

18.091 Lab 1

The Computer May Lie

Alejandro Ochoa

May 19, 2005

1 Introduction

In this experiment we investigated the orbits of several seeds under iteration of three functions. We will discuss how our data compares to what we expect from theory, which will lead to the discussion of computation as performed by a computer. The functions chosen are

1. $F[x] = x^2 - 2$ on the interval $(-2, 2)$,
2. $F[x] = x^2 - 2.1$ on the interval $(-2, 2)$,
3. $D[x] = \text{Frac}[2 * x]$, the doubling function on the interval $[0, 1)$.

2 Methods

Some of the seeds were chosen by me, while others were randomly generated by Mathematica using a function. Note that Mathematica is software capable of doing rational arithmetics, which implies the format in which we type the number matters. For example, typing 0.1 for a seed will make Mathematica do all the computations with 1/10 as a floating point number, with limited precision, while typing 1/10 will force Mathematica to make exact calculations with that rational number.

3 Data and Observations

For each function, on the table we listed in the first column the seeds used, and in the second column the observed behaviour of the orbits of the corresponding seed. Seeds with five or six significant digits were generated by Mathematica. Double quotation marks indicate the behaviour is the same as described above. Plots of sample orbits are also included for the first and third functions to better illustrate the behavior of such orbits.

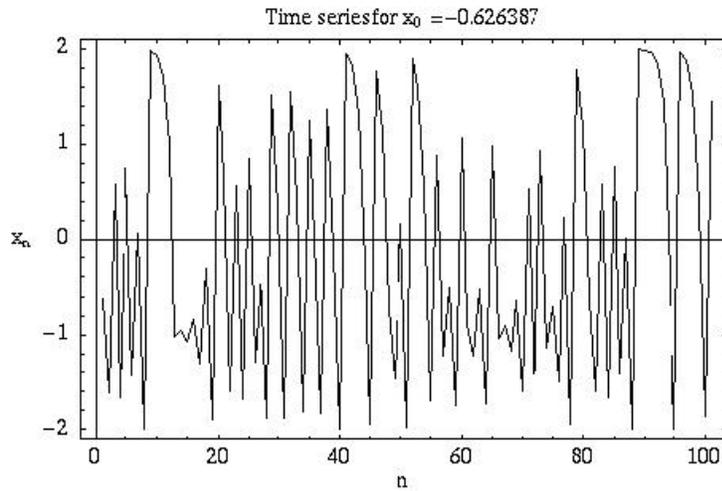


Figure 1: The orbit of a seed under 100 iterations of $F[x] = x^2 - 2$

1. $F[x] = x^2 - 2$,

Seed	Behavior
-1	Fixed point.
0.1	Apparently chaotic, bounded in $[-2, 2]$.
0.13	"
-0.0468936	"
-0.510909	"
-0.66461	"
0.706763	"
0.12377	"
-0.626387	"
-0.374533	"
0.499365	"

2. $F[x] = x^2 - 2.1$,

Seed	Behavior
-1	Diverges to ∞
1	"
0	"
0.25	"
1.985	"
-0.185049	"
-0.283494	"
-0.643112	"
-0.987201	"
0.252688	"

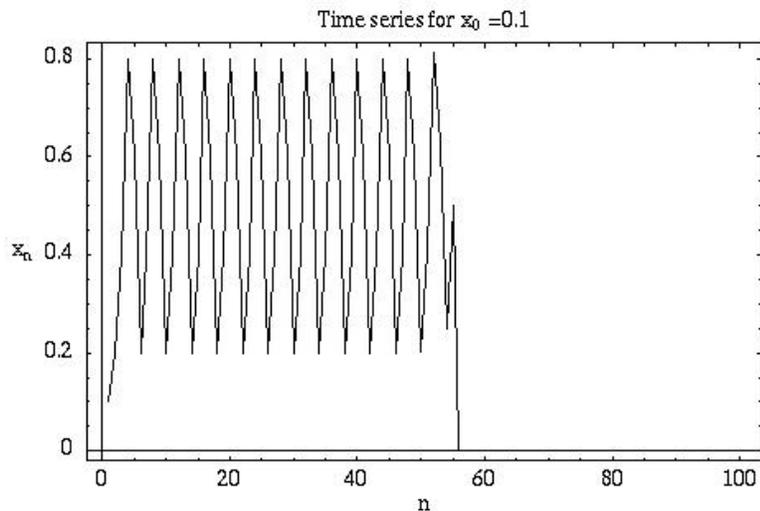


Figure 2: The orbit of the seed $x_0 = 0.1$ under 100 iterations of D

3. $D[x] = 2 * x - Floor[2 * x]$,

Seed	Behavior
0.1	Eventually fixed, converges to 0
0.0628138	"
0.860487	"
0.53821	"
0.762382	"
0.495847	"
0.392426	"
0.142857	"
0.33	"
1/7	Periodic
33/100	Eventually periodic

4 Discussion

For the function $F[x] = x^2 - 2$ the experiment with the computer suggests that all orbits are chaotic (see Figure 1), except we happened to select the point $x_0 = -1$ which is fixed under iteration by F . However, we can algebraically verify that the equation $F[x] = x$ has another solution, namely $x = 2$, so there are two fixed points for this function that the computer would not have found for us, for example, if it only sampled random numbers. We did not identify any periodic points either, while we know there are infinitely many from theory.

For the second function $F[x] = x^2 - 2.1$ the computer strongly suggests that all seeds in $(-2, 2)$ diverge to infinity, but this is not the case because the

equation $F[x] = x$, or $x^2 - x - 2.1 = 0$, has two solutions, namely

$$x_0 = \frac{1 \pm \sqrt{9.4}}{2},$$

and one of them, $x_0 \simeq -1.0330$ is in the interval we sampled. Hence, among all of the points that diverge to infinity, there is one that remains fixed and was not detected with the computer.

The function D is by far the most interesting case. Consider $x_0 = 0.1$, which we know has an exact representation as a rational number, namely $x_0 = 1/10$. The point $1/10$ is eventually periodic, entering the 4-cycle

$$\frac{1}{5}, \frac{2}{5}, \frac{4}{5}, \frac{3}{5}, \frac{1}{5}, \dots$$

after the first iteration. However, the computer shows that $x_0 = 0.1$ is eventually fixed, converging to zero. Looking at Figure 2, we can observe that 0.1s orbit starts with a seemingly regular orbit as we expect it because it enters a 4-cycle. However, something happens around the 50th iteration, the orbit becomes irregular and then drops to zero where it remains fixed.

It is plausible that the difference in the outcomes of the orbits of 0.1 and $1/10$ is due to the fact that the result of a computation on a floating-point number (the logical representation of decimals on a computer) is bound to lose precision because floating-point numbers only store a finite amount of digits, while the software that deals with rational arithmetics only needs to store integers and has an unlimited capacity to do so, always giving an exact answer. Although the loss of precision is very small, after 50 iterations of the function on 0.1 we can observe the loss of precision built up to give a wrong answer which is not reasonably close to the right answer.

For the first two examples, we were led to an incorrect hypothesis because we only sampled a finite amount of points. There are a dense and uncountable amount of points which share the behaviour of the points we sampled in those intervals. However, there are two and one points of interest respectively for each F on the given interval, which display the very appealing property of remaining fixed under iterations of F , and the probability that the computer would randomly choose these points is close to zero. The third example showed us that the loss of precision generated by storing numbers as floating-point numbers can become significant when analyzing orbits of seeds of chaotic systems.

Whenever we use computers to model dynamical systems, we must always keep in mind the two caveats we discovered, namely, that we should not generalize the behavior of an interval of seeds by analyzing just a few seeds, and that we should be careful when considering results from a large amount of computations on floating-point numbers, given that these results are not guaranteed to be neither correct nor approximately correct.