计算物理 hw17

PB18020616 李明达

December 2020

摘要

这是计算物理第 17 次作业，作业题目是以 $x_{n+1} = \lambda sin(\pi x)$ 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。

1 算法和程序

1.1 算法

本题的思路是，先给定一个初始的 x_0，经过多步热化以后，此时会出现一个或多个不动点（或混沌），这时**舍去热化点**，从后面的点里找出各个 λ 值处的不动点，将其画在图上。

随后我们对这些点先用排序方法 (这里我们采用 quick sort 办法) 进行排序，然后统计有多少个不同的点来记录是二周期分叉，还是四，八，十六，三十二分叉，我的方法是：对特定 λ 值输出的 x，先进行顺序排序，然后比较相邻的两个点的大小，通过记录相差大于 10^{-5} 的相邻点的个数，来估计分叉点。举个例子，如果排序后的两个相邻点一个为 0.5，一个为 0.8，则说明两个点在两个“分支”里；而如果一个点是 0.500011，另一个是 0.500012，则说明两个点在同一个“分支”。用这种方法我们可以计算出分支的数量，如果分支是 2 个，则为二周期分叉，如果是 4 个则为四分叉，通过搜寻**临界点**，来找出分叉处，从而计算 Feigenbaum 常数。

对于这个函数，对于 $\lambda = 0.4$ 的情况，可以用图像法画出来两个不动点，如图 1所示

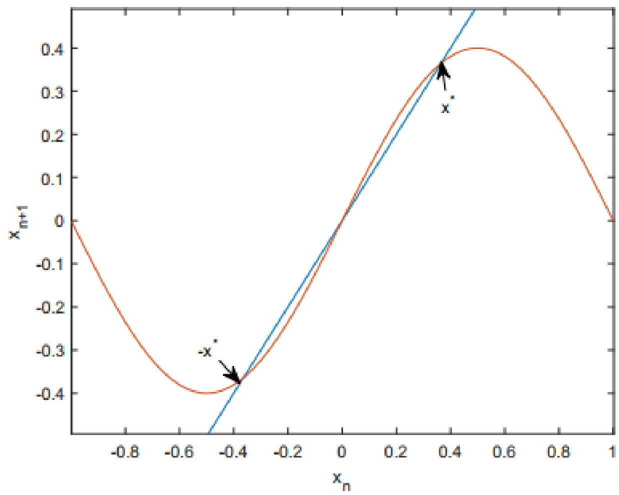

图 1: $\lambda = 0.4$ 时的不动点

1.2 程序

我的程序包括三个部分：第一个部分是随机数产生器，这个是为了避免初始值不同而对图像的影响；第二个部分是 display 和 QuickSort 函数，用来对数组进行快速排序；第三个部分用来计算不同 λ 的不动点，并且在 quicksort 之后，计算不同 λ 值的时候的分叉个数。

1.2.1 随机数产生器

我们仍然使用 Schrage 随机数产生算法，这个和之前的都一样，所以不过多阐述。

1.2.2 display 和 QuickSort 函数

这两个函数参考网上已经成熟的代码，基本上是标准的了，不仅速度快，而且简单易懂。第一个函数用来把数组表示出来（用来检验是否排序成功），第二个函数用来对数组进行排序。

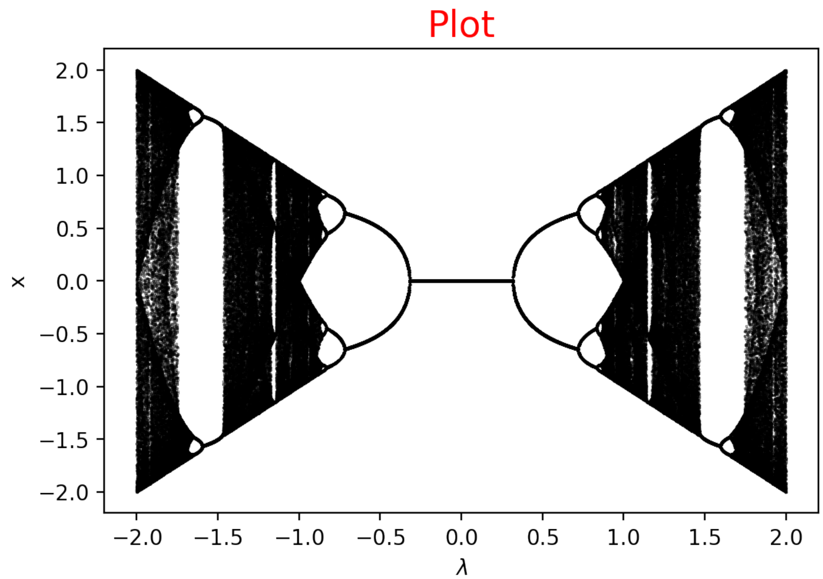

图 2: 系统状态随 λ 变化图

1.2.3 void diedai() 函数

diedai(float lambda) 函数的作用是在 lambda 值的时候求不动点，并且在 quicksort 之后，计算不同 lambda 值的时候的分叉个数，所使用的算法是我在 section 1.1 里提到的，通过“搜寻”临界值来找到分叉点。

2 实验结果

2.1 1．系统的状态随参数的变化图

通过上述的算法，我们取热化步数 N=1000，种子值在数据文件中。每一个 λ 我们用随机数生成 5 个初始值，分别对 5 个初始值寻找 20 步的迭代，所以每一个 λ 值有 100 个点的数据，画出来如图 2所示。

用随机数来产生初值是很重要的，我们会在最后讨论如果单一初值的情况。

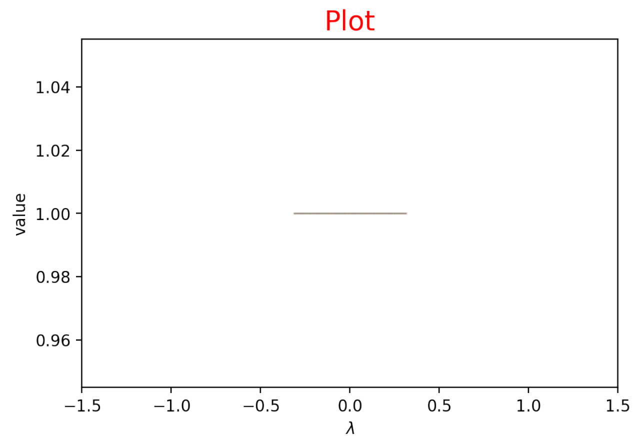

图 3: 定态时的 λ 值的区域。值为 1 代表有值，值为 0 代表没有值。可以看出，此时中间部分右界为 0.315037

2.2 2. 每个倍周期分叉处的 λ 值

通过我在 Section 1.1 里阐述的算法，我们可以得到定态（图 3）、二（图 4），四（图 5），八（图 6），十六（图 7），三十二（图 8）个分叉时 λ 的图。由于有四个分叉肯定有两个分叉，所以这些图是叉包含关系的，三十二分叉的图包含的 λ 值最多，而我们通过比较不同的图来找出分叉点的位置。这些图如下所示：

所以我们可以总结出，在定态、二、四、八、十六、三十二个分叉临界点 λ 的值为 0.315037、0.718032、0.832030、0.858030、0.863030、0.869030，可以计算成下表**??**.

表 1: 所计算得出的分叉点和 Feigenbaum 常数

分叉	λ	Feigenbaum 常数
定态	0.315037	null
二	0.718032	null
四	0.832030	4.665304
八	0.858030	4.384538
十六	0.863030	5.200000
三十二	0.869030	null

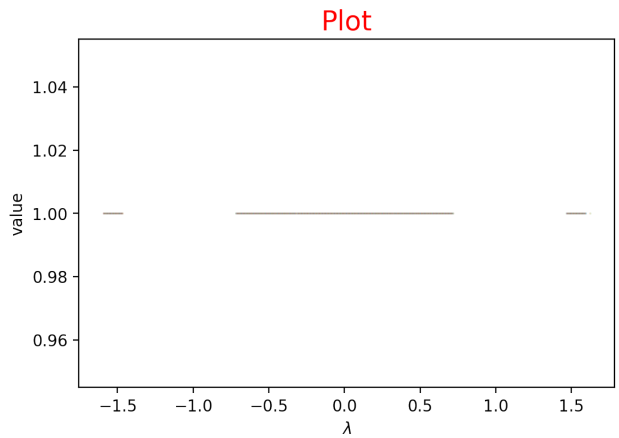

图 4: 有 2 个分叉时的 λ 值的区域。值为 1 代表有值，值为 0 代表没有值。可以看出，此时中间部分右界为 0.718032

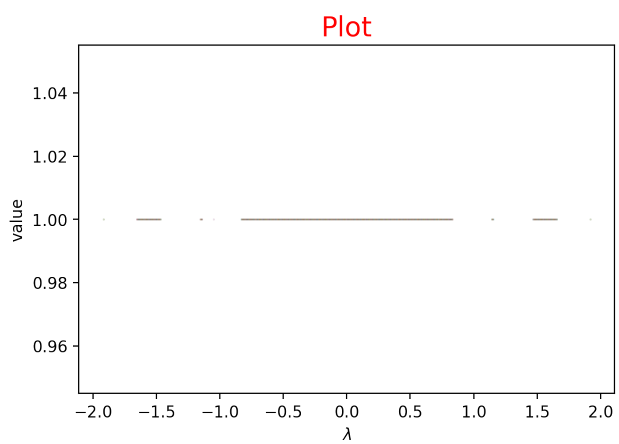

图 5: 有 4 个分叉时的 λ 值的区域。值为 1 代表有值，值为 0 代表没有值。可以看出，此时中间部分右界为 0.832030

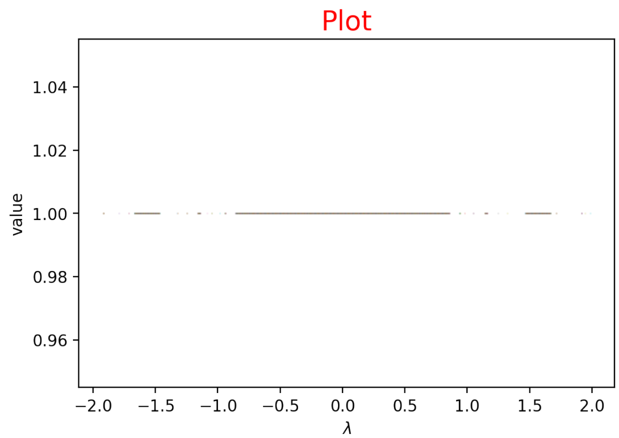

图 6: 有 8 个分叉时的 λ 值的区域。值为 1 代表有值，值为 0 代表没有值。可以看出，此时中间部分右界为 0.858030

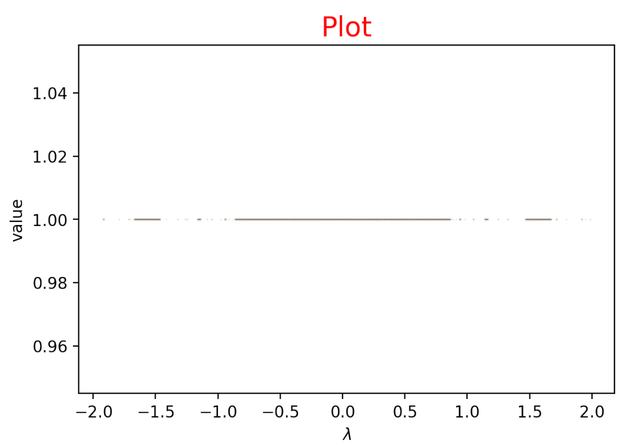

图 7: 有 16 个分叉时的 λ 值的区域。值为 1 代表有值，值为 0 代表没有值。可以看出，此时中间部分右界为 0.863030

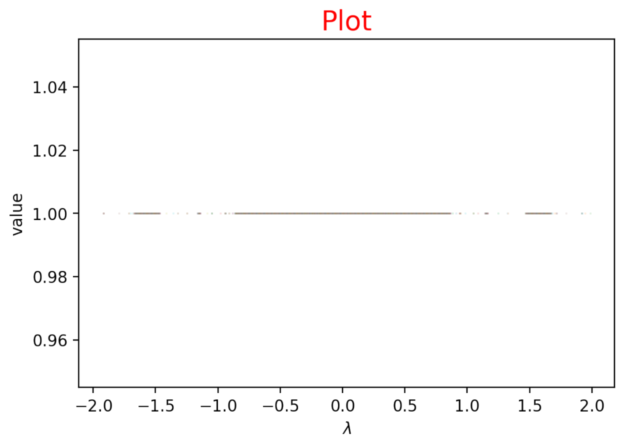

图 8: 有 32 个分叉时的 λ 值的区域。值为 1 代表有值，值为 0 代表没有值。可以看出，此时中间部分右界为 0.869030

因为每次我只取了 100 个，所以最多只能做到 64 分叉，但是这个误差会很高，所以我只做到了 32 分叉，前两个值相对比较准确，但最后一个值因为有限个采样，已经不准，因此可以确定 Feigenbaum 常数大概在 4.5 左右。

3 总结

本实验我按照作业要求，首先画出以 $x_{n+1} = \lambda sin(\pi x)$ 为迭代方程的系统状态随参数 λ 的变化图（图中包含了定值状态、倍周期分叉和混沌状态），然后通过我在 Section 1.1 列出的算法，计算了各个倍周期分叉处的 λ 值，据此求了相应的 Feigenbaum 常数。

4 进一步讨论

这里我想讨论一下随机取初值的重要性

一开始我只取了 $x_0 = 1$，但这时我发现，如果给定一个单一的值，在区间 $x \in [0.4, 1.0]$ 只能画出来上半支，如图 9，同时在-1.5 左右的值也是残缺的。所以为了找出所有的不动点，我们要用随机数产生器来进行初值的产生。

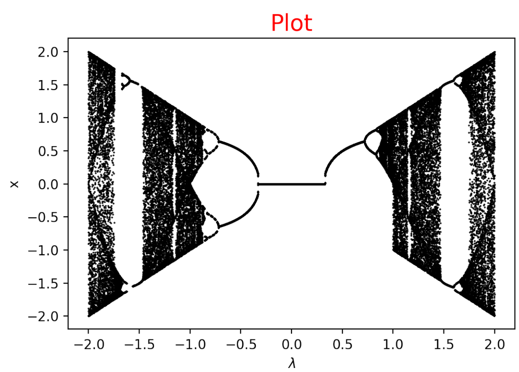

图 9: 如果只取单一的初值，可能画出来是残缺的图