第二章 迭代-分形图形

苏湘宁 邮箱：suxn@hainanu.edu.cn

迭代-分形图形

- 生活中的分形结构
- 经典分形图形
- 迭代与递归
- 分形与分形维

2.1 生活中的分形结构

- 分形结构：

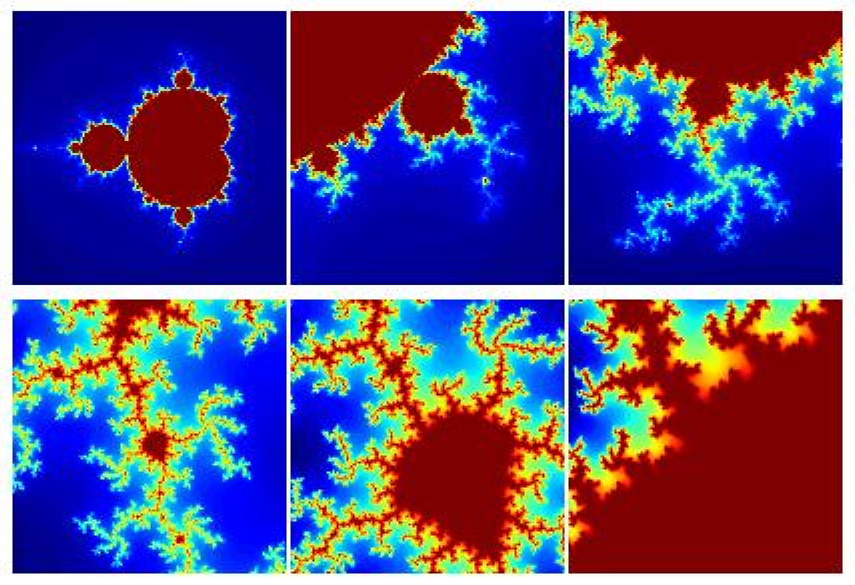
典型特征是自相似性，即局部放大后看到的小尺度形貌和整体大尺度形貌相似。

山脉

羊齿叶

海岸线

2.2 经典分形图形



曼德勃罗特集
(Mandelbrot set，1975)

曾被称为“上帝的指纹”。
这个点集均出自公式:

$$Z_{n+1}=Z_n^2+C$$

对于非线性迭代公式

$$Z_{n+1}=Z_n^2+C$$

所有使得无限迭代后的结果能保持有限数值的复数z的集合(也称该迭代函数的Julia集)连通的C，构成曼德勃罗集

2.2 经典分形图形

复变函数迭代：

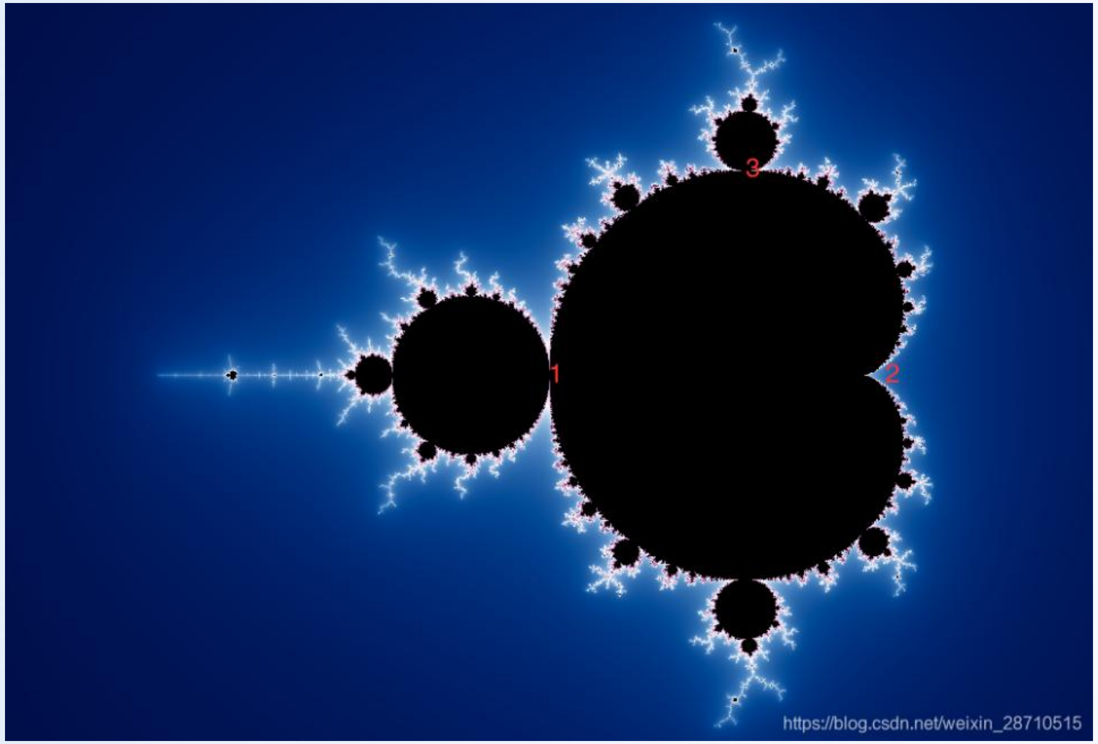
朱利亚（Julia）集合是一个在复平面上形成分形的点的集合。以法国数学家加斯顿·朱利亚（Gaston Julia）的名字命名:

$$Z_{n+1}=Z_n{}^m+C_0$$

当m=2 时，如果令$C=C0$($C0$ 为常数)进行迭代，所得的Z集合称之为Julia集。

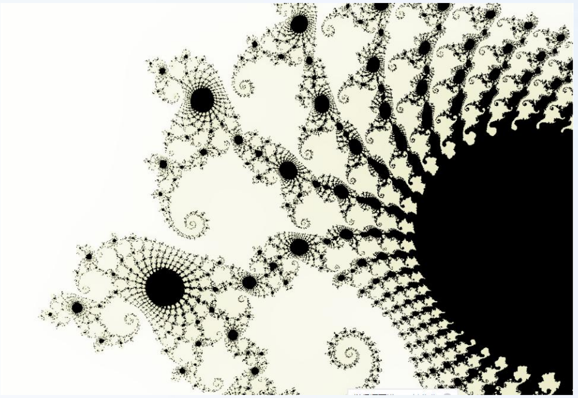
2.2 经典分形图形

每个局部与整体都存在一种相似性

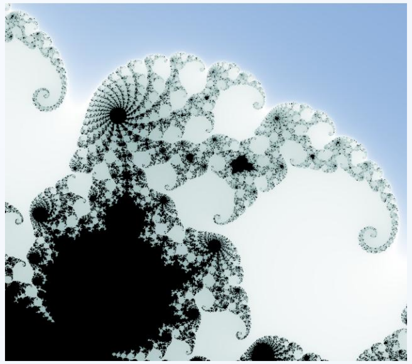


曼德勃罗特集

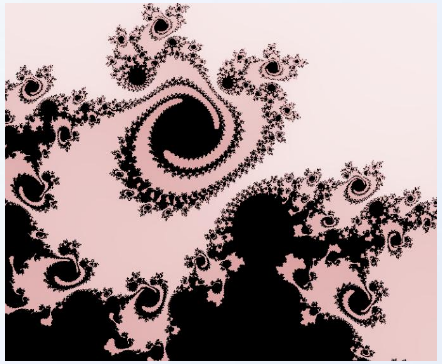
（Mandelbrot set，1975）



“Seahorse Valley” (海马山谷)



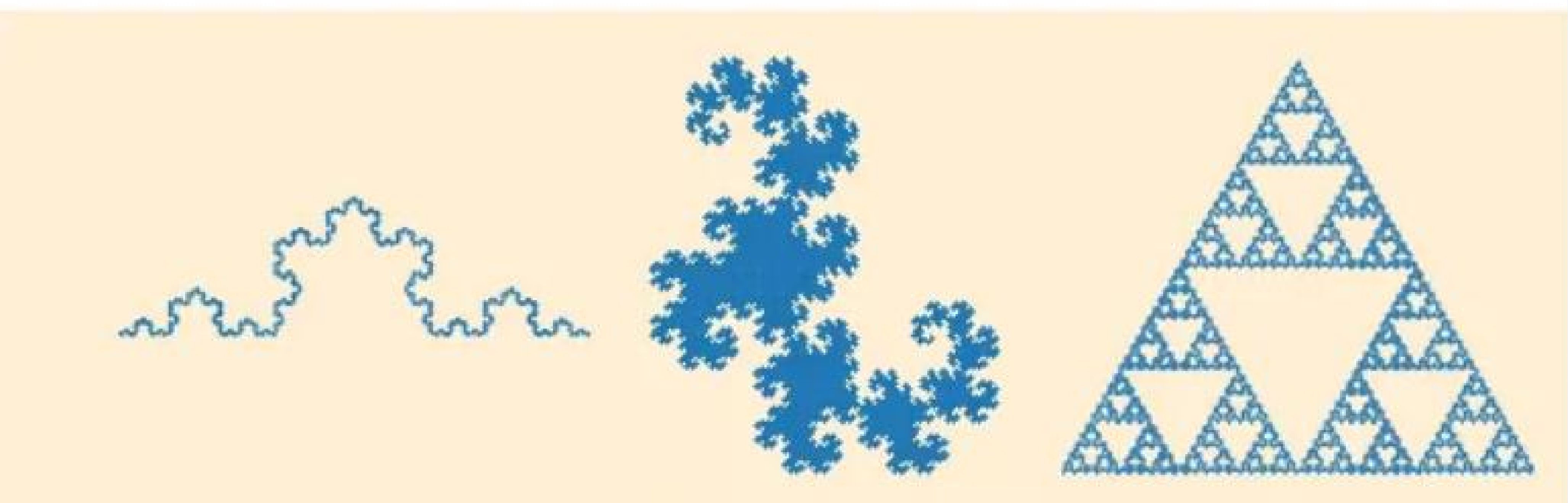
“Elephant Valley”
（大象山谷）



“Triple Spiral Valley”
(三重螺旋)

2.2 经典分形图形

科赫曲线、分形龙、谢尔宾斯基三角



2.2 经典分形图形

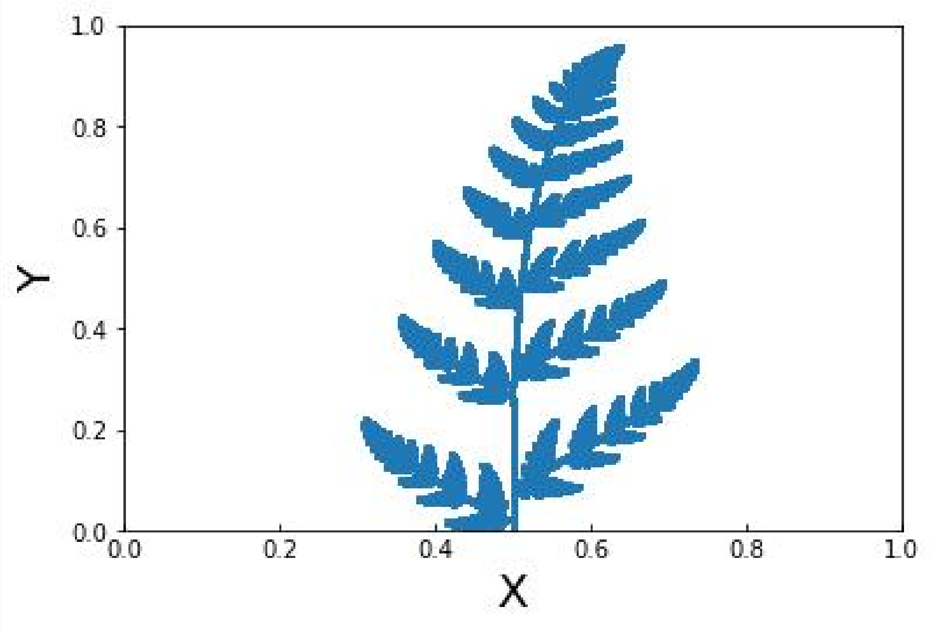
- **羊齿叶：**

  (1) 建立模型

  羊齿叶迭代的模型。

  (2) 确定数值计算的方法。

  (3) 编写程序



在计算机上能自动地连续进行成千上万次的重复计算，就是要形成能自我复制的数据链，也就是要形成能连续进行推算的固定的逻辑关系，用到for循环的运算

2.2 经典分形图形

- 羊齿叶：有四条迭代规则：

$$x^{(k+1)} = 0.5$$
$$y^{(k+1)} = 0.27*y^{(k)}$$

$0.0 < rand < 0.02$

$$x^{(k+1)} = -0.139x^{(k)} + 0.263*y^{(k)} +0.57$$
$$y^{(k+1)} = 0.246*x^{(k)} + 0.224*y^{(k)} -0.036$$

$0.02 < rand < 0.17$

$$x^{(k+1)} = 0.17x^{(k)} - 0.215*y^{(k)} +0.408$$
$$y^{(k+1)} = 0.222*x^{(k)} + 0.176*y^{(k)} -0.0893$$

$0.17 < rand < 0.3$

$$x^{(k+1)} = 0.781x^{(k)} + 0.034*y^{(k)} +0.1075$$
$$y^{(k+1)} = -0.032*x^{(k)} + 0.739*y^{(k)} +0.27$$

$0.3 < rand < 1.0$

- 羊齿叶：四条迭代规则：

```
for irun in range(1,nrun):
    rand= random.random()
    if rand < 0.02:
        xx[irun] = 0.5
        yy[irun] = 0.27*yy[irun-1]
    elif rand >= 0.02 and rand < 0.17:
        xx[irun] = -0.139*xx[irun-1]+0.263*yy[irun-1] + 0.57
        yy[irun] = 0.246*xx[irun-1]+0.224*yy[irun-1] -0.036
```

2.2 经典分形图形

- 羊齿叶：四条迭代规则：

```
elif rand >= 0.17 and rand < 0.3:
    xx[irun] = 0.17*xx[irun-1]-0.215*yy[irun-1] + 0.408
    yy[irun] = 0.222*xx[irun-1]+0.176*yy[irun-1] -0.0893
else:
    xx[irun] = 0.781*xx[irun-1]+0.034*yy[irun-1] + 0.1075
    yy[irun] = -0.032*xx[irun-1]+0.739*yy[irun-1] +0.27
```

2.2 经典分形图形

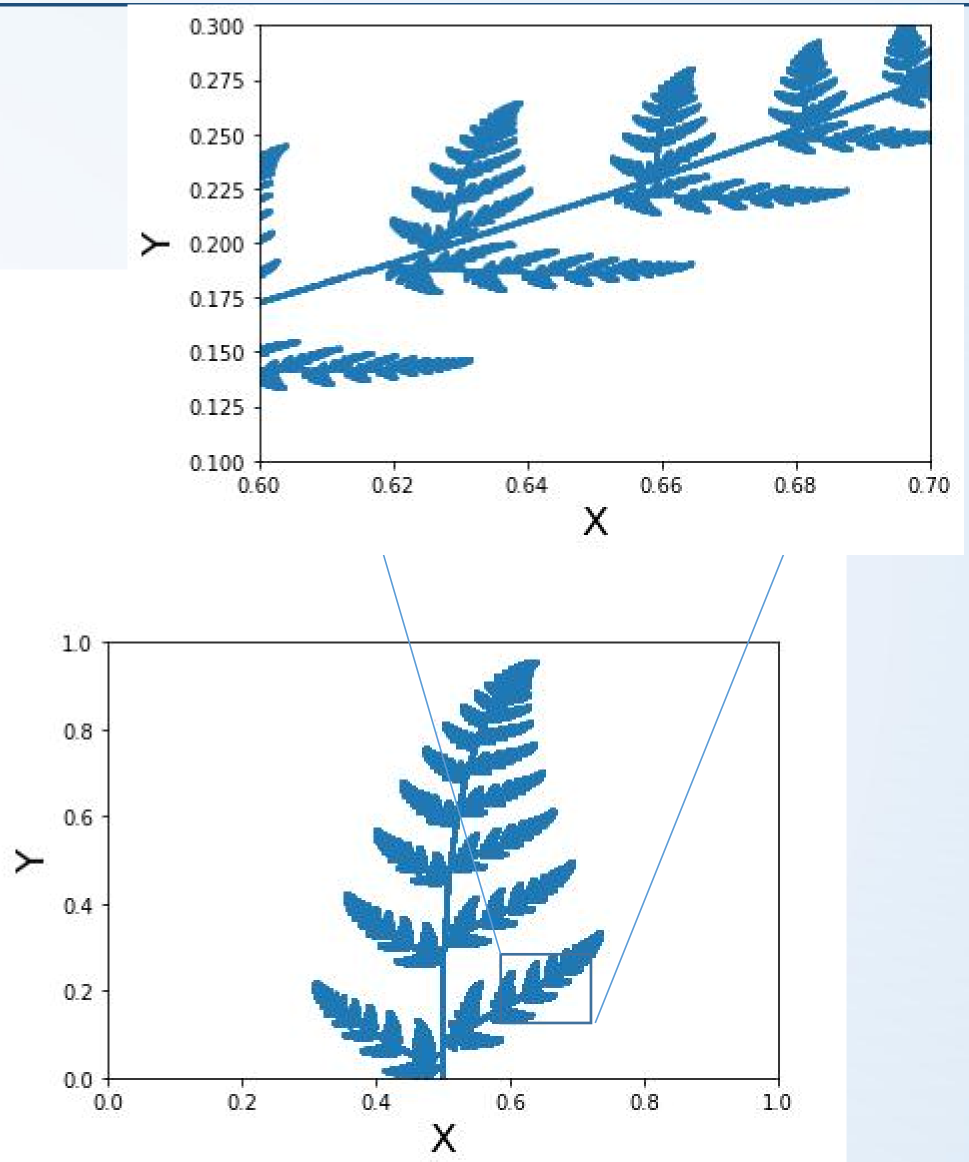
- 羊齿叶：引入模块，设置初始条件

```
import numpy as np
import matplotlib.pylab as pl
import random
nrun = 1000000
xx = np.zeros(nrun)
yy = np.zeros(nrun)
xx[0] = 0.5
yy[0] = 0.0
```

2.2 经典分形图形

- 羊齿叶：画图展示

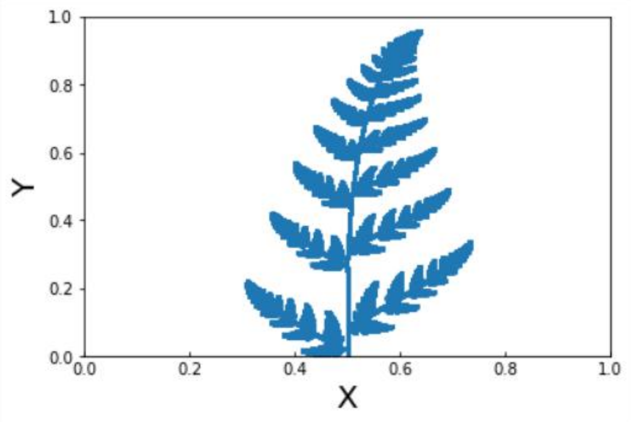
```
pl.plot(xx, yy, 's', markersize=1)
pl.xlim(0,1)
pl.ylim(0,1)
pl.xlabel(r'X', fontsize=20)
pl.ylabel(r'Y', fontsize=20)
pl.show()
```



2.2 经典分形图形

- 羊齿叶：运行全界面

```python
import numpy as np
import matplotlib.pylab as pl
import random

nrun = 1000000
xx = np.zeros(nrun)
yy = np.zeros(nrun)

xx[0] = 0.5
yy[0] = 0.0

for irun in range(1,nrun):

    rand = random.random()
    if rand < 0.02:
       xx[irun] = 0.5
       yy[irun] = 0.27*yy[irun-1]
    elif rand >= 0.02 and rand < 0.17:
       xx[irun] = -0.139*xx[irun-1]+0.263*yy[irun-1] + 0.57
       yy[irun] = 0.246*xx[irun-1]+0.224*yy[irun-1] -0.036
    elif rand >= 0.17 and rand < 0.3:
        xx[irun] = 0.17*xx[irun-1]-0.215*yy[irun-1] + 0.408
        yy[irun] = 0.222*xx[irun-1]+0.176*yy[irun-1] -0.0893
    else:
        xx[irun] = 0.781*xx[irun-1]+0.034*yy[irun-1] + 0.1075
        yy[irun] = -0.032*xx[irun-1]+0.739*yy[irun-1] +0.27


pl.plot(xx, yy, 's', markersize=1)
pl.xlim(0,1)
pl.ylim(0,1)
pl.xlabel(r'X', fontsize=20)
pl.ylabel(r'Y', fontsize=20)
pl.show()
```




```
In [49]: S
```

2.3 迭代与递归

- 递归：

1、定义

递归是一种计算机算法，它是程序调用自身的编程技巧，在程序设计语言中广泛应用。一个过程或函数在其定义或说明中有直接或间接调用自身的一种方法，它通常把一个大型复杂的问题层层转化为一个与原问题相似的规模较小的问题来求解

2、必要条件

（1）. 子问题须与原始问题为同样的事，且更为简单：

（2）. 不能无限制地调用本身，须有个出口，化简为非递归状况处理

2.3 迭代与递归

- 递归：

3、优点

1. 可以将大问题转化为小问题，减少代码量

2. 可以去掉不断重复的代码，使代码精简，提升可读性

4、缺点

递归调用浪费空间，递归太深还容易造成堆栈的溢出。

2.3 迭代与递归

- 迭代：

1、定义

迭代是重复反馈过程的活动，其目的是为了逼近或求得所需目标或结果。每一次对过程的重复称为一次“迭代”，而每一次迭代得到的结果会作为下一次迭代的初始值。重复执行一系列运算步骤，从前面的量依次求出后面的量的过程。

例如利用迭代法“求某一数学问题的解。对计算机特定程序中需要反复执行的子程序*一组指令)，进行一次重复，即重复执行程序中的循环，直到满足某条件为止，亦称为迭代。

2.3 迭代与递归

• 迭代：

2、必要条件

1. 问题可以转化为一个解决方案的重复操作，某次操作的输出是下次操作的输入. 3、

优点

（1）. 可以将重复问题转化为单一问题重复操作，减少代码量

（2）. 代码运行效率高，时间只因循环次数增加而增加，但是没有额外的空间开销：

4、缺点：代码不如递归简洁

2.3 迭代与递归

- 递归与迭代的联系

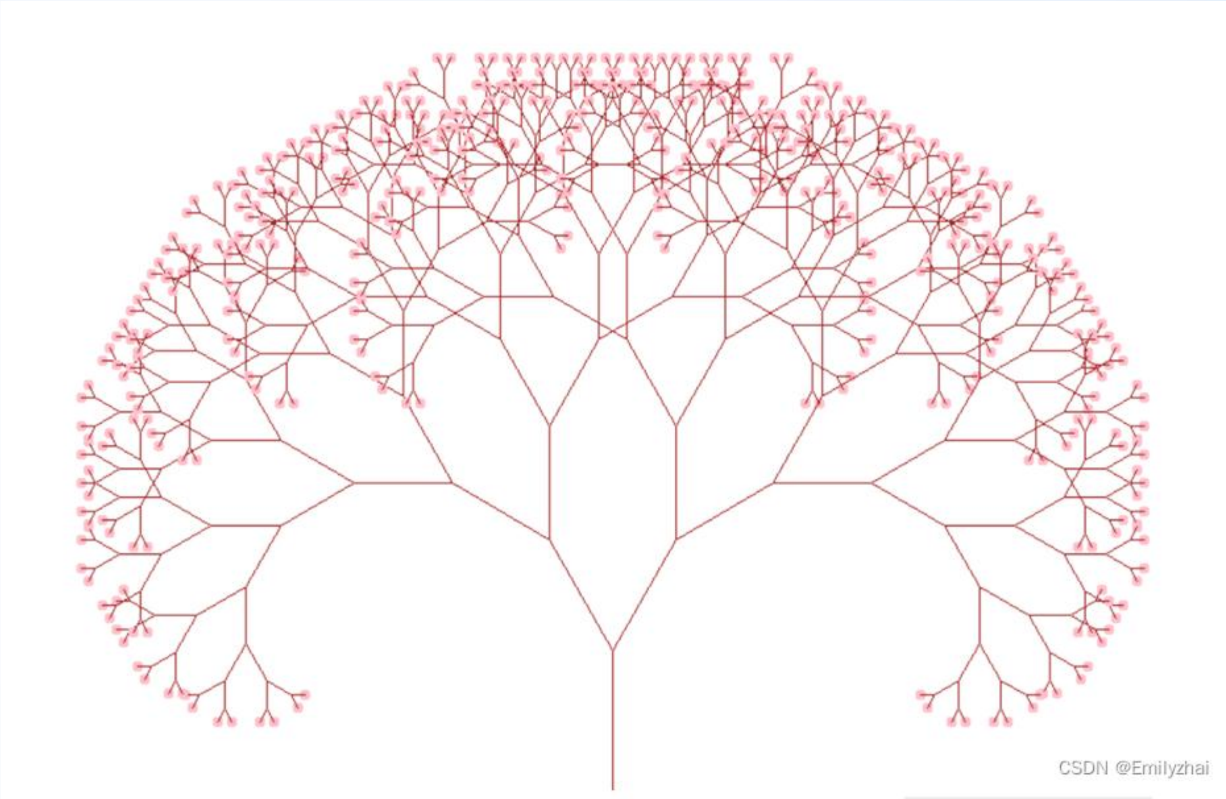
1. 递归和迭代都是循环，通过不断重复执行某些操作，从而达到我们要求的结果。

2. 他们都是将复杂重复问题简化，通过反复调用实现复杂功能。

2.3 迭代与递归

- 四、递归与迭代的区别
- 1，递归是不断调用自身，假设递归自己是一个函数的话，那递归的返回值会当作参数再次传入自身。迭代是将某一个初值设定不断放入某一个盾环体，得到的值成为一个新值再次放入循环体中，通过盾环体中的操作，逐步得到我们想要的结果。总结一下就是，程序调用自身称为递归，利用变量的原值推出新值称为迭代。

- 四、递归与迭代的区别



2.4 分形与分形维

- 1. 定义及来源

部分与整体以某种方式相似的形体被称为分形(fractal) 。1975 年，Mandelbrot 创立了分形几何学(fractalgeometry) 。在此基础上，形成了研究分形性质及其应用的科学，称为分形理论(fractaltheory)。

具有分形结构的物理系统有：动力系统的 相空间(或子集)结构， 一些系统的能谱，真实物理系统(浸透、晶体生长)。

2.4 分形与分形维

- **2. 发展历程：Mandelbrot**
- 1967年，在美国《科学》杂志上发表“英国的海岸线有多长”提出了分形的概念。
- 1973年，在法兰西学院讲课时，首次提出了分维和分形几何的设想。
- 1975年，用法文出版了分形几何的第一部著作《分形：形状、机遇和维数》。
- 1977年该书再次用英文出版。它集中了1975年以前Mandelbrot 关于分形几何 的主要思想，它将分形定义为豪斯道夫维数严格大于其拓扑维数的集合，总结了根据自相似性计算实验维数的方法。
- 1982年，出版《自然界的分形几何》，将分形定义为局部以某种方式与整体 相似的集，重新讨论盒维数。
- 2004年，出版《分形与混沌：Mandelbrot 集与其他》。这是一本论文集，收集 了直到2003年的许多重要研究论文，其中有些是原始的未发表过的论文。

2.4 分形与分形维

- 3. 分形理论的意义

一是分形整体与局部形态的相似， 启发人们通过认识部分来认识整体，从有限中认识无限；

二是分形揭示了介于整体与部分、有序与无序、复杂与简单之间的新形态、新秩序；

三是分形从一特定层面揭示了世界普遍联系和统一的图景。

- 4. 分形维

在分形理论中，是用分形维数(简称分维)，来定量描述分形结构自相似程度、不规则程度或破碎程度。

对于分形，和普通维数(0, 1, 2, 3)相对应的维数称为分形维数。通常，它们的维数值不是整数。

对于不同的对象，分维有不同的计算方法：

豪斯道夫维数，相似维数，盒维数

$\underbrace{\text{豪斯道夫维数，相似维数}}_{\text{完全自相似结构}}$

完全自相似结构

不是完全自相似：如海岸线

2.4 分形与分形维

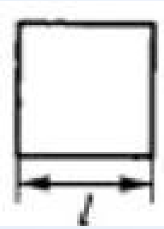
- 4. 1豪斯道夫维数

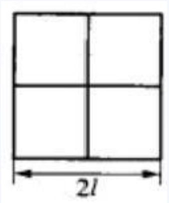
取长度为l 的线段，放大2倍后的长度21。边长为l 的正方形，每边长放大2倍 的面积为$4=l^2$ 。边长为l 的立方体，每边长放大2倍的体积为$8=l^3$ 。如所示，结果整理如下：

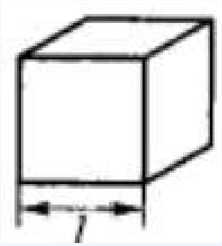
一维图形(线段) $2^1=2$

二维图形(正方形) $2^2=4$

三维图形(立方体) $2^3=8$







2.4 分形与分形维

- 4.1豪斯道夫维数
- $L^D=K$，$D=1,2,3$
- 取对数

$$D_f = D = \frac{\log K}{\log L}$$

这就是豪斯道夫维数。

推论：对于正规几何图形，分子为分母整除，D_f 为整数，是欧几里得维数。 对非规则图形，分子与分母不是总可以整除，D_f一般是分数，称为分维。

- 4.1豪斯道夫维数

例1 以 Sierpinski三角形为例，设想从一个小三角形开始，将每边扩大2倍，得与之相似的大三角形，面积为小三角形4倍。将中间一个小三角形舍去，实际面积为小三角形的3倍。按定义计算分维数

$$D_f = \frac{\log K}{\log L} = \frac{\log 3}{\log 3} = 1.5849$$

勤动手，多思考！

苏湘宁

邮箱：suxn@hainanu.edu.cn