

# 计算物理作业十五

于浩然 PB19020634 2021.12.18

## 1 作业题目

以  $x_{n+1} = \lambda \sin(\pi x_n)$  为迭代方程进行迭代：

- (1) 画出系统状态随参数  $\lambda$  的变化图，要求在图中体现出定值状态、倍周期分叉和混沌状态。
- (2) 列出各个倍周期分叉处的  $\lambda$  值，求相应的 Feigenbaum 常数。

## 2 算法简介

### 2.1 相关概念

对于如题目所给的以  $\lambda$  为参数的非线性函数的迭代，随着参数  $\lambda$  的增大系统有如下几种状态：

- 绝灭：即  $x$  最终趋于零。
- 定态：系统具有一定值，在图上呈现为一条线。
- 倍周期分叉：曲线一分为二，对应着周期为 2 的解的出现；参数进一步增大时，周期数再次加倍；随参数增大分叉出现得越来越快。
- 混沌：当参数足够大时，体系进入永不落入定态的涨落态，可认为周期为无穷大，轨迹点永不重复。
- 周期窗口：在总的混沌范围内出现的具有规律周期的小窗口，这种结构具有无穷嵌套性。

**Feigenbaum 常数**用于描述迭代混沌中倍周期分叉的标度行为，其中

- $\delta$  用于描述横轴方向倍周期分叉的标度行为：

设第  $m$  个分叉处  $\lambda$  值为  $\lambda_m$ ，前后分叉间距的比值趋于一个常数：

$$\lim_{m \rightarrow \infty} (\lambda_m - \lambda_{m-1}) / (\lambda_{m+1} - \lambda_m) = \delta = 4.669201 \quad (1)$$

- $\alpha$  用于描述纵轴方向倍周期分叉的标度行为：

在图上作  $x_n = 1/2$  的直线与放大的分叉曲线相交，设在第  $m$  个分叉处纵向间距为  $d_m$ ，则

$$\lim_{m \rightarrow \infty} d_m / d_{m+1} = \alpha = 2.502908 \quad (2)$$

## 2.2 具体方法

对于足够密集取的  $\lambda$  值, 由给定初值  $x_0$  计算迭代  $N - m, N - m + 1, \dots, N$  次的结果  $x_n$ , 以  $\lambda$  为横轴, 以  $x_n$  为纵轴, 画出散点图, 恰当取参数便能体现定值、倍周期分叉和混沌状态.

得到图像后容易找出各分叉点处的  $\lambda$  值, 由此可以计算 Feigenbaum 常数值.

## 3 编程实现

使用 FORTRAN90 进行编程, 通过简单的函数迭代得到迭代结果  $x_n$  和对应  $\lambda$  值的数组. 不妨取  $\lambda$  步进值  $\text{step} = 0.00001$ , 共取  $\text{lnum} = 100000$  个, 即  $\lambda \in [0, 1]$ ; 对每个  $\lambda$  值迭代  $\text{ites} = 10000$  次, 并取最后的  $\text{counts} = 30$  个数记录至文件并随后绘图.

由于能够辨认的分叉数极其有限, 只需辨认并考虑图中前几个分叉处的坐标值, 并可由此直接计算相应的 Feigenbaum 常数.

## 4 计算结果

### 4.1 系统状态随参数 $\lambda$ 变化图

$\lambda \in [0, 5]$  的图像总体如下, 显然从图中可看到 2.1 中所讲的各种状态:

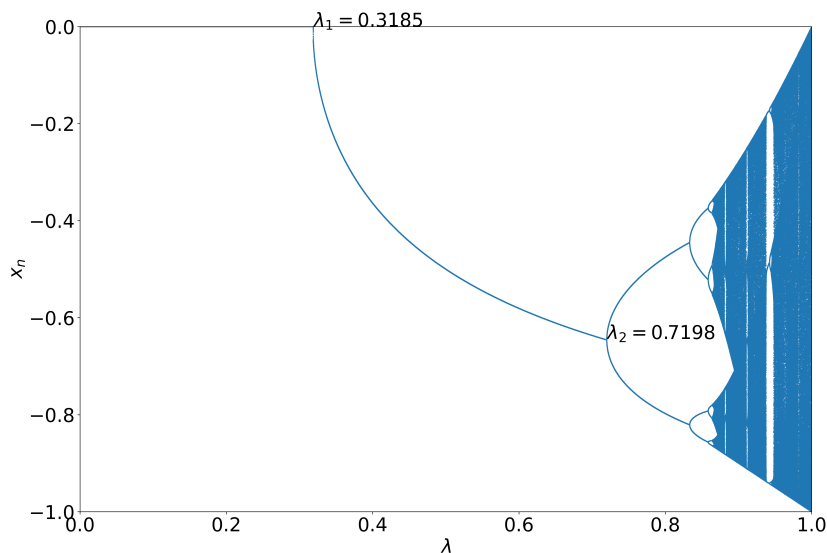


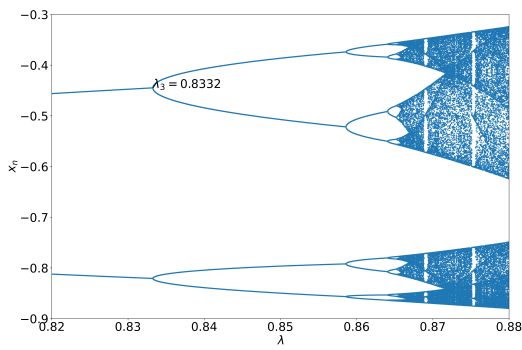
图 1: 系统状态随  $\lambda$  变化图

- 当  $\lambda \in (0, 0.318)$  时, 系统处于绝灭态, 迭代结果均为零;

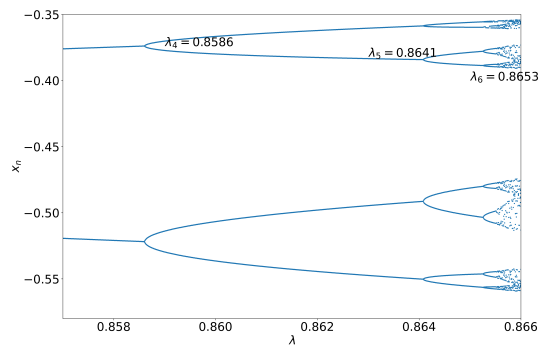
- 当  $\lambda \in (0.318, 0.720)$  时, 系统处于定态, 这时迭代最终趋于一个不为零的唯一结果;
- 当  $\lambda \in (0.720, 0.870)$  时, 系统进入了倍周期分叉, 分叉速度越来越快, 最终到达混沌;
- 当  $\lambda > 0.870$  时, 系统进入混沌, 这时仍能看到规律性的周期窗口.

## 4.2 图像缩放并寻找分叉点

我们从上面图中通过缩放已经将前两个分叉点的坐标确定到小数点后四位, 还要进一步确定其他的分叉点, 为了直观不妨取图二中  $\lambda \in (0.82, 0.88)$ ,  $x_n \in (-0.9, -0.3)$  区域进行缩放, 再选取  $\lambda \in (0.857, 0.866)$ ,  $x_n \in (-0.58, -0.35)$  区域, 读取  $\lambda_m$  值如下图:



(a)  $\lambda \in (0.82, 0.88)$



(b)  $\lambda \in (0.857, 0.866)$

图 2: 系统状态随  $\lambda$  变化图

根据这些点即可计算 Feigenbaum 常数.

### 4.3 Feigenbaum 常数计算

将各分叉处  $\lambda$  值列表如下:

表 1:  $\lambda_m$  值与计算所得  $\delta$  值表

$m$	1	2	3	4	5	6
$\lambda_m$	0.3185	0.7198	0.8332	0.8586	0.8641	0.8653
$\lambda_m - \lambda_{m-1}$		0.4013	0.1134	0.0254	0.0055	0.0012
$(\lambda_m - \lambda_{m-1})/(\lambda_{m+1} - \lambda_m)$		3.539	4.465	4.618	4.583	

与  $\delta$  理论值比较图如下: 由图可得, 除第一个点外计算结果较好.

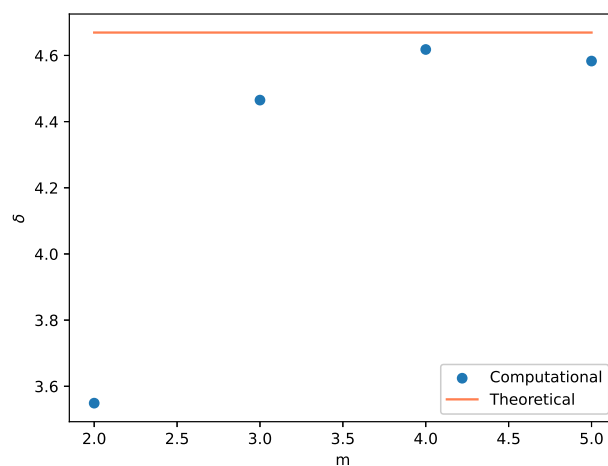


图 3:  $\delta$  计算值情况

在图 1 基础上作  $x_n = 1/2$  直线, 与曲线相交如下图:

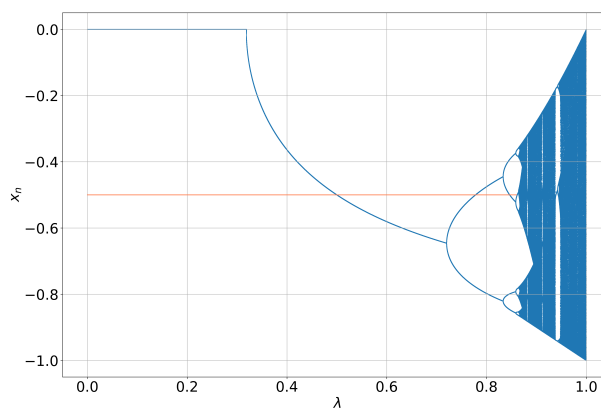


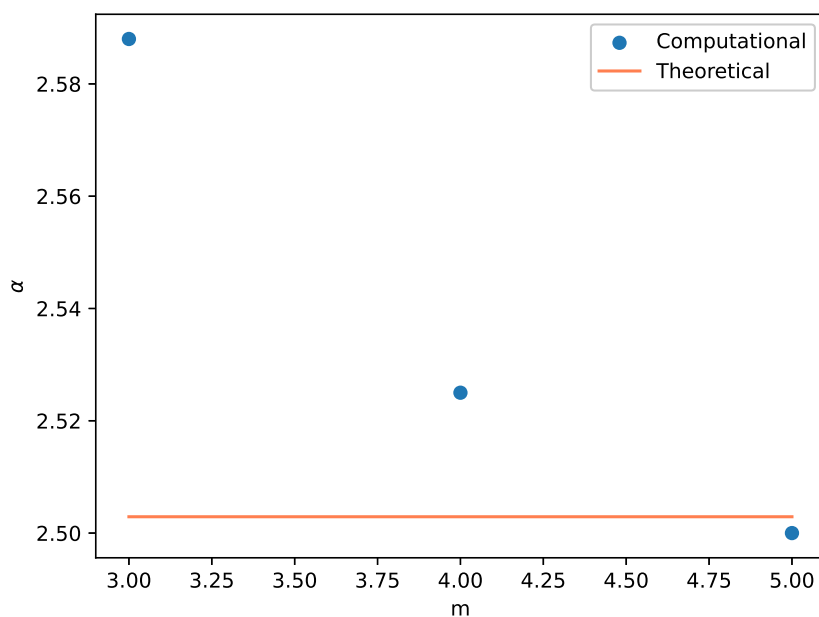
图 4:  $x_n = 1/2$  与曲线相交图

从图中读取交点并读取  $d_m$  计算  $\alpha$  值列表如下:

表 2: 计算所得  $\alpha$  值表

$m$	3	4	5	6
$d_m$	0.2777	0.1073	0.0425	0.0170
$d_m/d_{m+1}$	2.588	2.525	2.5000	

与理论值比较如下图:

图 5:  $\alpha$  计算值情况

可见计算的精度不高, 也算能够接受.

## 5 结论

本题中模拟了迭代函数体系的混沌状态, 通过直观的图像加深了对其的理解. 计算的精度受到 `step` 取值的约束 (本题中 `step=0.00001`), 在加上读取数据造成的误差, 导致计算的精度不够高. 此外, 对于这种分形的结构, 计算机的浮点运算在缩放的区域内误差将会不可控制, 同样需要考虑.

## 6 源代码

Fortran90 代码如下:

```
1  MODULE CHAO
2  IMPLICIT NONE
3  REAL(KIND=8) :: pi = 3.1415927
4  CONTAINS
5  REAL(KIND=8) FUNCTION y(lambda, x)
6      REAL(KIND=8), INTENT(IN) :: lambda, x
7      y = lambda * SIN(pi * x)
8  END FUNCTION y
9  SUBROUTINE Iteration(step, ites, lnum, counts, filename)
10     REAL(KIND=8), INTENT(IN) :: step
11     INTEGER(KIND=4), INTENT(IN) :: lnum, ites, counts
12     CHARACTER(LEN=*), INTENT(IN) :: filename
13     REAL(KIND=8) :: lambda, x, sol(lnum*counts, 2)
14     INTEGER(KIND=4) :: i, j
15     lambda = 0
16     DO i = 1, lnum
17         x = 1
18         DO j = 1, ites - counts
19             x = y(lambda, x)
20         END DO
21         DO j = 1, counts
22             x = y(lambda, x)
23             sol((i-1)*counts + j, 1) = lambda
24             sol((i-1)*counts + j, 2) = x
25         END DO
26         lambda = lambda + step
27     END DO
28     OPEN (1, file=filename)
29     WRITE (1, *) sol
30     CLOSE (1)
31 END SUBROUTINE Iteration
32 END MODULE CHAO
33
34 PROGRAM MAIN
35     USE CHAO
36     IMPLICIT NONE
37     CALL Iteration(0.00001_8, 10000, 100000, 30, 'sol.dat')
38 END PROGRAM MAIN
```