

Answers to P-Set # 01, (18.353/12.006/2.050)j

MIT (Fall 2023)

Rodolfo R. Rosales (MIT, Math. Dept., room 2-337, Cambridge, MA 02139)

September 18, 2023

Contents

1 Computer exercises with a 1-D map	1
1.1 Statement: Computer exercises with a 1-D map.	1
1.2 Answer: Computer exercises with a 1D map	3

List of Figures

1.1 Typical behavior near a period doubling	5
1.2 Chaos beyond c_∞	5

1 Computer exercises with a 1-D map

1.1 Statement: Computer exercises with a 1-D map.

The objective of this problem is to give an elementary first introduction to concepts such as: fixed point, stability, bifurcations, and chaos, via “experimental” (i.e., numerical) computation. We do this by using a very simple mathematical model. This model is highly abstract, but we will argue later in the course that the model is also (to a significant degree) representative of the behavior of many real systems. In addition, this assignment also introduces the kind of computing problems we shall often, but not always, assign.

In this model we will assume that the system is described by a single time-dependent variable, $x(t)$. This variable could represent, say, the globally averaged temperature on the Earth’s surface, the size of a particular population of animals on some secluded island (or in a Petri dish), some particular stock market average, etc. Furthermore, we will suppose that we are interested only in the values $x_n = x(t_n)$ at discrete times $t_n = n \Delta t$, where Δt is some suitable interval of time (say, a day). The evolution

of x in time may then be written as

$$\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n). \quad (1.1)$$

where \mathbf{F} is some function that describes the dynamics. For any of the examples

mentioned above, it is obvious that the “true” \mathbf{F} would involve complicated equations. Rather than going into that kind of detail, we consider below two

rather simple choices for \mathbf{F} . That is:

$$\mathbf{F}(\mathbf{x}) = \mu \mathbf{x}, \quad (1.2)$$

and

$$\mathbf{F}(\mathbf{x}) = c \pi \sin(\mathbf{x}), \quad (1.3)$$

where μ and c are (real) constants. **These are your tasks:**

1. For (1.1–1.2), without using the computer, what can you predict about the behavior of \mathbf{x}_n , given an initial condition $\mathbf{x}_0 \neq \mathbf{0}$? (Note: of particular interest is the behavior for $n \rightarrow \infty$).
 - 1a. Qualitatively, what is different about the cases $|\mu| < 1$, $|\mu| > 1$, and $|\mu| = 1$?
 - 1b. What is the qualitative difference between the evolution with $\mu > 0$ and $\mu < 0$?
 - 1c. Check your conclusions numerically (do not hand in any plots for this, you can even use a calculator).

Note: In this case $\mathbf{x} = \mathbf{0}$ is a *fixed point* of the evolution (i.e.: $\mathbf{x}_{n+1} = \mathbf{x}_n$ if $\mathbf{x}_n = \mathbf{0}$), and your conclusions, in particular, speak to the *stability* of this fixed point (what happens if a small perturbation is applied).

2. We now turn to (1.1–1.3). For this example a MatLab script, `IterateSineMap`, is supplied. Please **read the script description at the top of the `IterateSineMap.m` file**, which describes in detail what the script does and how to use it. **The aim of this part of the problem is to explore the behavior of (1.1–1.3) as the parameter c changes.** Notice that, **if $0 \leq c \leq 1$, then for $0 \leq x_n \leq \pi$, we have $0 \leq x_{n+1} \leq \pi$.** Thus we will restrict the exploration to these ranges and ask **How does x_n behave for n large?** (`IterateSineMap` was specifically designed for this task).

The system (1.1–1.3) exhibits many behaviors. For example, x_n may approach a constant (a fixed point) as $n \rightarrow \infty$, or it may approach a *periodic cycle*, where $x_{n+p} = x_n$ (p is the period), or the $n \rightarrow \infty$ behavior may be chaotic (see remark 1.1), or it may exhibit *intermittent chaos*, where the sequence alternates (seemingly randomly) between being close to a periodic cycle, and chaotic bursts.¹ These various behaviors correspond to different values of c , with the transition from one to another occurring at *critical values* of c (of which there are many; infinitely many, in fact).

A value $c = c_c$ is called a critical value if the long term evolution for x_n changes *qualitatively* as c crosses c_c . These qualitative changes are called **bifurcations**. For example, on one side of c_c the system may be attracted to some fixed point x_* , and to a different fixed point on the other side. Or maybe the behavior switches from fixed point to periodic of some period $p > 1$, or the period changes across c_c , etc.

These are the questions you are asked to investigate:

- 2a.** What happens for $0 \leq c < c_1 = 1/\pi \approx 0.3183$, and for $c_1 < c < c_2$, where² $c_2 \approx 0.72$. Can you explain why c_1 happens at $1/\pi$? *Hint: Linearize the map for x small, and use part 1.*

Note that c_1 and c_2 are *critical values, or bifurcation points*.

- 2b.** What happens for c slightly above c_2 ? *Hint: Check $c = 0.73$ first, and then explore more carefully.*

- 2c.** In fact, there is an infinite sequence of critical points $c_1 < c_2 < c_3 < \dots$, with $\lim_{n \rightarrow \infty} c_n = c_\infty$ (where $c_\infty \approx 0.8655791$). In each “window” $c_n < c < c_{n+1}$, the limit behavior of x_n is very simple — in part 2b you should already have discovered what happens for $c_2 < c < c_3$.

Compute the first few c_n , say, c_3 and c_4 , and describe the behavior in the corresponding windows, say $c_3 < c < c_4$ and $c_4 < c < c_5$ — note that you do not need to know c_5 to ascertain the behavior in $c_4 < c < c_5$. Provide values with, at least, 4 accurate decimal digits: $c_n \approx 0.abcd$.

Approximate values are as follows: $c_3 \approx 0.805$, $c_4 \approx 0.850$, $c_5 \approx 0.863$, and $c_6 \approx 0.865$. Note that beyond c_8 4 digits are not enough to tell the values apart, so the calculations get increasingly harder.

- 2d.** From your results in 2c, **can you guess what the pattern is for $c_n < c < c_{n+1}$.**

- 2e.** Now look at what happens slightly beyond c_∞ . Specifically, **what do you see for $c = 0.866$?** Do your best here; see remark 1.1.

- 2f. What do you see for $c = 0.880876$?**

- 2g. What do you see for $c = 0.8814$?** Note: in fact, the behavior you see here happens for a whole window, $0.880877 \dots < c < 0.881464 \dots$

Optional: keep going a little beyond $c = 0.881464 \dots$, what happens?

- 2h. What do you see for $c = 0.9395$?** Note: in fact, the behavior you see here happens for a whole window, $0.937819 \dots < c < 0.940943 \dots$

Optional: keep going a little beyond $c = 0.940943 \dots$, what happens?

¹The MatLab script provided allows you to check for these behaviors.

²A more precise value is $c_2 = 0.719961682979535 \dots$. We will see how this can be computed later in the semester.

Remark 1.1 For this problem we will not define chaos (will be done later). Here simply check that the behavior is not periodic of any “reasonable” period,[†] e.g.: $1 \leq p \leq 16$ (many can be excluded by simple eye-sight). ♣

[†] Because you are doing this in a computer, there is only a finite number of values x_n can take, so the computed sequence x_n will always be periodic ... but the period can be huge, for any practical purposes “infinite”.

1.2 Answer: Computer exercises with a 1-D map

The **problem solution**, with the same item numeration used in the statement, **is after the next 3 remarks**.

Remark 1.2 Because of the issues pointed in remark 1.4, there will be no penalty if you do not get c_3 and c_4 with 4 accurate digits — in fact, you only need 3 to be able to answer the questions where they matter. The *key thing* I hope you found is the behaviors described below in the answer, albeit not with the amount of detail here. ♣

Remark 1.3 Calculating the various thresholds, “naive” approach. I will illustrate how to do this, using the MatLab script SineMapCompute, with the example of c_3 — the value at the boundary between the period 2 and period 4 attractors. Let us begin by assuming that you have already found some value of c for which a period 2 attractor happens; say: $c = 0.75$. Then start “slowly” increasing c till you loose the period 2; say, you try: $c = 0.77, 0.79, 0.81, 0.83, 0.85$. OK, now you will know that $0.83 < c_3 < 0.85$, and that $c = 0.85$ gives period 4. Now you can repeat the process with a smaller Δc , and get, say: $0.831 < c_3 < 0.834$. And so on.

Efficiency: You can make the process above more efficient by doing “bisection”. First do a few relatively large Δc , and find that, say $0.77 < c < 0.85$. Then try the middle point, 0.81 , and learn that $0.81 < c < 0.85$. Repeat. You could even program this approach, but the time that will take you to write and debug the program will likely be longer than just doing it yourself.

Important: (i) As you get closer to the critical value, you will notice that the dynamical system converges to it slower, so you need to make the parameter $n1$ (how many iterates are done before looking at the attractor). (ii) SineMapCompute includes a parameter, $n3$, that allows you to check if the answer has period $n3$. This is critical because near c_3 the period 4 solutions on $c > c_3$ cannot be distinguished “by eyesight” from a period 2. (iii) If the period check tells you that the error in satisfying, say, the period 4 condition is 10^{-14} (or some other rather small number), this does not mean “it is not period 4”. You have limited accuracy, so something that small has to be interpreted as “it may be period 4” — see remark 1.3. ♣

Remark 1.4 Limits on accuracy. You may think that, if you are working in double precision (with about 16 digits) you should be able to get the c_n with a lot of digits (if you are patient enough). But this is not true, the approach in remark 1.3 will, generally, not be able to provide you with more than a few digits, 6 at best (and it is rather easy to get confused and make mistakes) — “why” is explained in some detail in remark 1.6

Can you do better? The answer is **yes, significantly better**, but it requires using theoretical knowledge not yet introduced in the lectures. Using such knowledge, much better approaches to computing the thresholds can be developed. I used them to get my numbers in the final answers, though when I wrote the problem statement I used rough values for the thresholds obtained with the naive approach, and some were not accurate. ♣

- The solution to (1.1–1.2) can be written explicitly:
$$x_n = \mu^n x_0. \quad (1.4)$$

It follows that:

- When $|\mu| < 1$, $x_n \rightarrow 0$ as $n \rightarrow \infty$. Hence the fixed point $x = 0$ is an attractor, in particular: stable.
When $|\mu| > 1$, $|x_n| \rightarrow \infty$ as $n \rightarrow \infty$. There is no attractor, and $x = 0$ is unstable.
When $|\mu| = 1$, $|x_n| \equiv |x_0|$. There is no attractor, but $x = 0$ is “neutrally” stable (perturbations neither grow nor decay in size).
- When $\mu > 0$ all the iterates have the same sign. No “oscillations”.
When $\mu < 0$ the iterates alternate sign. Period 2 “oscillations” (damped if $|\mu| < 1$, and growing if $|\mu| > 1$).

- The **solution below was done using the supplied MatLab script** IterateSineMap.

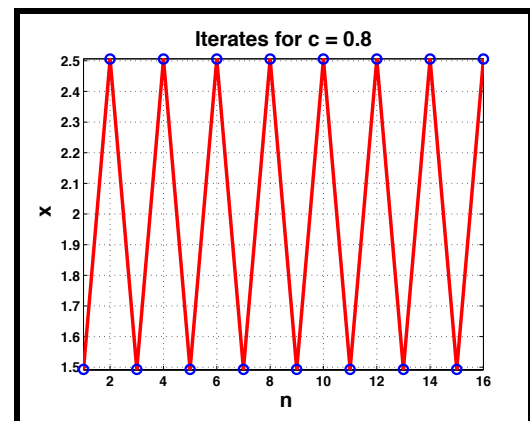
2a. For $0 \leq c < 1/\pi$, $x_n \rightarrow 0$ as $n \rightarrow \infty$.

For $1/\pi < c < c_2$ ($c_2 = 0.719961682979535\dots$), $x_n \rightarrow x_*$, where $x_* > 0$ is another fixed point of the map, $x_* = c\pi \sin(x_*)$ —as c grows, x_* grows as well. Hence $c = 1/\pi$ is a bifurcation point.

Notice that, for $0 \leq c < 1/\pi$, the equation $x = c\pi \sin(x)$ has only one solution, $x = 0$, while for $c > 1/\pi$ a second solution, $x = x_*$, appears on $0 < x < \pi$. Graphically you can see this because for $c < c_1$ the curve $y = c\pi \sin(x)$ is below $y = x$ for $0 < x \leq \pi$, while for $c > c_1$ a portion bulges above. Finally, linearizing (1.1–1.3) for x_n small yields $x_{n+1} = c\pi x_n$. This is (1.1–1.2) with $\mu = c\pi$. Thus we see that, as c crosses $1/\pi$, the origin changes stability.

Note about c_2 . Obviously the value given above was not computed using IterateSineMap. It was done as follows: c_2 is the value of c at which x_* ceases to be stable. From item 1 we know that this means that the derivative of the map there must have modulus 1, and it is easy to see graphically that it must be -1 (not 1). Thus we get the two equations: $x_* = c_2 \sin(x_*)$ and $-1 = c_2 \cos(x_*)$. Using the first to eliminate c_2 then gives $0 = 1 + x_* \cos(x_*)/\sin(x_*)$, which can be solved to find x_* for $c = c_2$. Then $c_2 = x_*/\sin(x_*)$. A more involved version of this process can be used to find the other critical values c_n . ♣

2b. Above c_2 there is a “window” $c_2 < c < c_3$, where $x_n, n \rightarrow \infty$, approaches a period 2 state. Thus $c = c_2$ is a bifurcation point. The figure on the right shows the period 2 state for $c = 0.8$, after 2×10^6 iterations from a random x_0 . The “deviations from periodic calculation” (not shown) confirms[†] the period 2. For c slightly above c_2 the amplitude of the oscillations is small, with the amplitude growing with c — later on we will show that they grow like $\sqrt{c - c_2}$ when $c - c_2$ is small. Note that, depending on x_0 , the “phase” of the asymptotic oscillations can change by 1.

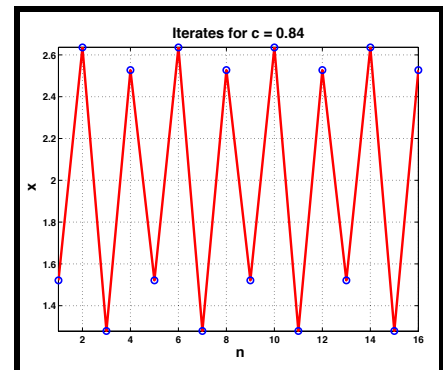


[†] The “deviations from periodic calculation” plots the difference $x_{n+p} - x_n$ ($p = 2$ in this case). Since the x_n are $O(1)$, and each is represented by 16 digits only,³ if the difference $x_{n+1} - x_n$ is smaller than 10^{-16} , then it is zero. Yet IterateSineMap plots $x_{n+p} - x_n$ using a vertical scale of $\pm 10^{-100}$. Do not be fooled by this scale, it is a default I set-up because a plot needs some vertical scale.

2c. A careful⁴ calculation yields $c_3 \approx 0.8332663537$, $c_4 \approx 0.8586090599$, $c_5 \approx 0.8640841737$, \dots , with $c_\infty \approx 0.8655792688$.

In each “window” $c_n < c < c_{n+1}$, the limit behavior of x_n is a periodic state with period 2^{n-1} . For the obvious reason, this is called a period doubling cascade. The figure on the right shows the period 4 state for $c = 0.84$, after 2×10^6 iterations from a random x_0 . The “deviations from periodic calculation” (not shown) confirms the period 4. Note: depending on x_0 , the asymptotic oscillations’ “phase” may change by 1, 2, or 3.

At the lower end of each window the periodic state is very close to the prior window periodic state. This is illustrated in figure 1.1 — again, the “error” behaves like $\sqrt{c - c_n}$.



Note: In the problem statement I gave approximate values for some c_n 's. The intention was to give you a rough idea of what the values are, to limit your search to a small range of c . Unfortunately I wrote these values with 3 digits

³Technically, I should be doing this analysis using binary notation, but for the point made here this is not important.

⁴These numbers were not obtained using IterateSineMap, but using a process similar to the one used for c_2 , explained in item 2a. With IterateSineMap it is very hard to get more than a few digits — see remarks 1.3, 1.4, and 1.6.

(not sure what I was thinking!), which probably gave the wrong impression as to how accurate the values were. What I should have stated is: $0.82 < c_3 < c_4 < c_5 < 0.865$. Also, you were asked to compute c_3 and c_4 only. ♣

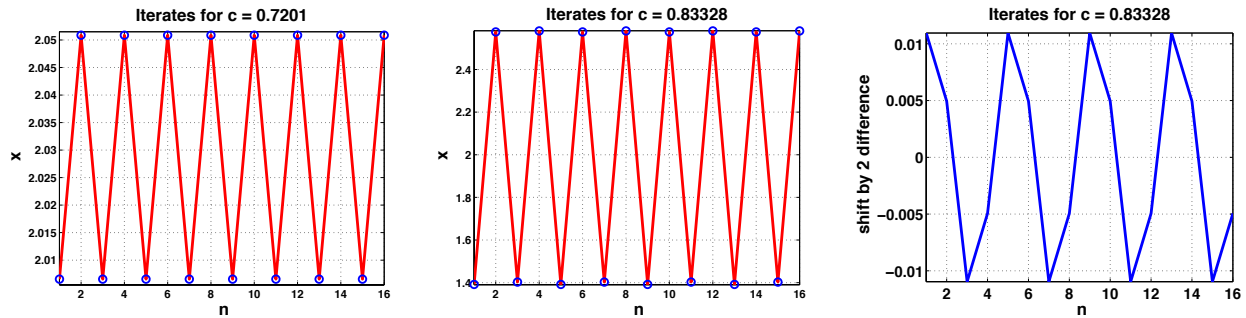


Figure 1.1: **Typical behavior near a period doubling.** The left panel shows the limit behavior for $c_2 < c = 0.721$ (very close to the left end of the period 2 window) after 2×10^6 iterations of the map. The middle panel is the limit behavior (also after 2×10^6 iterations) for $c_3 < c = 0.83328$ (very close to the left end of the period 4 window). Notice that the sequence in the middle panel seems to be period 2, but it is not, as the right panel shows — though the deviations from period 2 are small. The deviations from period 2 (resp. period 4) calculations, not shown, for the left panel (resp. middle panel) confirm the periods 2 and 4.

2d. The answer to the question in item 2d is included in the answer to item 2c.

2e. Slightly beyond c_∞ we find chaotic attractors, as illustrated by the left panel in figure 1.2. Following remark 1.1, I tested this attractor for all the periods between 2 and 17, and many more (which it failed) — but see remark 1.5. However, **there is a second test for chaos that is useful to perform.** Namely:

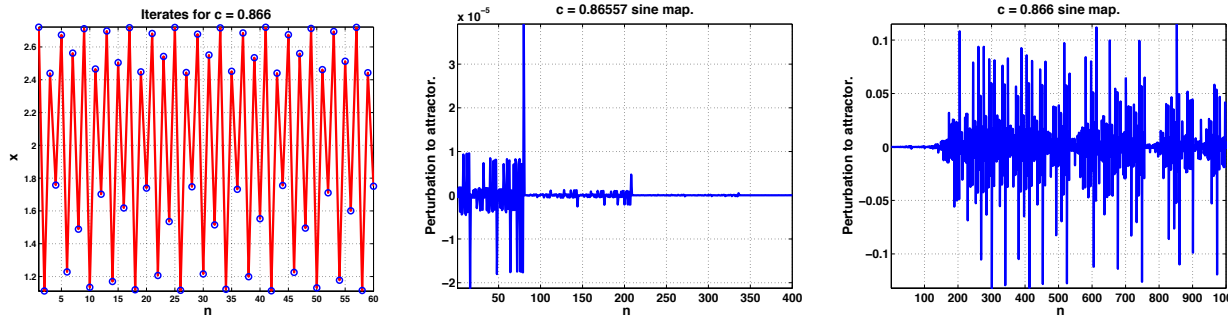


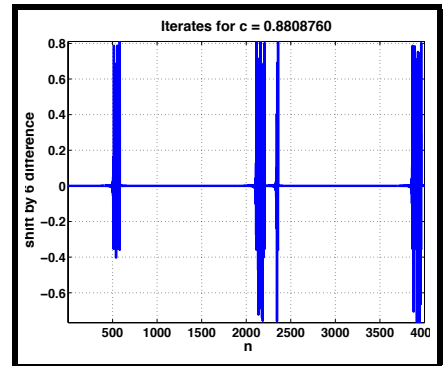
Figure 1.2: **Chaos beyond c_∞ .** The left panel shows the limit behavior for $c = 0.866$, slightly above c_∞ , after 2×10^6 iterations of the map. The middle panel shows the behavior of a 10^{-5} perturbation applied to a period 128 attractor ($c = 0.86557$). The right panel shows the behavior of a 10^{-5} perturbation applied to the attractor on the left panel.

test the sensitivity of the attractor to small perturbations. To do this we first iterate the map for $1 \leq n \leq n_1 + n_3$, where n_1 is large enough that transients decay and x_n is in the attractor for $n > n_1$ (e.g.: $n_1 = 10^6$). At n_1 we start a second iteration, with a perturbed initial data $y_{n_1} = x_{n_1} + \delta$, where δ is small (e.g.: $\delta = 10^{-5}$). Then [†] we plot the difference $y_n - x_n$ for $n_1 < n \leq n_1 + n_3$. The results of this test are shown in the middle and right panel of figure 1.2. When the test is used on a **periodic attractor**, the **perturbation** never grows beyond a few times the initial size, and it eventually **decays** (in the figure we used $c = 0.86557$, which gives a period 128 attractor). On the other hand, for a **chaotic attractor**, the **perturbation grows**, at first exponentially, till it reaches a size orders of magnitude larger than δ , and then it does not decay (although its size may fluctuate wildly). For $c = 0.866$ we found a magnification of δ by a factor of 10^4 . This **sensitivity to perturbations is a very important property of chaos, and the basis for the “butterfly effect”**.

† This can be done with the script `IterateSineMapAttSensi.m`, handed out with this answer. This script has a parameter n_2 , used to plot the attractor, x_n , for $n_1 < n \leq n_1 + n_2$. Do not take n_2 large, or you will not see much.

Remark 1.5 Errors in period testing for $c = 0.866$. Some extremely long periods gave small errors in the period test. For example, $p = 5040 = 2^4 3^2 5^7$ lead to $\max_n |x_{n+p} - x_n| = O(10^{-12})$. I do not know the reason for this; I am not even sure it is not a “numerical effect” — i.e.: I do not trust a calculation with such a long period. ♣

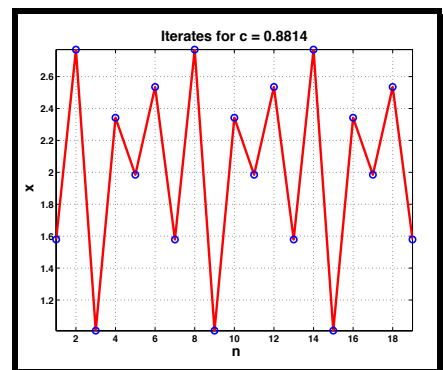
- 2f. The picture on the right shows the behavior of “the error” for the “period 6 check” calculation of the attractor for $c = 0.880876$. What we see is long intervals, $O(1000)$ in n , where the error is small, interrupted by short bursts where the error is much larger. In terms of the attractor (plot not shown), this corresponds to long intervals where the attractor almost looks like a period 6 sequence, with short bursts where it looks “random”† (no pattern). This behavior is called “**intermittent chaos**”, because it resembles the phenomena of “intermittent turbulence” observed in some fluid dynamics experiments.



† We use the word random loosely here. The behavior is deterministic!

In fact, for $c > c_{6,0}$ (where $0.880877 < c_{6,0} < 0.880880$) there is a “window” where the attractor is a period 6 sequence — see the answer to item 2g. Below this window, $0 < c_{6,0} - c$ small, period 6 intermittent chaos happens — you can check this with `IterateSineMap` for $0.880870 \leq c \leq 0.880876$. Here, as c decreases, the effect gets less visible: the near-period 6 intervals get shorter, and the period 6 error in these regions gets larger. Eventually you just see uninterrupted chaotic behavior (there is no sharp boundary, though). The explanation to this behavior is analogous to the “critical slowdown” near critical points. For $0 < c_{6,0} - c$ small, if x_n gets close to a member of the (stable) period 6 cycle that appears at $c = c_{6,0}$, the dynamics gets trapped there, and evolves slowly away from being near the period 6 cycle. Once it gets away, it “resumes” its chaotic behavior, till at some later time, x_n again gets close to the cycle, and so on. Note that the time intervals between chaotic bursts are also chaotic in length.

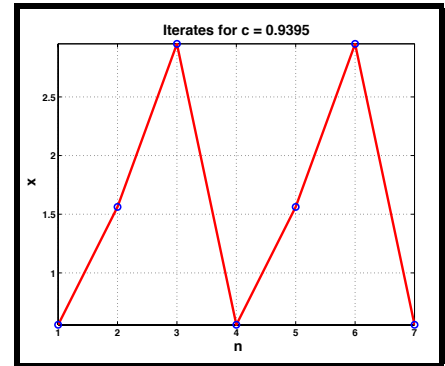
- 2g. The picture on the right shows the $c = 0.8814$ attractor. It is a period 6 sequence — the “deviations from periodic calculation” (not shown) confirms this period. This behavior occurs for a whole window, $c_{6,0} < c < c_{6,1}$, where $0.8818 < c_{6,1} < 0.8819$.† Beyond this window, another period doubling cascade occurs, with $c_{6,0} < c_{6,1} < \dots < c_{6,n} < c_{6,\infty}$, where the attractor has period 6×2^n for $c_{6,n} < c < c_{6,n+1}$, and chaos is found beyond $c_{6,\infty}$.



† The problem statement has the overly precise $c_{6,1} = 0.881464\dots$, which happens to have an incorrect 4-th digit!

2h. The picture on the right shows the $c = 0.9395$ attractor; a period 3 sequence — the “deviations from periodic calculation” (not shown) confirms this period. This behavior persists for the window $c_{3,0} < c < c_{3,1}$, where $0.937 < c_{3,0} < 0.938$ and $0.942 < c_{3,1} < 0.943$. Beyond this, **another period doubling cascade occurs**, with $c_{3,0} < c_{3,1} < \dots < c_{3,n} < c_{3,\infty}$, where the attractor has period 3×2^n for $c_{3,n} < c < c_{3,n+1}$, and chaos is found beyond $c_{3,\infty}$.

† The problem statement has the overly precise $c_{3,1} = 0.940943\dots$, which happens to have an incorrect 3-rd digit!



Note that, **for $0 < c_{3,0} - c$ small, period 3 intermittent chaos happens** — the discussion in item **2f** has an exact analog here.

Remark 1.6 Errors in computing thresholds. The first thing to point out is that, as c crosses a c_n going up, the “perturbation” to the prior attractor (e.g: failure to satisfy the period 2 condition) has size $O(\sqrt{c - c_n})$ when $c - c_n$ is small. Now, if you have only 16 significant digits,⁵ you can only tell that you “have crossed c_n ” only once $\sqrt{c - c_n} \approx 10^{-16}$; you will not be able to tell the effect of smaller changes because they are beyond the resolution you have. This gives you the largest penalty, at best you can hope for 8 digits in c_n .

But this is not the end of it. You also have to consider the fact that in fixed point calculations, the last digit jumps around. The behavior is not “random” (i.e.: it is deterministic⁶) but its effect is quite similar to a random perturbation. Because the approach to the attractor is very slow close to c_n , these perturbations are able to derail it, creating a fuzzy region near c_n , where it is not clear which side of the critical number you are. This makes things worse, and 2 or more digits may be lost. As an **example** of this, consider what happens with the period 3 check for c near **0.9395** (see item **2h**). Running the “deviations from periodic calculation” for several values of c , in the range $c = 0.9391$ to $c = 0.9401$, gave results where the error oscillated between 0 and $O(10^{-15})$ for changes in the 4-th (!!) decimal digit in c — note that, while this sort of stuff happens with any machine, the details depend on the machine (and the algorithm used to compute the sine). ♣

THE END.

⁵Technically, I should be doing this analysis using binary notation, but at the rough level here this is not important.

⁶If you run the calculation twice, in the same computer, you will get the same answer.