

## Chaos (A brief look)

### DUE DATES FOR EXERCISES

**P11, P12:** Wednesday, November 28, 2007; 5 pm.

### *Comment:* **STOP!**

Nearly everyone in this class has met the goal of developing a familiarity with programming, which I'm quite happy about, and has done well on the programming assignments. (Wondering if you're in this category? If you're not doing well, you've recently received a cautionary email from me.) This is the last programming assignment, and though it's not terribly difficult, it's certainly not simple. For many of you, **your time will be better spent** working on the coupled oscillator project, working on homework, and studying the midterm and older assignments in preparation for the final. I encourage you to read this assignment, since you might find it interesting, but please **don't feel compelled to do the exercises**. Mull it over during the Winter break. Really. If you want to hand it in, note the due date above.

## 1 Chaos (Motivation)

Throughout our course we've been able to analyze various physical systems and derive fairly simple equations of motion that describe them. These include, for example, the motion of a pendulum, swinging back and forth. If we start with some initial conditions, we know the motion of the pendulum for all subsequent times. If we very slightly change the initial conditions, the motion slightly changes, since the amplitude and phase of our solution change a little bit. Right?

For centuries this was assumed to be the case, until it was discovered that equations of motion only slightly more complicated than those of a simple harmonic oscillator are really not so simple – in many cases they can exhibit behavior that depends extremely sensitively on initial conditions, and that moreover seems to have no discernible periodicity. This is termed **chaos**, or **deterministic chaos**. (“Deterministic” means that the motion at some time is specified by what preceded it – i.e. it's not random.) A few people about 100 years ago, most notably Henri Poincaré, began to realize the existence and consequences of chaos. With the advent of computers in the 1960's and 1970's, though, our understanding of chaos bloomed. Now we realize that there are many “simple” systems that, though deterministic, are completely unpredictable, impacting not only fields as diverse as physics, weather prediction, and economics, but our basic outlook on nature.

There are many books on chaos and “non-linear dynamics.” One excellent place to start is a non-technical (i.e. “popular”) book, James Gleick's *Chaos: Making a New Science* (1988).

In this assignment, we'll briefly look at a few manifestations of chaos.

## 1 Population dynamics and the logistic map

The mathematical modeling of biological populations – animals in the wild, bacteria infecting a populace, etc. – is an important tool for understanding ecology. Surprisingly, even very simple models of population dynamics can lead to very surprising conclusions:

**151.** Lots of things are mysteries. But that doesn't mean there isn't an answer to them. It's just that scientists haven't found the answer yet.

For example, some people believe in the ghosts of people who have come back from the dead. And Uncle Terry said that he saw a ghost in a shoe shop in a shopping center in Northampton because he was going down into the basement when he saw someone dressed in gray walk across the bottom of the stairs. But when he got to the bottom of the stairs the basement was empty and there were no doors.

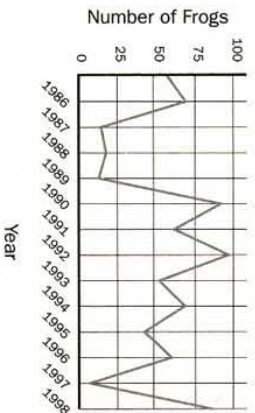
When he told the lady on the till upstairs, they said it was called Tuck and he was a ghost of a Franciscan friar who used to live in the monastery which was on the same site hundreds of years ago, which was why the shopping center was called **Greyfriars Shopping Center**, and they were used to him and not frightened at all.

Eventually scientists will discover something that explains ghosts, just like they discovered electricity, which explained lightning, and it might be something about people's brains, or something about the earth's magnetic field, or it might be some new force altogether. And then ghosts won't be mysteries. They will be like electricity and rainbows and nonstick frying pans.

But sometimes a mystery isn't a mystery. And this is an example of a mystery which isn't a mystery.

We have a pond at the school, with frogs in it, which are there so we can learn how to treat animals with kindness and respect, because some of the children at school are horrible to animals and think it's funny to crush worms or throw stones at cats. And some years there are lots of frogs in the pond, and

some years there are very few. And if you drew a graph of how many frogs there were in the pond, it would look like this (but this graph is what's called *hypothetical*, which means that the numbers aren't the real numbers, it is just an *illustration*)



And if you looked at the graph you might think that there was a really cold winter in 1987 and 1988 and 1989 and 1997, or that there was a heron which came and ate lots of the frogs (sometimes there is a heron who comes and tries to eat the frogs, but there is chicken wire over the pond to stop it).

But sometimes it has nothing to do with cold winters or cats or herons. Sometimes it is just maths.

Here is a formula for a population of animals

$$N_{\text{new}} = \lambda (N_{\text{old}}) (1 - N_{\text{old}})$$

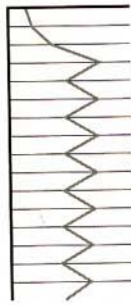
And in this formula  $N$  stands for the population density. When  $N = 1$  the population is the biggest it can get. And when  $N = 0$  the population is extinct.  $N_{\text{new}}$  is the population in one year, and  $N_{\text{old}}$  is the population in the year before. And  $\lambda$  is what is called a constant.

When  $\lambda$  is less than 1, the population gets smaller and smaller and goes extinct. And when  $\lambda$  is between 1 and 3, the

population gets bigger and then it stays stable like this (and these graphs are hypothetical, too)



And when  $\lambda$  is between 3 and 3.57 the population goes in cycles like this



But when  $\lambda$  is greater than 3.57 the population becomes chaotic like in the first graph.

This was discovered by Robert May and George Oster and Jim Yorke. And it means that sometimes things are so complicated that it is impossible to predict what they are going to do next, but they are only obeying really simple rules.

And it means that sometimes a whole population of frogs, or worms, or people, can die for no reason whatsoever, just because that is the way the numbers work.

from “The Curious Incident of the Dog in the Night-time” by Mark Haddon (Vintage Books, New York, 2003), p. 100-102.

(From *The Curious Incident of the Dog in the Night-Time*, a very good novel in which an autistic boy investigates the murder of a neighbor’s dog).

Let’s examine the phenomenon described above. Consider a simple population model, in which a population can grow, but due to environmental constraints (e.g. a finite supply of land or food) it has a finite “capacity” it cannot exceed. Define  $x_n$  as the population in generation  $n$  as a fraction of the maximum population. (So that  $x$  must be between 0 and 1.) We assume that:

- (1) The population of generation  $n+1$  is proportional to  $x_n$  (which makes sense), and
- (2) The population of generation  $n+1$  is proportional to  $1-x_n$ , i.e. the population grows faster the farther it is from the maximum capacity.

Combining these:  $x_{n+1} = r x_n (1-x_n)$ , where  $r$  is some parameter. (This is known as the “logistic map.”)

**(Exercise P11, 6 pts.)**

(a, 1 pt.) Write a simple MATLAB program that implements the logistic map for  $n=1$  to 40 (i.e. 40 generations) and  $r=2.7$ . Run it three times for three different values of  $x_1$  -- use the last three digits of your UO ID number, each divided by 10. (I.e. if your ID ends in ...145, try 0.1, 0.4, and 0.5). (If any digit is 0, pretend it’s 1.) Plot all of the resulting  $x(n)$  graphs together. You should find that they all rapidly converge to the same value. **Hand in your plot.**

- (b, 1 pt.) Inspired by the convergence of (a), determine by applying simple algebra to the logistic map equation (i.e. without any programming) the value to which  $x$  converges, as a function of  $r$ . (*Hint:* You should find that for  $r=2.7$ ,  $x_\infty=0.6296\dots$ )
- (c, 1 pt.) Run your program of part (a), but for  $r = 3.75$  and  $x_1 = 0.1 d$ ,  $0.1001 d$ ,  $0.0999 d$ , where  $d$  is the last digit of your UO ID number (or 1 if that's 0). Hand in your plot of each of these three  $x(n)$ .
- (d, 3 pts.) Hopefully, your plot of (c) looks different than the plot of (a) – not only is there no convergence, but very similar values of  $x_1$  lead to very different  $x_n$  later on. Perhaps the logistic map is not as simple as it seems! Suppose we ran the logistic map for 1000 generations, from some initial value  $x_1 = 0.1 d$ , as above. Suppose we considered the last 300 values of  $x$  (i.e. generations 701 to 1000). For the setup of part (a) we'd think that these all have the same value, while from part (c) we'd suspect that they won't, and that  $x$  is meandering over a wide range. **Write a program to calculate and plot**, as a function of  $r$  from  $r=2.7$  to  $r=4.0$  in steps of 0.005, all the last 300 values of  $x$  that occur at that  $r$ . Here's a piece of what your answer should look like (i.e. a “zoomed in” view of part of the plot.) (I recommend drawing points rather than e.g. circles, since there are lots of data to plot: `plot(... 'k.')`);

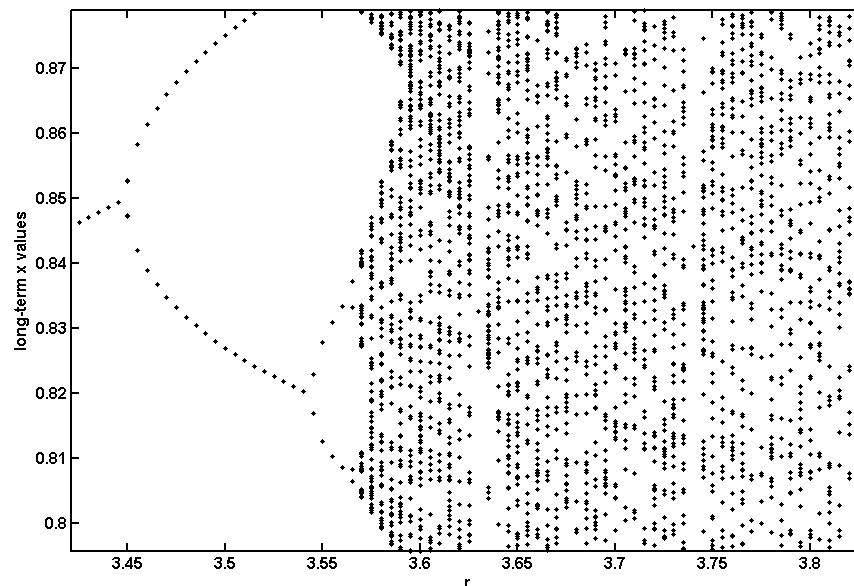


Figure 1.  $x(701:1000)$  versus  $r$  for the logistic map.

This is known as a “bifurcation diagram.” You should see that some values of  $r$  lead to a single “stable” population. Others lead to two – the population oscillates between these. Others lead to three, and so on. Others lead to an infinite number – the population is “chaotic,” a term we have not really defined, but whose characteristics are becoming clearer. Interestingly, there are “islands of stability” – ranges of  $r$  that are surrounded by chaotic regions, but are not themselves chaotic. Can you find one?

**Hand in your plot and your program.**

*Suggestion:* You'll probably find it easiest to loop through all the  $r$  values in a “for” loop. For each  $r$ , determine the  $x$  values you want to plot, and plot them – display all of these values in the same figure, being sure to avoid overwriting the previous plot by enabling “hold on.” You'll need to plot many  $x$  values for a single  $r$  value; one way is to make an array of “duplicate”  $r$  values. For example: `“r * ones(1, 4)”` creates a 1x4 array that is  $[r \ r \ r \ r]$ .

## 2 Chaos in driven pendulums

In class we examined the motion of a forced, damped harmonic oscillator, with the equation of motion  $\ddot{x} = -\omega_0^2 x - \gamma \dot{x} + \bar{F} \sin(\omega t)$ , where  $\omega$  is the driving frequency,  $\omega_0$  is the natural frequency,  $\gamma$  is the driving factor, and  $\bar{F}$  is the amplitude of the driving force in some natural units (e.g. force divided by the mass of the oscillator). This describes the forcing of a mass on a spring, or a LCR circuit, or a swaying pendulum. Surprisingly, chaos arises when we make very slight modifications to this system, for example, getting rid of the small angle approximation for the pendulum, for which:

$$\ddot{\theta} = -\omega_0^2 \sin(\theta) - \gamma \dot{\theta} + \bar{F} \sin(\omega t)$$

where  $\theta$  indicates the angle and  $\bar{F} = \frac{F_0}{ml}$ . (If this isn't clear, derive it – recall that the version without forcing came up in an earlier programming assignment.) How can we examine this system? By using what we've learned about calculating numerical solutions to differential equations!

**(Exercise P12, 11 pts.)** Apply the Runge-Kutta algorithm to numerically solve the above differential equation of motion for the driven pendulum. Use the parameters  $\omega_0 = 1$  (rad/sec),  $\gamma = 0.5$  (N s / m kg), and  $\omega = 0.7\omega_0$ . Use whatever initial conditions you like, e.g.  $\theta(0) = 0$  and  $\dot{\theta}(0) = -1.0$  radians/sec. Calculate  $\theta(t)$  and  $\dot{\theta}(t)$  for a time interval corresponding to a few thousand  $T$ , where  $T$  is the period of the driving. Run your program for  $\bar{F} = 0.4, 0.6, 0.8, 1.0, \text{ and } 1.2$  (N kg / m). Hand in your program and various plots:

- (a, 4 pts.) Plot the angular velocity  $\dot{\theta}$  for a time interval corresponding to  $20 T$ , starting at some time  $t$  long after the transient response of the oscillator has died away. You should have one plot per  $\bar{F}$  value.
- (b, 5 pts.) Plot together  $\dot{\theta}(t)$  as above for *two* situations in which the initial conditions differ only slightly (e.g. by 0.1%). Do the same for the “usual” pendulum, in which we make the small  $\theta$  approximation. (You'll have to modify your program, but in an easy way.) Turn in all your plots, and briefly comment on them.
- (c, 2 pts.) Plot the phase space diagram, i.e.  $\dot{\theta}$  vs.  $\theta$ , starting at a time after which transient behavior has decayed. (Optional: In case you'd like to investigate a clearer way to plot complicated phase space diagrams, look up “Poincaré sections,” and hand these in instead.) Turn in your plots, and briefly comment on them.

### Comments / hints:

- If you don't see any difference for the various  $\bar{F}$ , you've probably done something wrong. However, try “playing with” different  $\bar{F}$  values anyway, and see if you can find non-chaotic and chaotic regimes.
- You might need to “speed up” your programs. One simple thing to do is avoid having arrays that grow in size, since the memory allocated to them constantly has to be re-sized. If you're working with an array “x” that is being constructed by a loop and that eventually will be a matrix with N elements, then first, before the loop, define “x = zeros(1, N)”, for example, which makes x a 1xN array of zeros.
- Recall especially for (c) that  $\theta$  is an angle, so only its value “mod  $2\pi$ ” is meaningful. MATLAB's “mod” command may be useful.
- If you want to put multiple plots in one figure, look up the “subplot” command.