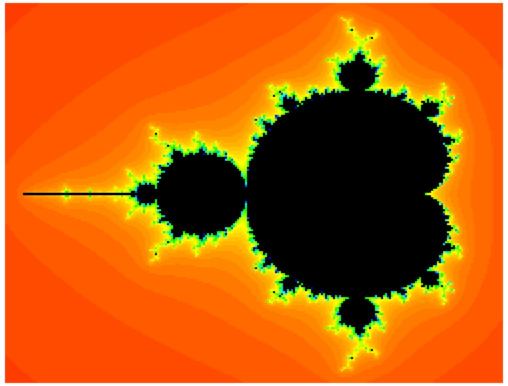
第五章 分形图形生成

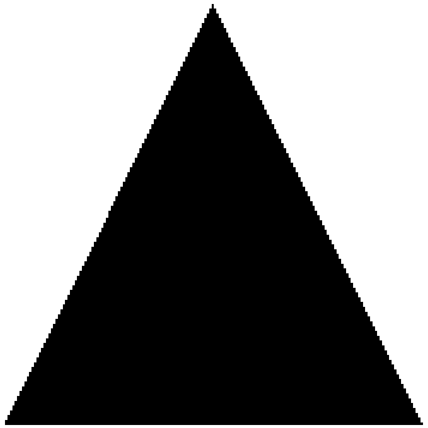
一、什么是分形（Fractal）

- 例子（数学定义的分形）



Mandelbrot集

Koch曲线



Sierpinski地毯

蕨类植物

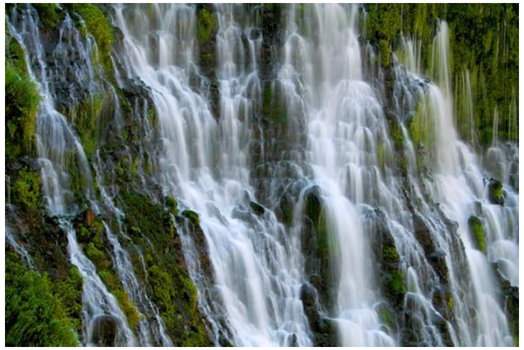
一、什么是分形（fractal）

- 例子（自然界中的分形）







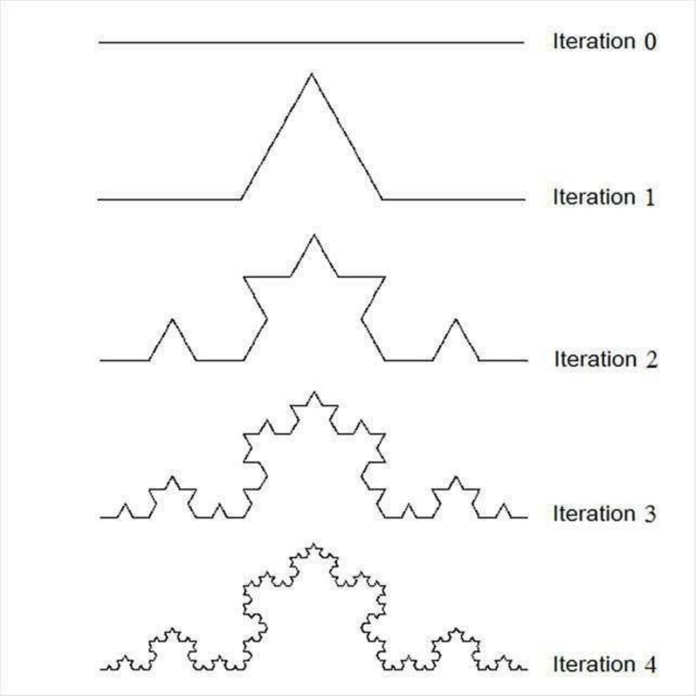
一、什么是分形（Fractal）

- 1、什么是分形
  - F具有精细的结构，在任意小的尺度之下总有复杂的细节；
  - F是不规整的，其整体和局部不能用传统的几何语言描述；
  - F通常具有相似性，近似的或统计的；
  - 一般的，F的分形维数大于其拓扑维数；
  - F以非常简单的方法确定，由迭代过程产生

Benoît B. Mandelbrot
(1924－2010)

一、什么是分形（fractal）

- 2、分形维数



$$M \cdot r^D = 1 \Rightarrow D = \frac{\log\frac{1}{M}}{\log r}$$

	比例系数r	相似子集的个数M	分形维数
直线	$r = \frac{1}{N}$	$M = N$	1
平面	$r = \frac{1}{N}$	$M = N^2$	2
立方体	$r = \frac{1}{N}$	$M = N^3$	3
Koch曲线	$r = (\frac{1}{3})^N$	$M = 4^N$	1.26

二、L系统

- 1、简单的L系统

(1) 定义：有序的三元素集合，G=<V, W, P>

其中：V: 字母表

W: 非空单词

P: 产生式的有限集合

Eg: V={a, b}

W=b

P: b->a，a->ab

⇩

G: b->a->ab->aba->abaab->abaababa->abaababaabaab……

二、L系统

- 1、简单的L系统

(2) 图形解释

F: 向前移动一步，步长为d，起始方向为α;

+： 向左转角δ

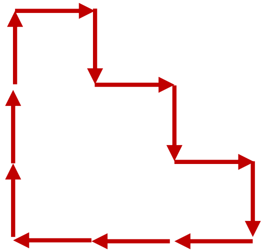
-： 向右转角δ

Eg1: V={F，+，-}

α =90°, δ=90°

W=FFF-F+F-F+F-F-FFF

F:

W: 

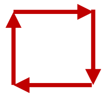
二、L系统

- 2、分形的生成

V={F， +, —};

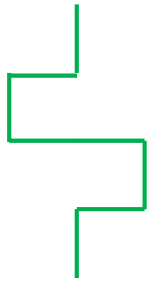
W=初始图;

P: 生成元



W=F-F-F-F

α =90°, δ=90°



P: F->F-F+F+FF-F-F+F

W=F

α =0°, δ=60°

P: F->F+F- -F+F

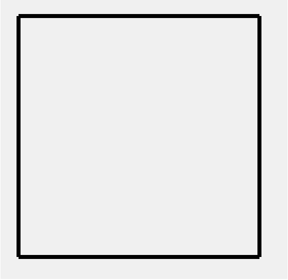
二、L系统

- 2、分形的生成

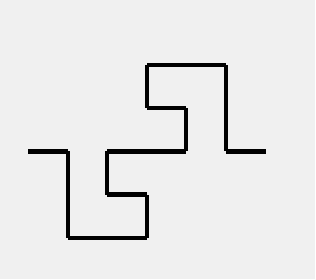
W=F-F-F-F

α =0°, δ=90°

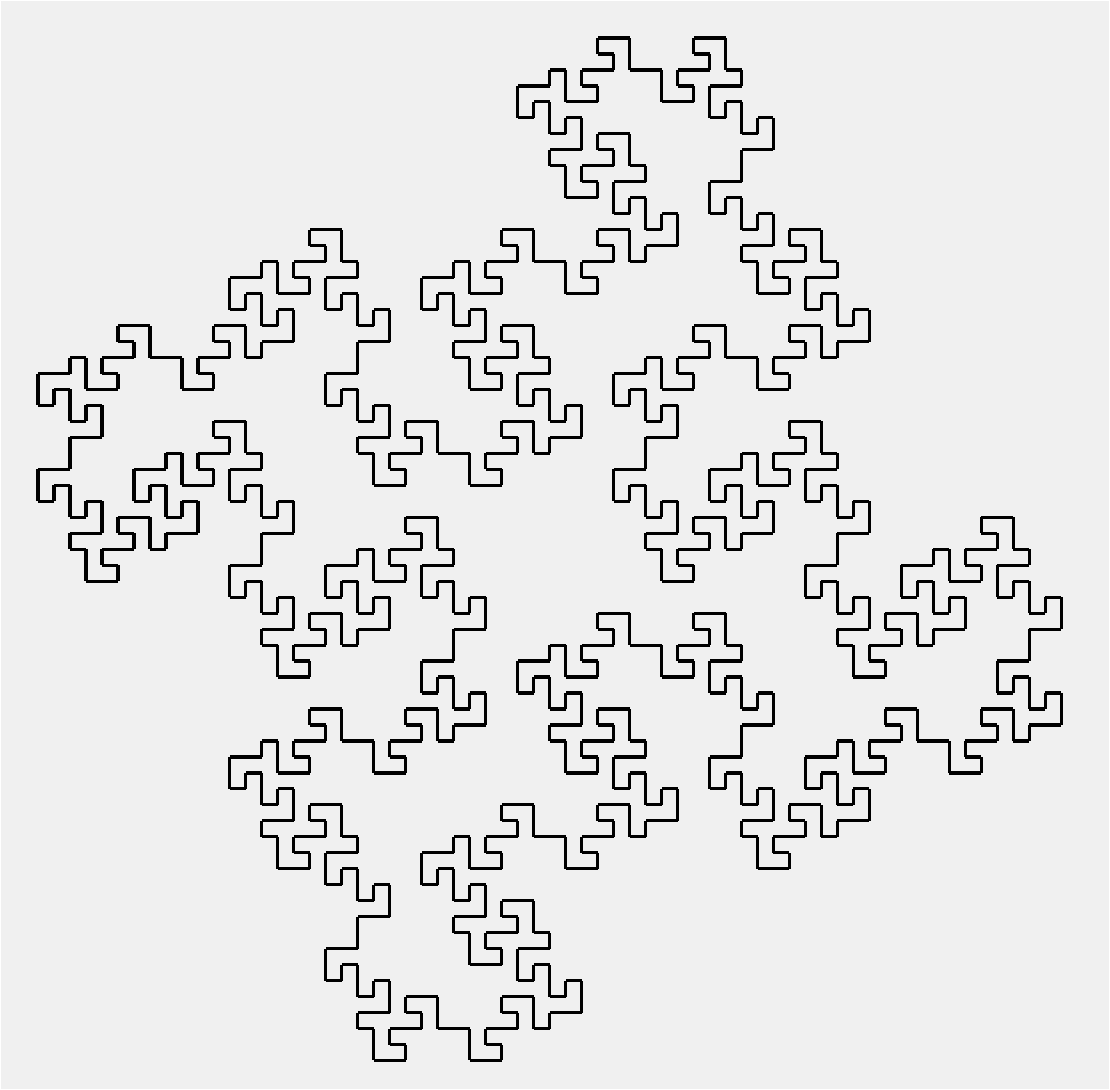
P: F->F-FF+FF+F+F-F-FF+F+F-F-FF-FF+F



初始图



生成元



n=2

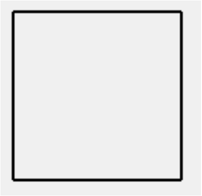
二、L系统

- 2、分形的生成

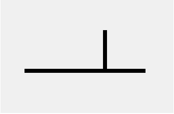
W=F-F-F-F

α =0°, δ=90°

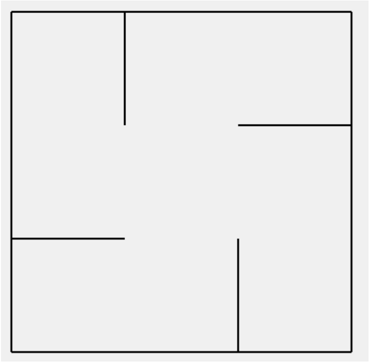
P: F->FF+F--F+F



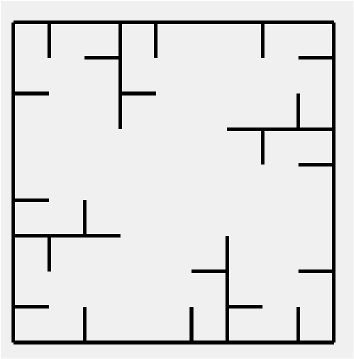
初始图



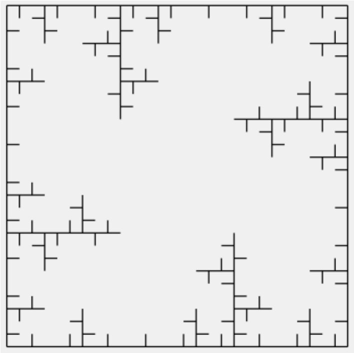
生成元



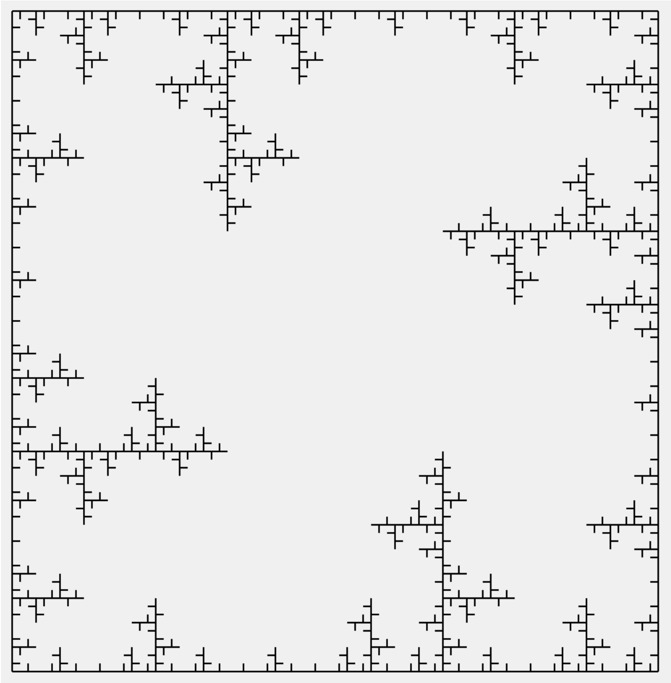
n=1



n=2



n=3



n=4

二、L系统

- 3、边改写

生成更复杂的图形，将直线多分成两种类型：

⇩

F_l：只在左边画图

F_r：只在右边画图

V={F_l ，F_r ，+, —}

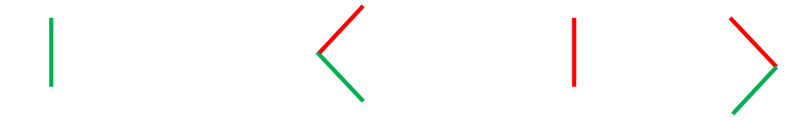
eg1:

W: F_l

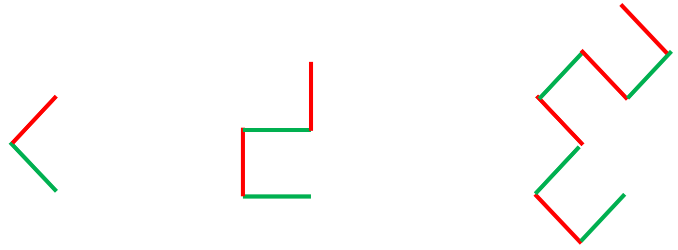
α =90°, δ=45°

P1： F_l -> + F_l — — F_r

P2： F_r -> — F_l + + F_r



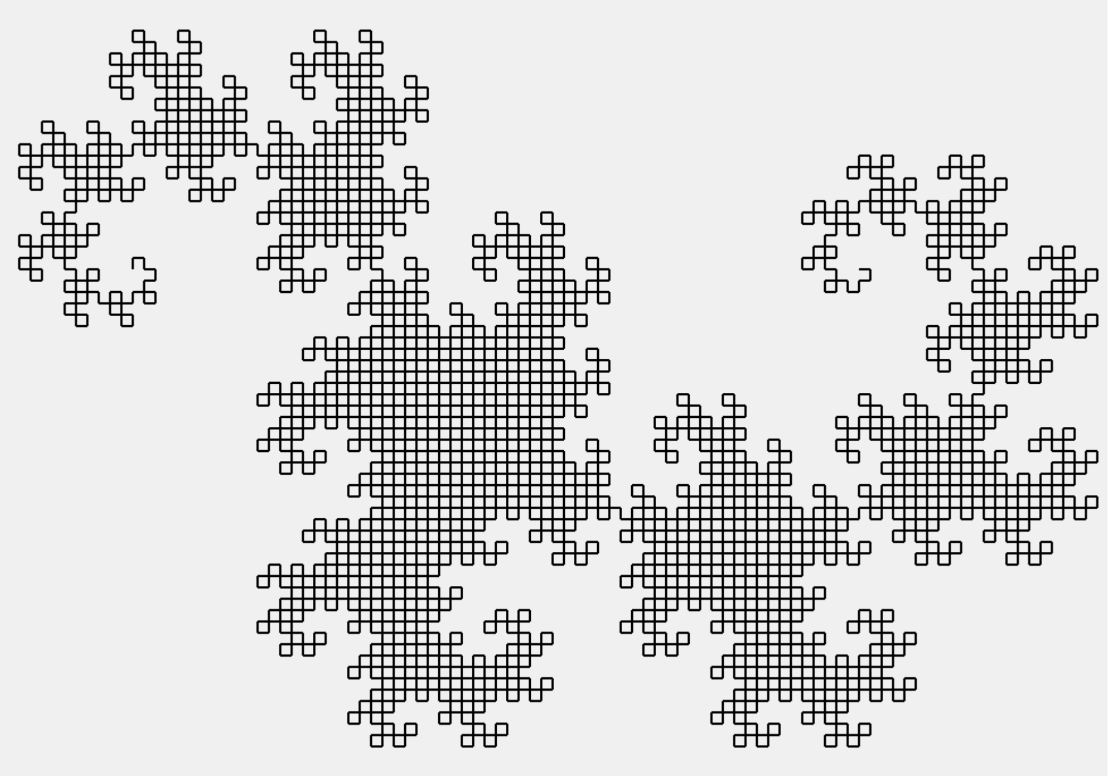
F_l　　P1　　F_r



n=1　　n=2　　n=3

二、L系统

- 3、边改写



二、L系统

- 3、边改写

eg2:

W: F_l

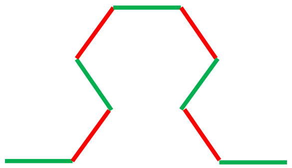
α =0°, δ=60°

P1： F_l ->+ F_l — F_l — F_r

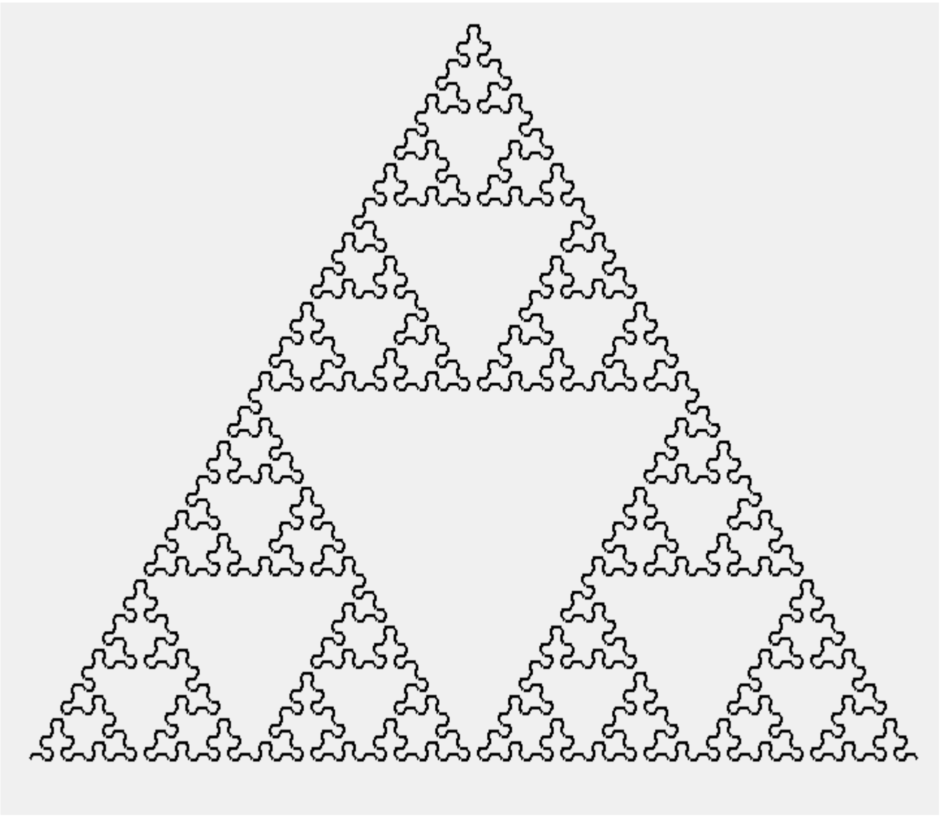
P2： F_r ->— F_l + F_r + F_l

W:F_l

n=1



n=2



n=6

二、L系统

- 4、分支结构

不是一笔绘成的，同一线段多次绘制，表示比较复杂的分叉，如树枝。

V={F，+，—，[，]}；

其中：[表示将当前状态（位置、方向）入栈
] 表示弹出一个状态作为当前状态

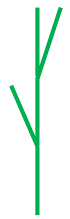
eg1：

W=F

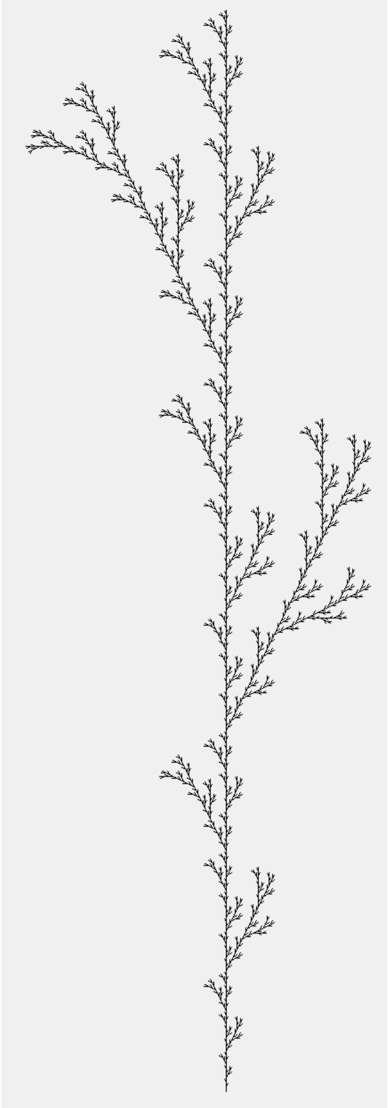
α =90°, δ=30°

P: F->F[+F]F[-F]F

W: F



P



n=5

二、L系统

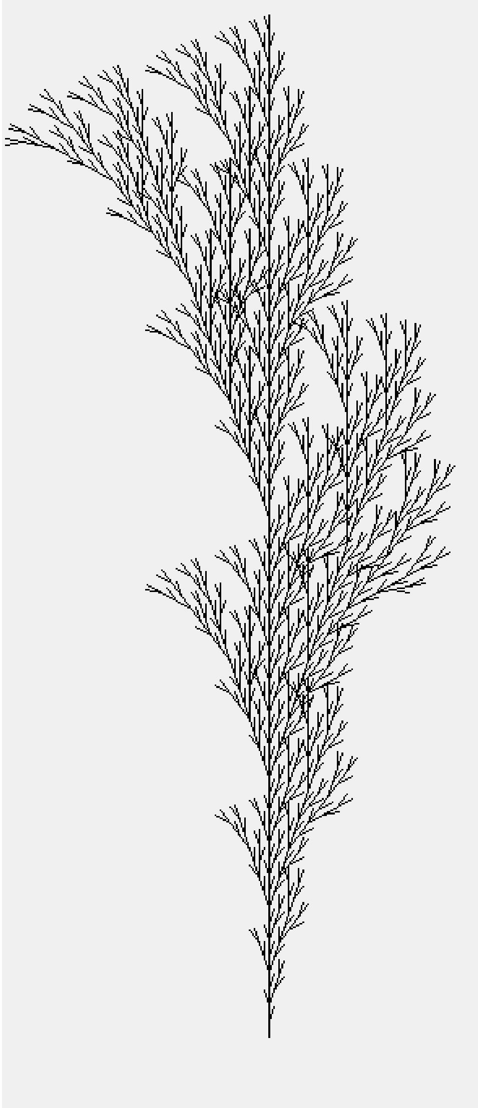
- 4、分支结构

eg2:

W=F

α =90°, δ=20°

P: F-> F[+F]F[-F][F]



n=5

二、L系统

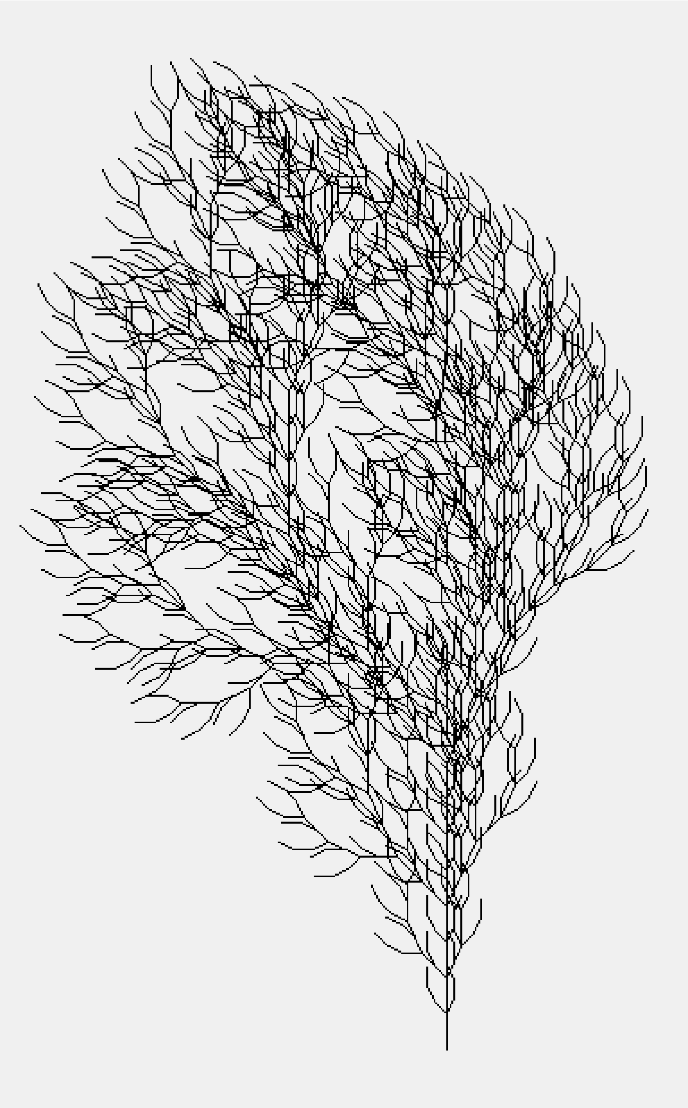
- 4、分支结构

eg2:

W=F

α =90°, δ=22.5°

P: F-> FF[-F+F+F]+[+F-F - F]



n=4

二、L系统

- 5、随机L系统
  - 保留特征，出现细节不同：
  - F的步长d是随机的
  - 产生式是随机的

G=<V, W, P, π>

eg1：

W=F，α =90°, δ=30°

P_1: F $\xrightarrow{\pi(p_1)}$ F[+F]F[-F]F

P_2: F $\xrightarrow{\pi(p_2)}$ F[+F]F[-F[+F]]

P_3: F $\xrightarrow{\pi(p_3)}$ FF[-F+F+F]+[+F-F - F]

$\pi(p_i)=\frac{1}{3}$

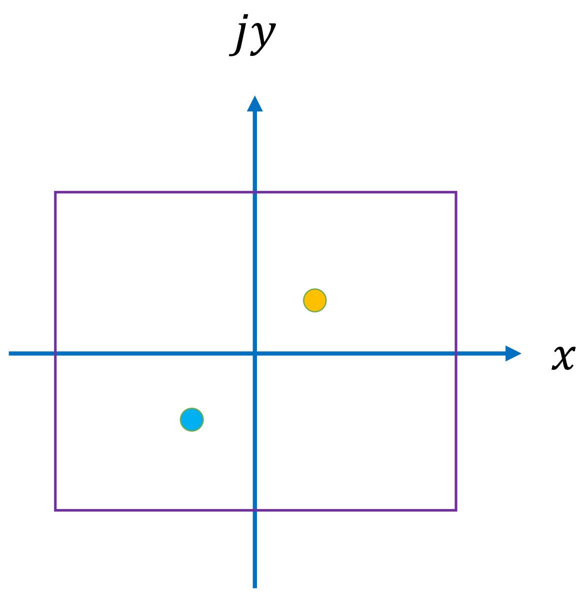
三、Julia集和Mandelbrot集

- 1、Julia集定义

$Z = Z^2 + C$ $\quad\quad$ $Z = x + jy$

给定一个C，计算复平面中每个Z是否收敛

Julia集是Z空间的分形图形



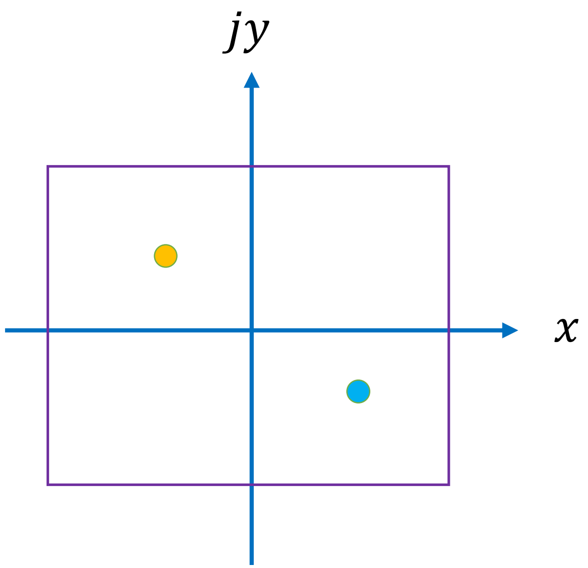
Z空间

- 2、Mandelbrot集定义

$Z = Z^2 + C$ $\quad\quad$ $Z = x + jy$

$Z_0 = 0$，计算复平面中每个C是否收敛

Mandelbrot集是C空间的分形图形



C空间

三、Julia集和Mandelbrot集

(x_{max}, y_{max})

$a \times b$

(x_{min}, y_{min})

- 3、Julia集生成算法

step1：设置$C=p+jq$，迭代次数N，颜色等级G，屏幕大小$a \times b$;

step2: 设置视窗中每一个点（n_x，n_y），计算对应的 $Z_0 = x_0 + jy_0$

$$x_0 = x_{min} + n_x\,\Delta x;$$
$$y_0 = y_{min} + n_y\,\Delta y;$$

$$\Delta x = \frac{x_{max} - x_{min}}{a-1}, \quad \Delta y = \frac{y_{max} - y_{min}}{b-1}$$

step3:计算： $Z_{k+1} = Z_k^2 + C$, 即： $\Longrightarrow$ $\begin{cases} x_{k+1} = x_k^2 - y_k^2 + p \\ y_{k+1} = 2x_k y_k + q \end{cases}$

三、Julia集和Mandelbrot集

- 3、Julia集生成算法

step4：计算$r = x_{k+1}^2 + y_{k+1}^2$

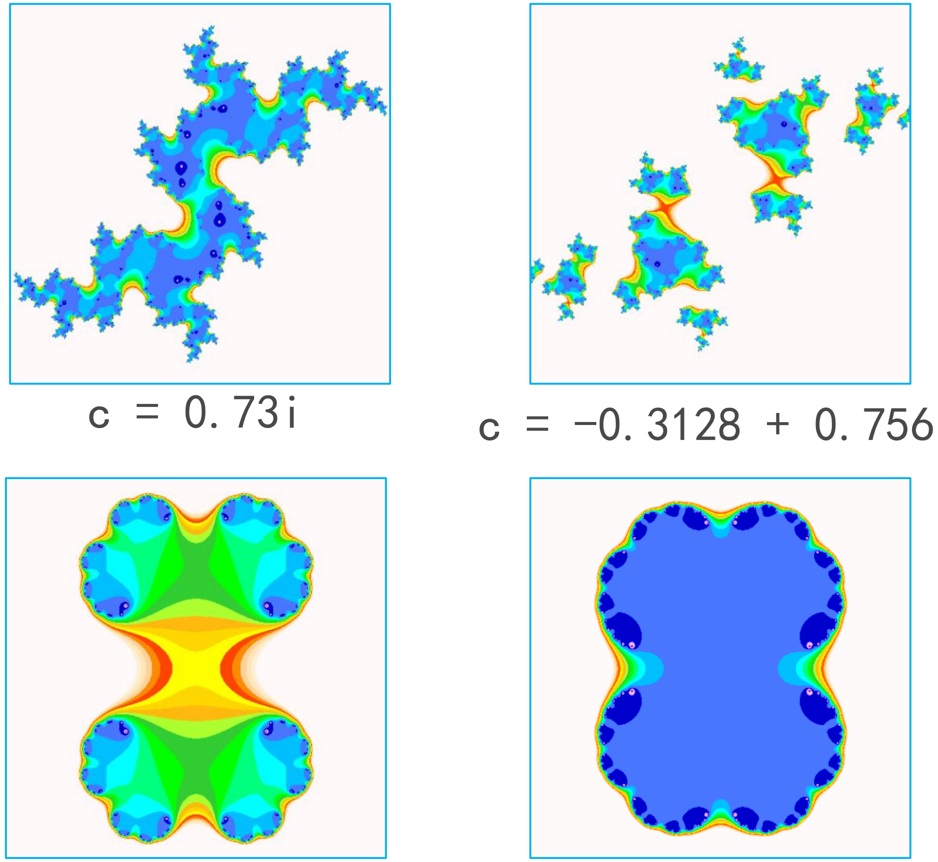
if r >2 or k =N, 转到step5;

else k = k +1, 回到step3;

step5：color=$k\frac{G}{N}$;

画点（n_x，n_y， color ）

step6：回到step2，直到$n_x, n_y = a, b$



c = 0.73i

c = -0.3128 + 0.756i

c = 0.45

c = 0.3

三、Julia集和Mandelbrot集

- 4、Mandelbrot集生成算法

step1：设置迭代次数N，颜色等级G，屏幕大小$a \times b$；

step2：设置视窗中每一个点（n_p，n_q）的C，$C = p + jq$

$$p = x_{min} + n_p \Delta x;$$
$$q = y_{min} + n_q \Delta y;$$

$$\Delta x = \frac{p_{max} - p_{min}}{a-1}, \quad \Delta y = \frac{q_{max} - q_{min}}{b-1}, \quad Z_0 = 0$$

step3:计算：$Z_{k+1} = Z_k^2 + C$, 即：

$$\Longrightarrow \begin{cases} x_{k+1} = x_k^2 - y_k^2 + p \\ y_{k+1} = 2x_k y_k + q \end{cases}$$

三、Julia集和Mandelbrot集

- 4、Mandelbrot集生成算法

step4：计算$r = x_{k+1}^2 + y_{k+1}^2$

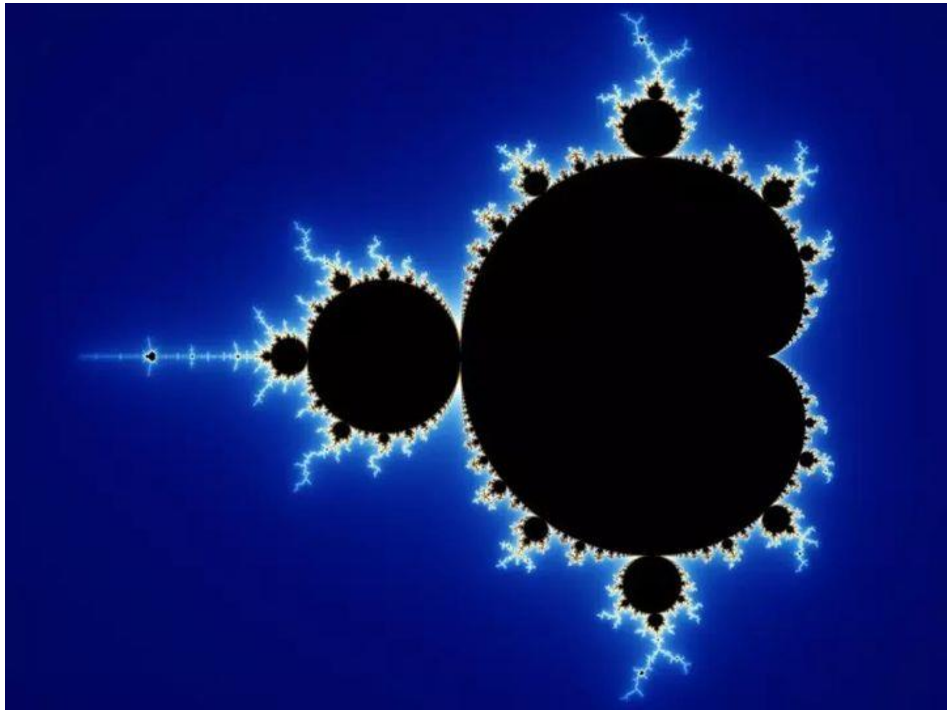
if r >2 or k =N, 转到step5;

else k = k +1, 回到step3

Step5：color=$k\frac{G}{N}$;

画点（n_p，n_q， color ）；

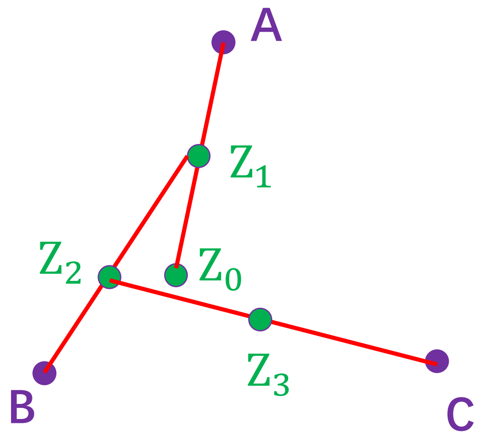
step6：回到step2，直到$n_p, n_q = a, b$



四、迭代函数系统

分形绘制的典型重要方法，在仿射变换的意义下，具有自相似结构，几何对象的整体被定义后，选定若干仿射变换，将整体形态变换到局部，迭代进行，知道满意造型。

- 1、混沌游戏



任取一点Z_0, 投掷硬币，如果硬币正面向上的概率为0.5，反面向上的概率为0.47，立起来的概率为0.03，则：

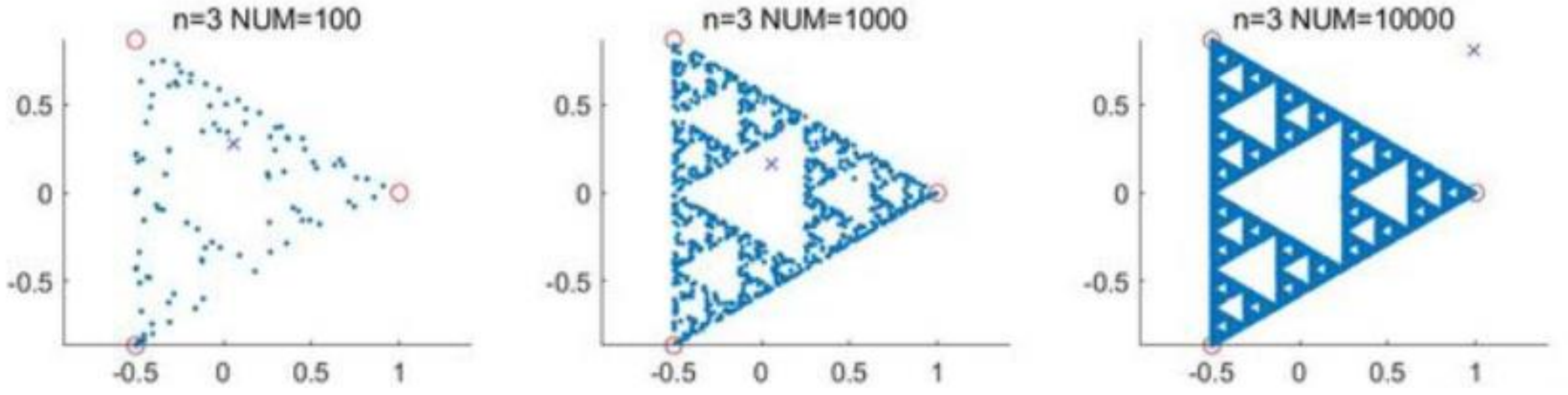
当硬币正面向上时：$Z_{k+1}=\dfrac{Z_k+A}{2}$

当硬币反面向上时：$Z_{k+1}=\dfrac{Z_k+B}{2}$

当硬币立起来时：$Z_{k+1}=\dfrac{Z_k+C}{2}$

四、迭代函数系统

- 1、混沌游戏



四、迭代函数系统

- 2、仿射变换

$$w(x) = Ax + b$$

$$w_k = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} r\cos\theta & -s\sin\varphi \\ r\sin\theta & s\cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

$$= \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

r：沿x轴方向的缩放倍数

s：沿y轴方向的缩放倍数

θ：沿x轴逆时针方向的旋转角度

φ：沿y轴逆时针方向的旋转角度

四、迭代函数系统

- 2、仿射变换

k	a	b	c	d	e	f	P（概率）
1	0	0	0	0.16	0	0	0.01
2	0.85	0.04	-0.04	0.85	0	1.6	0.07
3	0.2	-0.25	0.23	0.22	0	1.6	0.07
4	-0.15	0.28	0.26	0.24	0	0.44	0.85

The End