P*=*P*^*)

=(1,*K*cos*y*;1,1+*K*cos*y*)_(*P*=*P*^*)

其雅可比行列式为｜*J*｜=1，这说明此系统是保面积的，“保守”一词由此而来。上 式中的 *P*^*表示映射的不动点，*P*^*=(*x*^*,*y*^*)=(2π*m*,0)或者(2π*m*,π)，其中 *m*=0，±1，±2，…。不动点有两类：一类是椭圆不动点，一类是双曲不动点。

```
Program StandardMap;
uses Graph,Crt,dos;
var c,d,i,j,class:integer;
color,backcolor:word;
K,R,coef1,coef2,xs,ys,xe,ye,expn:real;
Gm,Gd,ErrorCode,x,y:integer;
label 10;
function rMOD(y,x:real):real;
begin
```

```
    if (y>x) AND (y>0) thenrepeaty:=y-x;until y
0;
    rmod:=y;
end;
begin
    ClrScr; writeln('Input parameter K=');
    readln(K); class:=4;
    Gd:=Vga; Gm:=VgaHi; InitGraph(Gd,Gm,'d:\pascal');
    if GraphResult<>grOK then  Halt(1);
    coef1:=100/PI;
    for d:=1 to class*6 do
    begin
        ys:=d/2;
        for c:=1 to class*6 do
            begin
              xs:=c/2;
              for i:=1 to 100 do
                  begin
```

```
                    ye:=ys+K*sin(xs); xe:=xs+ye;
                    xs:=xe; ys:=ye
                end;
            for j:=1 to 1000 do
                begin
                    ye:=ys+K*sin(xs); xe:=xs+ye;
                    xe:= rMOD(xe,R*PI); ye:=
rMOD(ye,R*PI);
                    x:=round(70*xe*2/R);
y:=round(70*ye*2/R);
                    if (x>0) and (x<650) then
                    if (y>0) and (y<479) then
                    putpixel(x+50,470-y,d);
                    xs:=xe; ys:=ye;
                    if keypressed then GOTO 10;
                end;
            end;
    end;
```

```
10:
    sound(500);
    delay(200);nosound;
    readln;
    closeGraph;
end.
```

注意，在上述程序中横坐标和纵坐标交换了一下。取 R 为 2，改变参数 K，得到图 8.4.1，可以看到当 K 变大时系统容易出现混沌。当固定 K=1.2，改变 R 时，会出现什么现象呢?这时得到图 8.4.2。

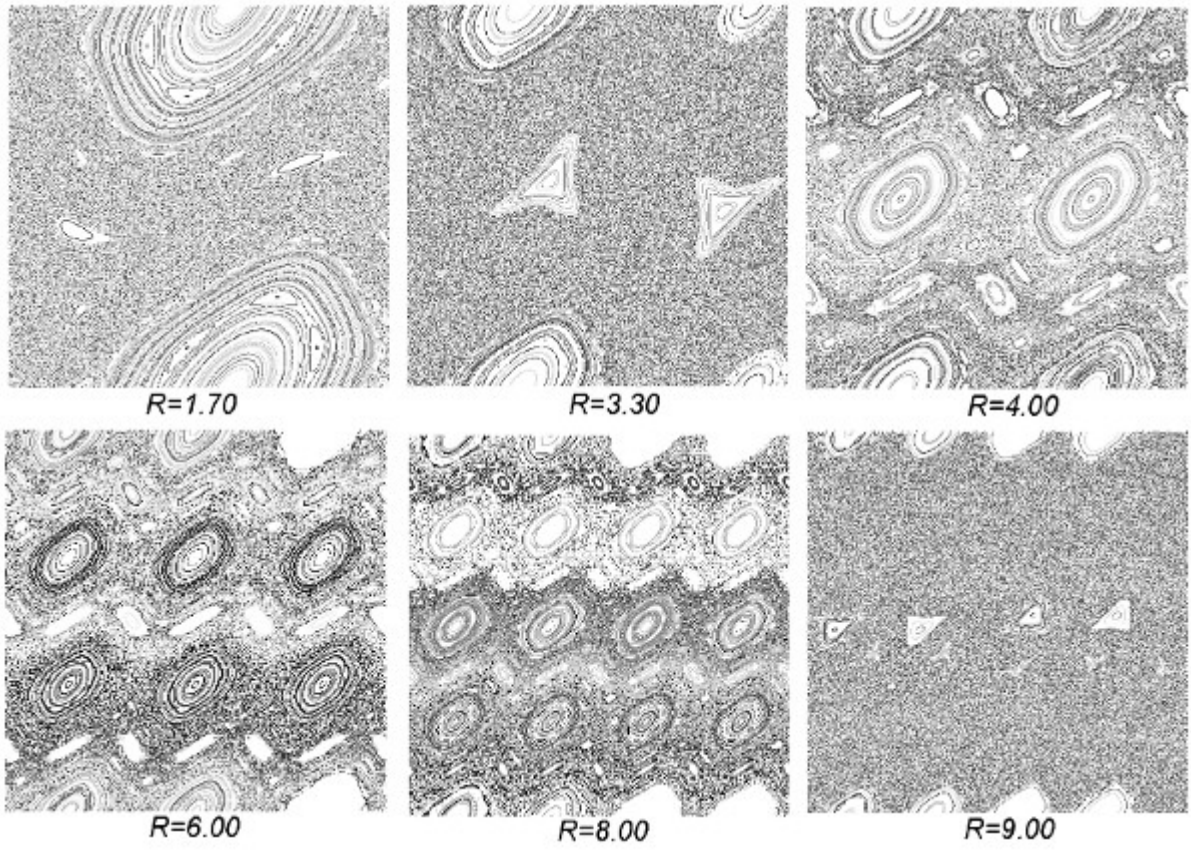


图 8.4.2 K取定为 1.2，改变 R得到的标准映射图

§8.5 埃农保面积映射

法国尼斯天文台埃农(M.Henon,1931- ，henon@obs-nice.fr)教授在研究天体力学过程中提出了许多二维映射，通常说的是埃农映射H：

$x\_(n+1)=1-ax\^2\_n+y\_n$,

$y\_(n+1)=bx\_n$,

其中 a 和 b 都是参数，当 $|b|<1$ 时，H是耗散的，当 $|b|=1$ 时，H是保守的。这个映射是埃农 1976 年提出来的，研究的人比较多，几乎每本混沌书都要提到,其中北京大学力学系黄永念教授用纯代数方法研究埃农映射，很有特色。

这里我们考虑埃农 1969 年提出来的一个保守映射，其形式为 G：

$$x\_(n+1)=x\_n\cos t-y\_n\sin t+x\^2\_n\sin t,$$

$$y\_(n+1)=x\_n\sin t+y\_n\cos t-x\^2\_n\cos t,$$

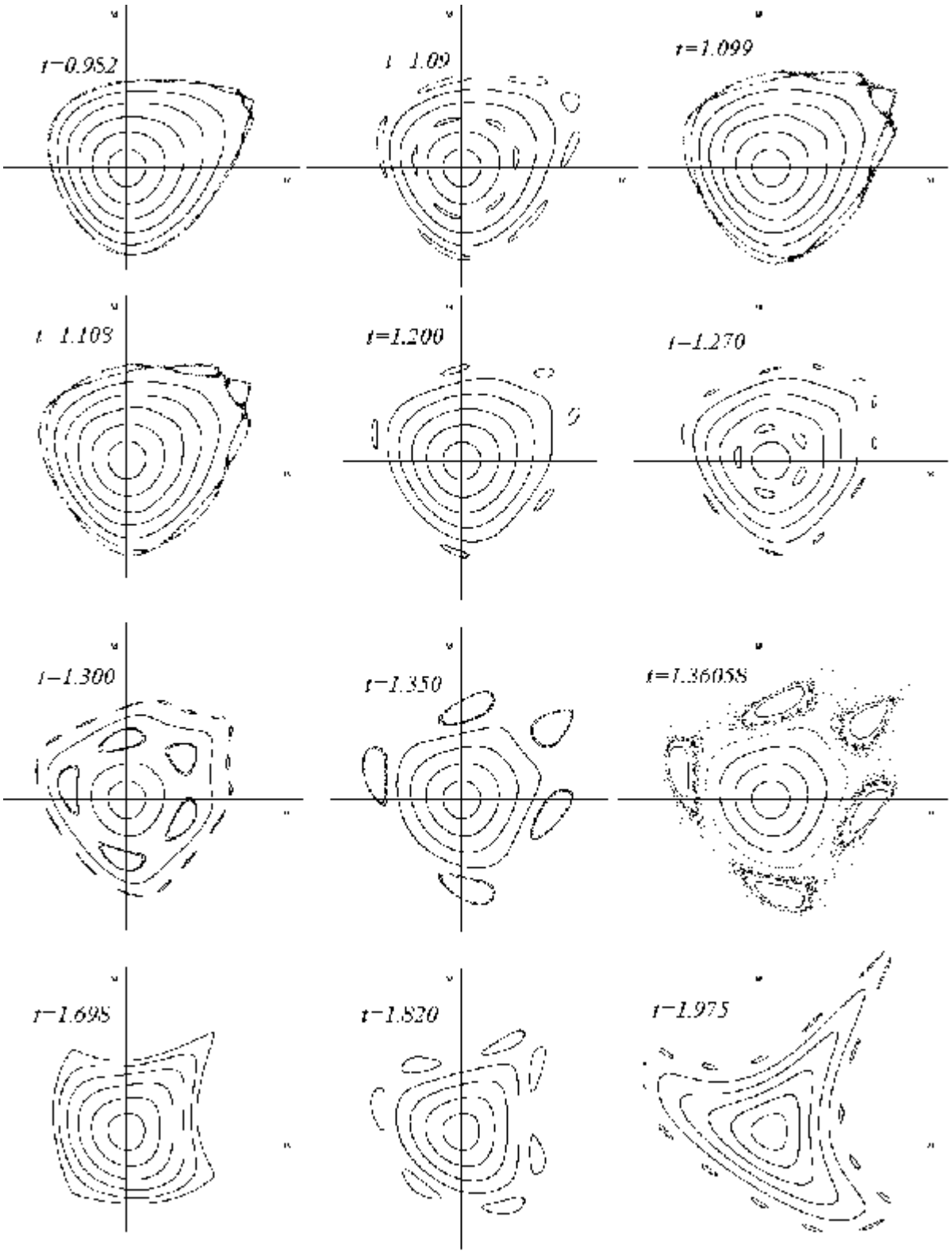


图 8.5.1 保面积埃农映射周期岛结构(改变参数 t 得到)

其中只有一个参数 t 可以取不同的值。这个映射的雅可比矩阵为

J=偏（$x\_(n+1)$, $y\_(n+1)$）/偏(x_n, y_n)

容易验证，J的行列式|J|=sin^2 t+cos^2 t=1，因而映射 G是保面积的。改变 t的取值，可以看到各种不同的环面、“周期岛”和随机层。

根据图 8.5.1，当 t=1.09 时，相图中外圈为周期 7 岛，内圈为周期 6 岛。当 t=1. 099，1.108，1.820 时，有周期 7 岛。当 t=1.2 时，有周期 6 岛。当 t=1.27 时， 外圈有周期 11 岛，内圈有周期 5 岛。当 t=1.30 时，外圈有周期 16 岛，内圈有周期 5 岛 。当 t=1.350，1.36058，10.0 时，有周期 5 岛。当 t=1.975 时，有周期 16 岛。 在图 8.5.2 中，参数 t取 15 时有周期 11 岛。

有兴趣的读者，还可以研究周期倍化分岔过程，用数值计算求得分岔普适常数。本书不作讨论，但给出基本结果如下：

耗散映射：δ=4.669，201，609，102 …，α=−2.502，907，875，095，892 …，

保守映射： δ=8.721，097，200…，α=−4.018，076 …

其中 δ是参数轴分岔间距之比， α是标度因子。这两个常数的存在意味着非 线性系统也存在很强的规律性。这些常数的真正意义仍然有待探索，你可以尝试用 π=3.141 ，592，653…、e=2.718，281，828 …、γ=0.577，215，664…之类常数拼凑，看看能否得出 δ或 α！π 是圆周率，e 是自然对数的底， γ 是欧拉(L.Euler，1707−1783)常数。

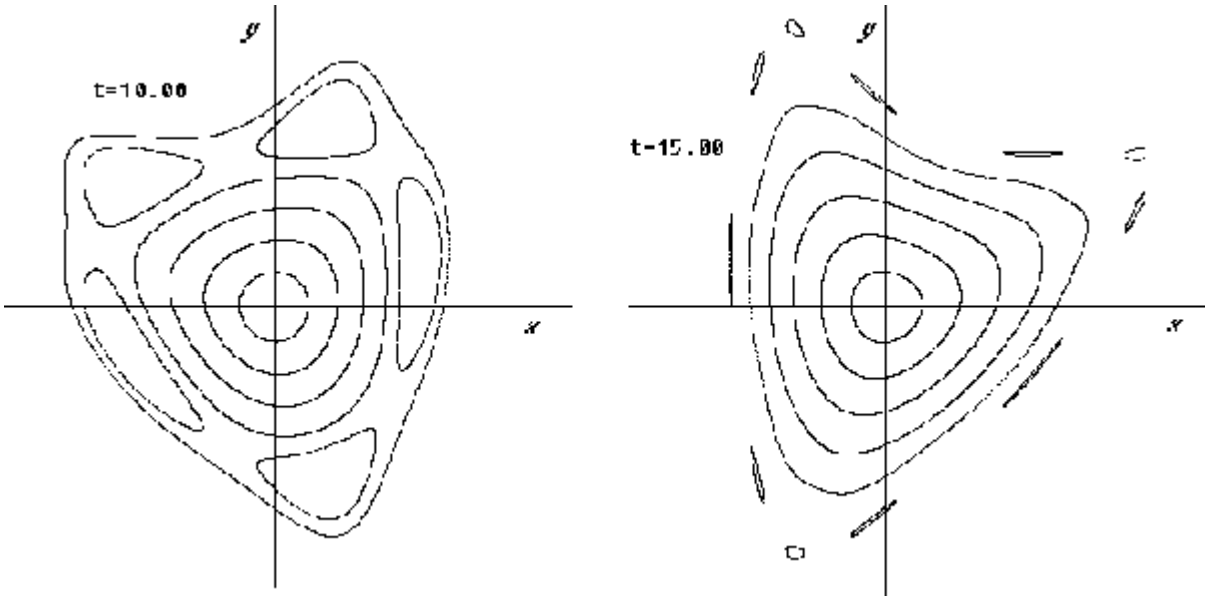


图 8.5.2 参数 t 分别取 10 和 15 时得到的周期 5 岛和周期 11 岛

数学史上，1744 年欧拉证明 e 是无理数，1873 年埃尔米特(C.Hermite,1822-1901)证明 e 是超越数；1761 年，朗伯(J.H.Lambert,1728-1777)证明 π 是无理数,1882 年林德曼(C.L.F.von Lindemann,1852-1939)证明 π 是超越数；至于欧拉常数 γ，不知道它是否是 无理数，更不知道它是否是超越数，但人们猜测它是超越数。非线性科学研究提出来的新常数 δ 和 α，至今仍然不知道它们是否是无理数、是否是超越数，但愿哪位读者能解决这个难题。

§8.6 国王映射

1994 年“科学与艺术奇才”皮克欧沃(C.A.Pickover,1957-),出版了一本有趣的小书《混沌奇境：分形世界虚拟历险记》(*Chaos in Wonderland,Visu al Adventures in a Fractal World*)。书中贯彻始

终使用了一个简单的二维映射，他称之为国王映射(King map)，在每一章开头，作者都给出一张不同面貌的映射图。

科学与艺术奇才——皮克欧沃(1957-)

用二维的国王映射可以生成许多类似三维的复杂曲面，调整参数，或者加上高阶摄动项，会得到更多的花样。国王映射的最初形式是

$x\_(n+1)=\sin(by\_n)+c\sin(bx\_n)$,

$y\_(n+1)=\sin(ax\_n)+d\sin(ay\_n)$,

其中 a, b, c, d 是可调参数。初始值 $x\_0=y\_0=0.1$。比如参数可以取这 样一组值: $a=-1.56918$, $b=2.679879$, $c=0.865145$, $d=0.744728$。国王映射计算程序如下:

```
{wond.pas,1994}
uses Graph,Dos,Crt;
var
    x,y,a,b,c,d, xnew,ynew:real;
    Gd,Gm:integer;
begin
Gd:=Detect; InitGraph(Gd,Gm,'D:\PASCAL');
x:=0.1;y:=0.1;
a:=-0.9666918; {-0.97}
b:=2.679879; {2.8}
c:=0.565145; {0.45,0.76}
d:=0.744728; {0.71}
{-1.86<x<1.86}; {-1.51<y<1.51};
```

```
repeat
      xnew:=sin(y*b)+c*sin(x*b); {-e*sin(x*(1-x)}
      ynew:=sin(x*a)+d*sin(y*a);
      x:=xnew;  y:=ynew;
      PutPixel(round(x*130)+300,225-round(y*160),15);
until KeyPressed;
CloseGraph;
end.
```

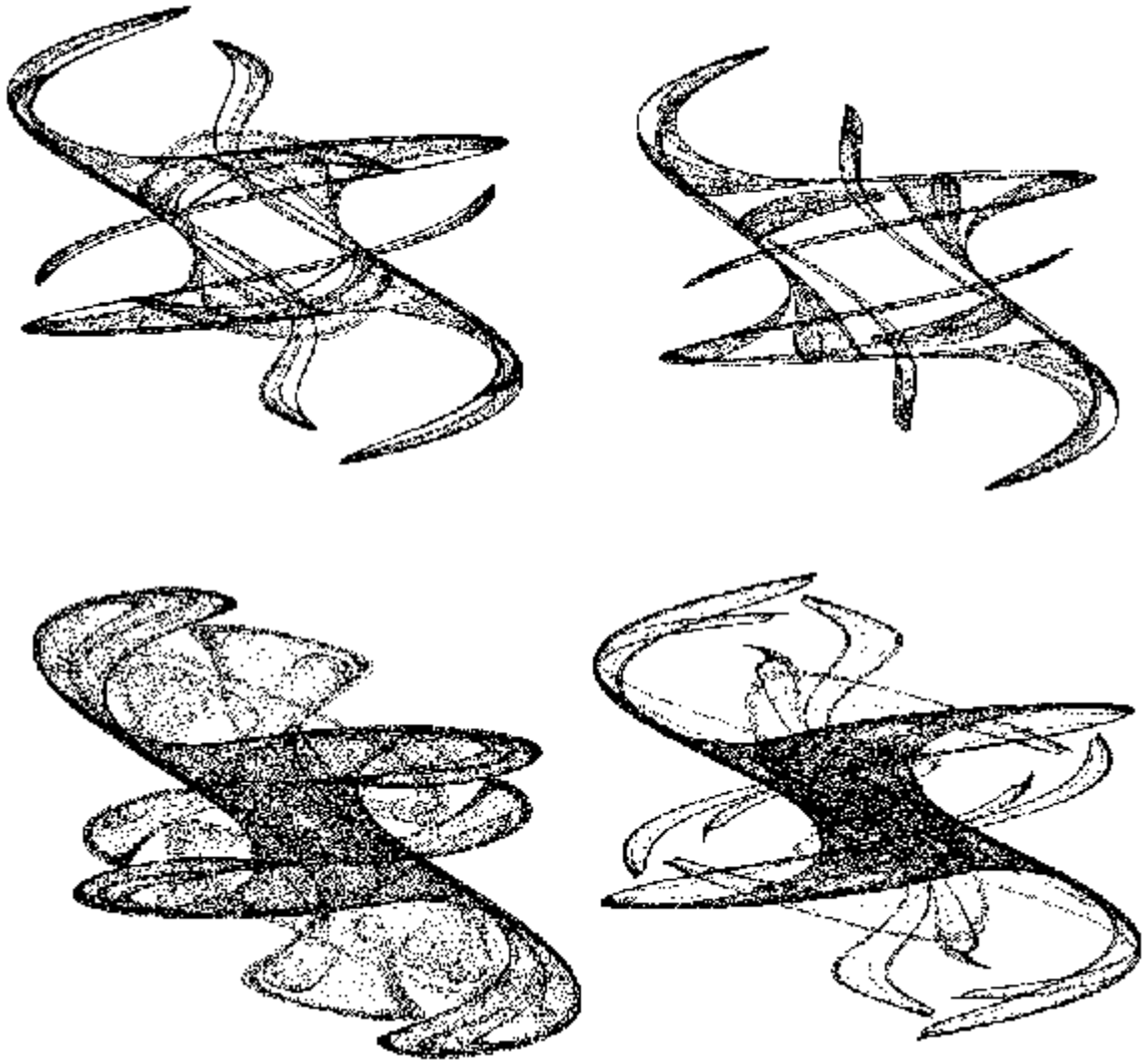


图 8.6.1 国王映射图谱

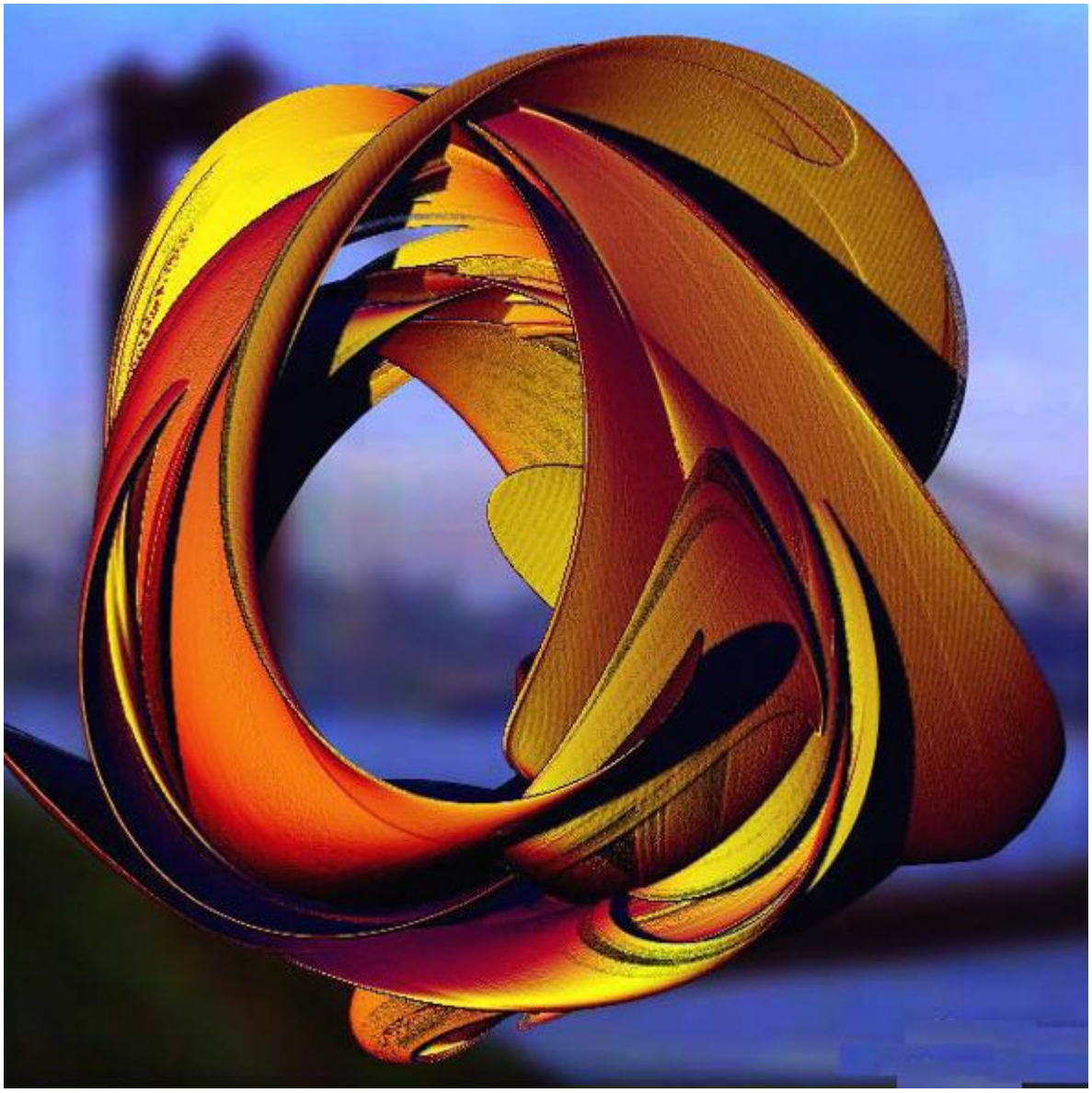


图 8.6.2 国王映射图谱

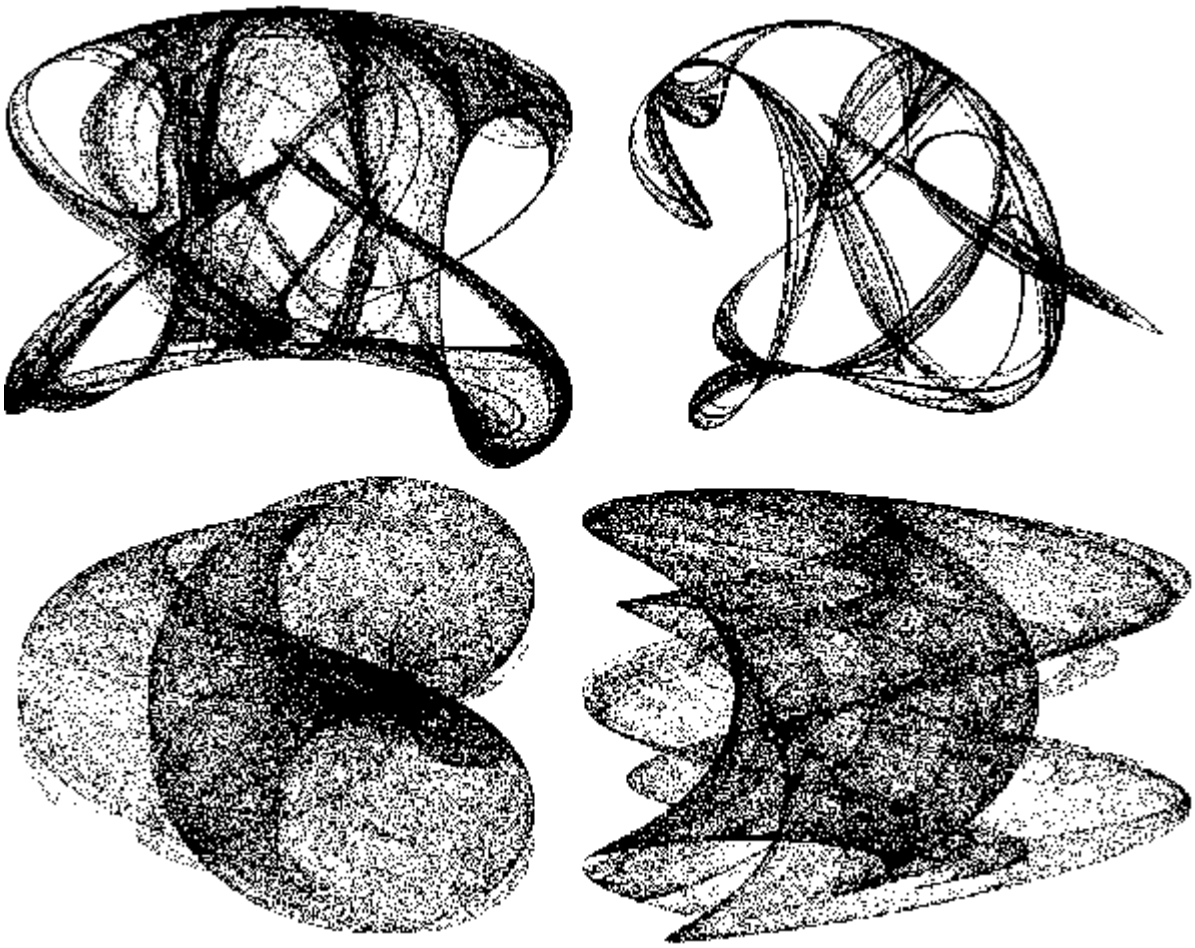


图 8.6.3 由映射得到的复杂分形曲面

最后我们给出另一种国王映射，它的特点是有三个迭代方程，描点时 z 可用也可不用 。映射的具体形式为:

$$x\_(n+1)=\sin(a\,y\_n)-z\cos(b\,x\_n),$$

$$y\_(n+1)=z\sin(c\,x\_n)-\cos(d\,y\_n),$$

$$z\_(n+1)=e\sin x\_n.$$

其中参数取值为 a=2.24, b=0.43, c=−0.65, d=−2.43， e 的取值可在 0.5 至 1.0 之间改变。

想得到清楚的国王映射混沌图，操作步骤是:

1)用 PASCAL 程序以颜色 15 向屏幕描足够多的点(用 1024×768 分辨率较好);

2)用屏幕剪裁方式调入到 Photoshop 或者 PhotoStyler 中去;

3)将图形反转;

4)将 16 色图形变成灰度图;

5)将灰度图变成 1 位的黑白图;

6)将图形存成标准格式的图形文件(如 BMP，TIF，GIF 或者 JPG)。

§8.7 三翅鹰映射

本节给出的映射很容易写出算法和程序，但迭代公式却不容易简单地写出，用它可以生成带有三个翅膀的雄鹰，所以称它三翅鹰映射。计算“三翅鹰”的程序如下:

```
{Feath.pas Generating animal feathers!}
uses Graph,Dos,Crt;
var
      x,y,k,aa,b,c,d:real;
      w,z,u,xnew,ynew:real;
      Gd,Gm,i,n,p:integer;
begin
```

```
Gd:=Detect; InitGraph(Gd,Gm,'D:\PASCAL');
aa:=-0.45; b:=0.93; c:=2-2*aa;
x:=1; y:=1;
w:=aa*x+c*(x*x)/(1+x*x);
repeat
        PutPixel(round(x*30)+250,200-round(y*25),15);
        z:=x;
        x:=b*y+w; {-x/3.5}
        u:=x*x; {/2}
        w:=aa*x+c*u/(1+u); {(1+u+x/4) ,or 3,5}
        y:=w-z;
until KeyPressed;
CloseGraph;
end.
```

这个程序简单透明，但作出的图形却不一般。修改某些项，可以让翅膀细些、有波纹等等。 这个例子给我们一个启示：我们甚至不必知道映射的具体公式，尝试一步一步写出算法，不断修改，便可以作出好看的图形。数学、计算机基础较好的读者可以考虑采取“体绘制”方法，计算并描绘出实映射的三维立体图。

用映射的办法生成高精度的图形,必须采用高分辨率的图形方式,如 VGA 640×480 或者 1024 ×768。必要时也可以先将数据写到文件中去，再转化为标准图形文件，那样做的缺点是费时间，但有时也值得试试。顺便一提，分形图的精度最好用象素个数来衡量，而不是像通常那样用多少 DPI 来衡量。只要象素足够多，分形图的质量就有保障。

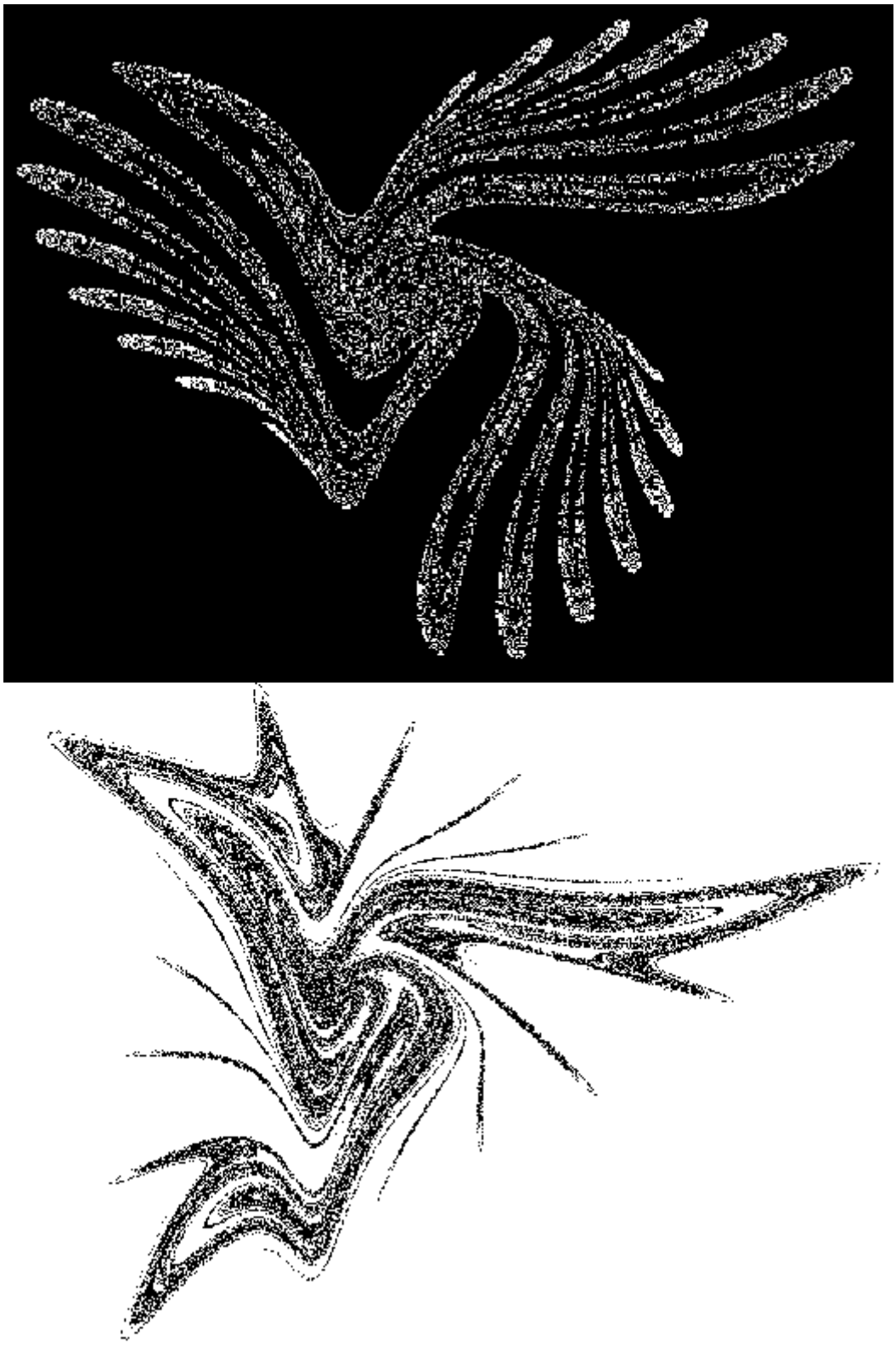


图 8.7.1 三翅鹰映射图谱

§9.1 如何获得软件

近年网络上流行一个非常好的分形图形软件 Fractint,它可以制作许多分形图形，有条件的读者都应该亲自试一试。“Fract”代表“Fractal”(分形)，“int”代表“integer”(整数)。之所以取这个名字，是因为 1988 年 9 月此软件第一个版本计算分形时采用了快速的 386 专用 32 位整数算法。Fractint 软件 1989 年 4 月出现 7.0 版，1990 年 12 月已有 12.0 版，1996 年有 19.5 版,现在已有案可 20.0 版。

Fractint 的开发过程是十分有趣的,其中有个“石头汤故事”(The Stone Soup Story),欲知详情，请参阅 Fractint 的说明书(以文件 Fractint.doc 存贮，见演示程序说明)。这个软件的创始人是泰勒(B.Tyler),他毕业于康奈尔大学数学系，是 PC 和 Unix 程序设计专家。后来参与者还有几十人,主要有魏格纳(T.Wegner)、奥苏茨 (J.Osuch)、娄沃(W.Loewer)、阿兰(K.C.Allen)。通过“石头汤”把素不相识的人物聚集起来，大家共同开发一个有趣的软件，不是为了钱，也不是为了名，这是信息时代、网络时代的一个奇迹。

本章以最简捷的方式介绍软件 Fractint 20.0。你也许奇怪，通常的软件版本不过 3.0,4.0,多的可能到 7.0，还没听说过 19.5,20.0 的。Fractint 软件确实与众不同，这表现在五个方面:

1)免费使用，可以任意从网上下载此软件的各种版本,甚至源代码!

2)集体开发，主要通过 Internet 进行。你如有兴趣并确有高见，也可

以申请参加。

3)由于作者多且能干，软件版本更新速度快。

4)兼容性强，计算速度快(由于采用整数运算，速度目前看是最快的)。

5)开放程度高，设计者并不想保守秘密。

新版 Fractint 软件都上载到 CompuServe 网络上，以两种自解包文件存贮：一是 FRAINT.EXE，为可执行文件及说明书；二是 FRASRC.EXE，为源代码。

最新版 Fractint 也可以从 CompuServe 的 GO GRAPHICS 论坛上找到。如果你没办法接触 CompuServe 网络，可以在 Internet 上搜索关键词：FRAIxxx.ZIP 和 FRASRxxx.ZIP,其中 xxx 表示版本号。也有用两位数表示的，这时应搜索 FRAINTxx.ZIP 和 FRASRCxx.ZIP。

Fractint 软件也可以从下面 WWW 地址直接获取：

http://spanky.triumf.ca/www/fractint/fractint.html

也可以用匿名 FTP 方式获取，方法是，通过 FTP，以 ANONYMOUS 登录到 spanky.triumf.ca 上，切换目录到 pub/fractals/programs/ibmpc,然后下载 fraintxxx.zip。

如果你还是找不到，可以尝试下面的地址：

ftp://ftp.simtel.net/pub/simtelnet/msdos/graphics/fraxxx.zip

想得到 X Windows 版本软件可试试下述地址：

ftp://ftp.cs.berkeley.edu/ucb/sprite/xf*.

您也可以向吉芬(N.Giffin)索取Fractint软件: noel@triumf.ca。

在本课程的演示程序中你们也可以下载这些软件。

现在,Fractint 已经有了 Windows 和 linux 下的版本.