计算物理 A——Homework 16

何金铭 PB21020660

1 题目描述

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 λ 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 λ 值，求相应的 $Feigenbaum$ 常数。

2 理论分析

2.1 收敛的判据

由于计算机的物理限制，系统不可能严格达到不动点，于是定义当：

$$\left| \frac{x_2 - x_1}{x_1} \right| < \varepsilon \tag{1}$$

则称 x_2 收敛于 x_1 $(x_2 \to x_1)$，即得到了一个不动点

2.2 定值状态

当 λ 在一定范围内取值时，可以让系统迭代足够长时间后，系统都只有一个定态 $x^\star$
对于此情况，可以数学分析得：$\lambda \leq \dfrac{1}{\pi}$ 必然成立，且不动点 $x^\star = 0$。

2.3 倍周期分叉

当参数 λ 在一定范围内取值时，可以让系统迭代足够长时间后，系统会出现一些准周期的结构。

一开始会出现周期为 k 的解，当参数进一步增高时，周期会加倍，加倍 m 次后变为 $2^m k$ 次的周期解。

2.4 混沌状态

当参数 λ 再增加的时候，则系统会进入混沌状态，不存在任何周期状态。

2.5 Feigenbaum 常数

定义分岔的编号为 m，第 m 次分岔值为 λ_m，Feigenbaum 证明：

$$\delta = \frac{\lambda_m - \lambda_{m-1}}{\lambda_{m+1} - \lambda_m} = 4.669201609102990671853203 8 \quad (m \to \infty) \tag{2}$$

给定一条直线 $x = x_0$ 和放大的分岔曲线相交于各点,其横坐标为 $R_1, R_2, R_3 ...$,得到 $R_1, R_2, R_3 ...$ 等分岔纵向间距 $d_1, d_2, d_3 ...$ 等值，Feigenbaum 发现：

$$\frac{d_m}{d_{m+1}} \to \alpha = 2.502907875095892822283902 87 \quad (m \to \infty) \tag{3}$$

2.6 定性分析

由于对迭代方程 $x_{n+1} = \lambda \sin \pi x_n$ 的性质不了解，这里先用计算机作图分析。取 $x_0 = \dfrac{1}{2}, \lambda \in [0, 1]$ 且 $\lambda = 0.1, 0.2, ..., 0.9, 1$，迭代次数 $N = 10^5$，分析图像。

1. 发现当 $\lambda \le 0.7$ 时只有一个不动点
2. 当 $\lambda = 0.8$ 时，有一个周期为二的解
3. 当 $\lambda = 0.9, 1.0$ 时，可能为混沌状态 (步数 $N = 10^4$)

部分结果如下图所示：

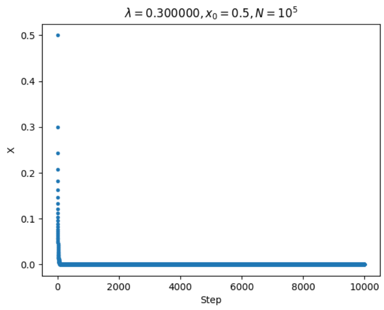

图 1: $\lambda = 0.3$ 时的迭代结果，此时有一个不动点

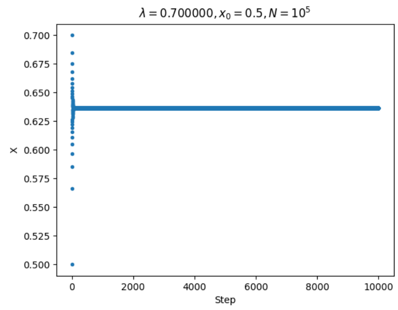

图 2: $\lambda = 0.7$ 时的迭代结果，此时有一个不动点

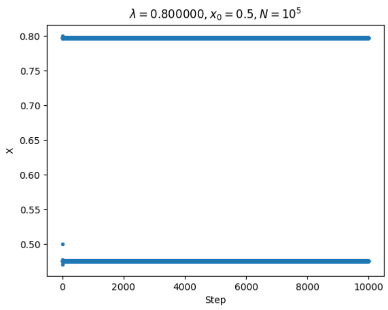

图 3: $\lambda = 0.8$ 时的迭代结果，此时有一个周期为 2 的解

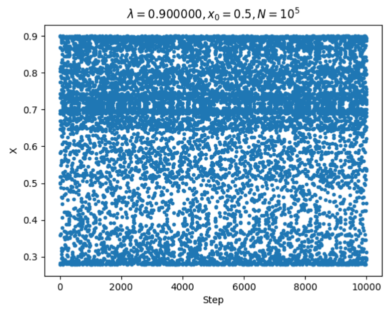

图 4: $\lambda = 0.9$ 时的迭代结果，此时可能为混沌状态

进一步改变其步长，可以大致观察得：

1. $\lambda \in [0.71, 0.72]$ 之间，由一个不动点变为了一个周期为 2 的解。
2. $\lambda \in [0.87, 0.88]$ 之间，由准周期状态变为了混沌态。

这里就只给出观察结果，不放置图片，助教可以自行修改参数得到。
观察得：此迭代系统所有可能的周期解为 $2, 2^2, 2^3, 2^4 ...$
这为之后的编程提供了便利。

3 算法模拟

3.1 讨论时一些量的取值

1. 迭代的初始点为 $x_0 = 0.3$
2. 在下面的讨论中取对于每一个 λ 的迭代次数为 10^5。
3. λ 讨论的区间是 $\lambda \in [-5, 5]$
4. 结合之前的定性观察图片，确定倍周期分叉只能是 $2^k (k \in \mathbb{N})$ 的形式，这里取 $k \leq 10$，即 $x^\star$ 的值最多取到 $2^{10} = 1024$ 个，这一点在之后的作图中会有所体现。
5. 对于大部分的 λ 值，从每次迭代的第 1000 次开始寻找 $x^\star$，到 $N = 8500$ 为止停止寻找。
6. 选取收敛满足的条件为 $\left|\dfrac{x_2 - x_1}{x_1}\right| < \varepsilon$，其中取 $\varepsilon = 10^{-5}$，实际在程序中，由于一些早期的尝试，发现 $\lambda \in [-0.1667, 0.1667]$ 的时候收敛较快，每次循环从第 100 次开始寻找 $x^\star$，到 $N = 5000$ 为止。

3.2 程序大致过程

1. 遍历 $\lambda \in [-5, 5]$，步长为 step。对于每一个 λ，代入初始点 x_0 迭代计算 10^5 次，储存于数组中。
2. 对数组搜索，对于 $k \in [0, 10]\,(k \in \mathbb{N})$ 的顺序，遍历数组的一部分，判断是否收敛，返回数组长度与所有的 $x^\star$
3. **特别的**，对于 $\lambda \in [-0.16670, 0.16670]$ 这一收敛速度快的部分，每次判断是否收敛的过程中，每次选取开始搜索的位置提前 $num = 10$，每次的 $\varepsilon = \varepsilon \times k = \varepsilon \times 10$，这里取 $k = 10$ 为误差系数。

4 程序说明

4.1 主要程序

chaos.c 混沌主程序。把所有功能写进了一个文件中的多个函数内

- **double func(double lambda,double x)** 迭代方程 $x_{n+1} = \lambda \sin \pi x_n$ 的表达式，返回函数值
- **double* iteration(double lambda)** 迭代循环，对于 $x_0 = 0.3$ 迭代 10^5 次，返回迭代数组
- **int charge(double x__1,double x__2,double k)** k 为误差放大系数（见上文）判别有无收敛，若收敛，返回 1；否则，返回 0
- **double abs__d(double a)** 一个 double 型绝对值函数，返回 a 的绝对值
- **double* search(double *x,double kk,double lambda,int num)** 寻找所有的满足条件的 2^k 个 $x^\star$，并返回长度 2^k 以及所有的 $x^\star$
- **void chaos()** 一个用于综合所有功能的函数，包括调用其他函数和读写程序，并且输出分岔点处的 λ。

4.2 程序结果

chaos.exe 混沌主程序。把所有功能写进了一个文件中的多个函数内，**编译时可以手动修改参数**

./data 文件夹路径，里面存放了各种导出的数据（由于其他文件过大，这里只放几个小文件，其他需要助教生成，**当** $step$ **过小时可能会运行过长时间**）

out_1e_4.txt 输出的 $step = 10^{-4}$ 时所有的 λ 和 $x^{\star}$ 的值，第一列为 λ，第二列为 $x^{\star}$

f_out_1e_4.csv 输出的 $step = 10^{-4}$ 所有的 λ 以及对应的分岔前的数目和分岔后的数目。

./pic 文件夹路径, 里面存放了各种由数据转来的图片，其具体含义请参见报告中的内容。

4.3 其他说明

1. 数据都写于 CSV 和 TXT 文件中
2. 其中 Python 程序用到的库有:

- matplotlib.pyplot：用于作图
- numpy：用于数据处理
- csv：用于读写 CSV 文件

5 结果分析

5.1 图像展示

5.1.1 $step = 10^{-3}$

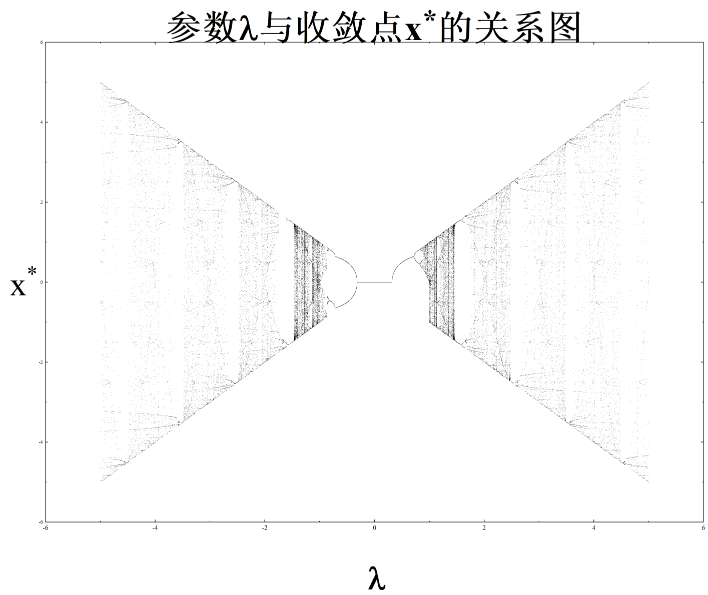

图 5: $step = 10^{-3}, x_0 = 0.3$ 时的 λ-$x^{\star}$ 图

5.1.2 $step = 10^{-4}$

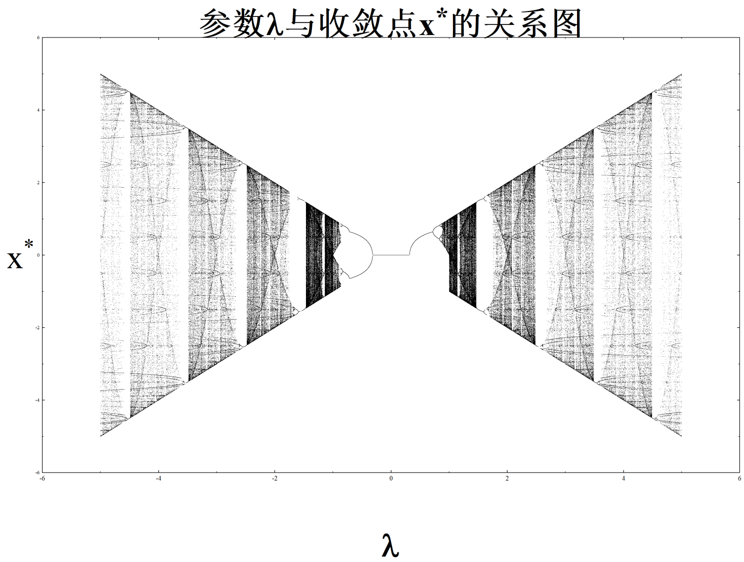

图 6: $step = 10^{-4}, x_0 = 0.3$ 时的 λ-$x^\star$ 图

5.1.3 $step = 10^{-5}$

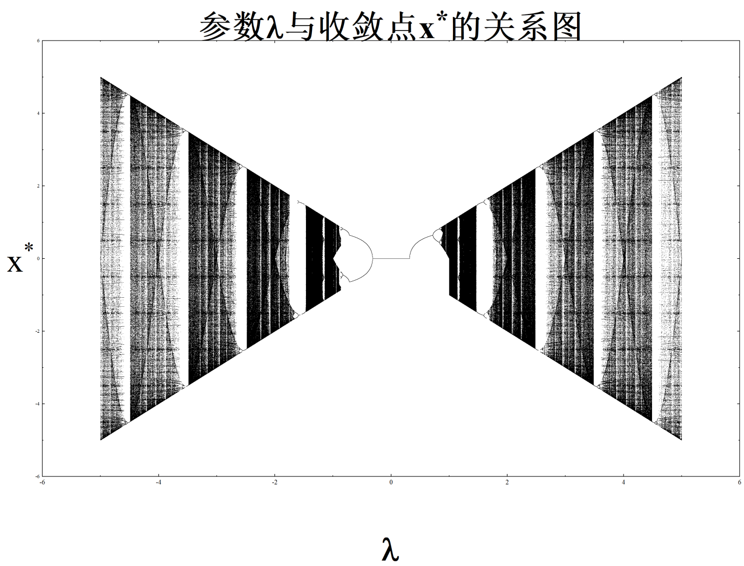

图 7: $step = 10^{-5}, x_0 = 0.3$ 时的 λ-$x^\star$ 图

5.1.4 小结

发现当 $step$ 越来越小的时候，发现 λ-$x^\star$ 图变得越来越密集，并且可以明显的看出其定值状态、倍周期分叉和混沌状态。

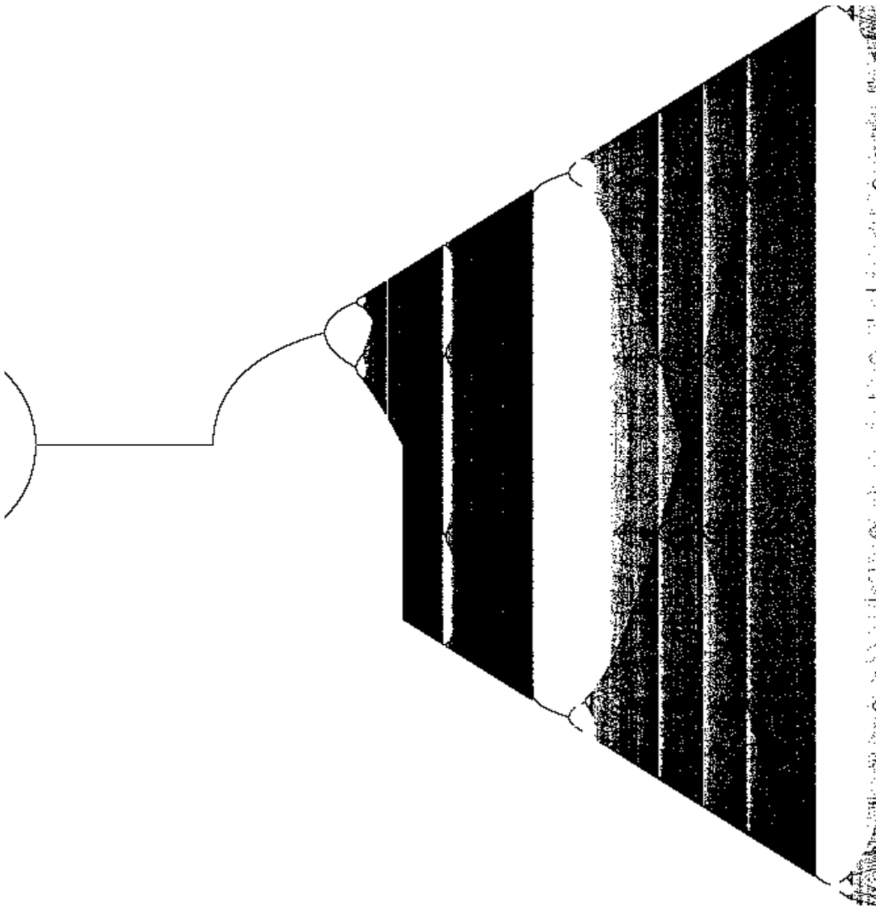

图 8: $step = 10^{-5}, x_0 = 0.3$ 时的 λ-$x^\star$ 局部放大图

并且发现图像中存在一些镂空的部分，这些部分是由于没有完全计算每个 λ 处，周期大于 2^{10} 的点。

5.2 Feigenbaum 常数的计算

可以计算 *step* 较大时的 Feigenbaum 常数，但是可能由于精度过低，意义不大，这里跳过。

由于 *step* 较小时作图计算量较大，这里就手动取区间来计算分岔处的 λ 值。

分岔点 m	λ
1	0.7199432
2	0.8332602
3	0.8586074
4	0.8640840
5	0.8652594
6	0.8655128

表 1: 各个分岔点的 λ_m

通过式 $\delta = \dfrac{\lambda_m - \lambda_{m-1}}{\lambda_{m+1} - \lambda_m}$ 得：

分岔点 m	δ	$\frac{\lvert\delta_0-\delta\rvert}{\delta_0}$
2	4.470592412574168	4.2536%
3	4.6282730161049335	0.8766%
4	4.6593500085075	0.2110%
5	4.638516179953583	0.6572%

表 2: 各个分岔点的 δ 值的估计以及误差表

发现一开始，随着分岔的变多，$\delta \to \delta_0$，误差越来越小。但当 $m = 5$ 时，误差又开始变大，可能的原因是，随着步长的减少，结果受双精度浮点数的计算误差的影响。

6 总结

1. 得到的系统状态与参数 λ 的关系中可以体现出定值状态、倍周期分叉和混沌状态。
2. 得到的 Feigenbaum 常数约为 4.6593500085075
3. 在这次作业中，第一次写和混沌有关的程序，遇到了内存没有及时清除的问题，增加了编程经验。