

Bifurcation Diagrams

1 Introduction

Today we are going to explore the concept of bifurcations and how Matlab can help us visualize these changes. We will begin with some basic definitions, then examine some plots, and finally create a *bifurcation diagram*

2 What is a bifurcation?

A *bifurcation* is a fundamental change in the nature of a solution. Ok, but what does that actually mean? Let's examine the logistic equation as an example. Recall that the logistic equation is given by $x' = \tilde{r}x(1 - x)$, where $x(t) > 1$ is the population at time t and \tilde{r} is some growth factor. To make our life simpler, we will actually use the logistic map, $x_{n+1} = rx_n(1 - x_n)$, $0 < x(t) < 1$ which is simply a discrete version of the logistic equation. The type of solution we have depends entirely on r . For example, when $r = 1.5$, we will find that the population stabilizes at a value of $\frac{1}{3}$. However, when $r > 3$, we will find that the solution becomes periodic, implying that the population oscillates between two values. This change from a single stable fixed point to a periodic solution is an example of a bifurcation. Similarly, when $r > 1 + \sqrt{6}$ there is another split. Now, the population will oscillate between 4 values. This is another bifurcation!

3 What does this look like?

Lets begin by writing an m-file for the logistic map, called *logistic.m*. This m-file should take a value of r , an initial condition $0 < x_0 < 1$ and number of steps N as input, and output a vector x containing all the population at all N steps.

```
function[x]=logistic(r,x0,N)
x(1)=x0;
  for i=1:N
    x(i+1)=r*x(i)*(1-x(i));
  end
end
```

First, try running the code with $r = 1.5$, $N = 30$, and try two different initial conditions (remember they need to be between 0 and 1!). You should see that the long-term behavior of the system does not depend on the initial conditions. This implies the system is *integrable* or

non-chaotic. Now, try running this code with $N = 50$ for $r = 1.5$ and $r = 3.2$. Plot both results together. What do you notice? Now try $r = 3.5$.

4 What happens next?

We just saw a fixed point turn into a period-2 oscillation turn into a period-4 oscillation. You may also have noticed that the bifurcation came faster the second time. So what happens when r is even larger? Try plotting with $r = 3.6$ and $N = 100$. Do you see any periodicity? The answer is more complex than yes or no. At roughly $r = 3.57$, the logistic map becomes *chaotic*. Chaos is difficult to define, but it generally refers to dynamical systems that are highly sensitive to initial conditions. Overlay several different plots with $r = 3.6$, $N = 100$, but change the initial condition each time. What do you notice?

5 Following the bifurcations

When studying dynamical systems, we often want to know when the system undergoes a bifurcation. A *bifurcation diagram* is a plot that shows the value of the changing parameter, r in our case, on one axis and the solution to the system on the other axis. Our next goal is to construct a bifurcation diagram for the logistic map

6 Building the Bifurcation Diagram

We want to build a bifurcation diagram that shows the solutions of the logistic map for $2.8 \leq r \leq 4$. Let's take a minute and think about how we are going to do this. We can set up a *for* loop that cycles through our r values, finding a solution vector x . Since these bifurcations happen quickly, let's use a small step size, say 0.0005. We can then plot each of these solution vectors within the *for* loop. Let's try to set this up

```
function bdiag(N)
for r=2.8:.0005:4
    x=logistic(r,x0,N)
    plot(r,x)
    hold on
end
end
```

Try running the code. It will take a little while to run-be patient. What do you notice? This bifurcation diagram has a lot of really interesting properties. For example, it displays *fractal* behavior. If you zoom in on either of the bifurcations near $r = 3.4$, the entire bifurcation diagram is recreated on a smaller scale. You can also clearly see the regions of chaos. We can also see that after an initial period of chaos, there is another region of stability at about $r = 3.85$, before another region of chaos.

This bifurcation diagram is satisfactory, but we can do a lot better. You can see that we have a lot of multiple lines where we would only expect one. For example, prior to $r = 3$ we should only

have one solution. We get multiple lines because of the *transient*. Essentially, the map takes a few steps to reach equilibrium. In fact, when r is near a bifurcation point, it takes the map a long time to reach equilibrium. We can get a much better map if we ignore the initial points from the map. Modify your code so that the first M points are not plotted (M should be another user input). Adjust M and N until you get a clear picture. You might be surprised how many points you need to throw away!

7 Bifurcations in the phase plane

Lets examine what a bifurcation looks like in the phase plane. For this investigation, we will use the Lorenz system, given by

$$\begin{aligned}x' &= 16(y - x) \\y' &= rx - y - xz \\z' &= xy - 4z\end{aligned}$$

Set up an m-file, called *lorenz.m* that takes as input t , a vector h containing the three initial conditions, and a value for r .

We will now generate several plots using *ode45*. Remember, we can use *ode45* to change the value of our parameter r using the following syntax

```
[T Y]=ode45(@lorenz, [to tf],[x0 y0 z0], [], r)
```

To begin, use the initial condition $(-13, -12, 52)$ with $r = 5$. Use a time range of $0 \leq t \leq 40$. Rather than plotting in all three dimensions, which is both time consuming and confusing, we will only examine two of the dimensions. Examine the phase space by plotting x values versus the z values. What type of solution do you have? Now increase r to 15, keeping everything else the same. What do you notice? Increase r again to 25. Finally, look at $r = 35$. A bifurcation has caused the solution to change from a stable fixed point to a chaotic attractor. This is the famous Lorenz butterfly-the iconic image of chaos theory. When chaotic, this system is highly dependent on initial conditions. Try plotting the Lorenz Butterfly using $0 \leq t \leq 3$ with two different initial conditions. Overlay the two plots. The two plots might look similar, but they are actually quite different. The same shape is generated, however the butterfly is traced out in completely different ways each time, no matter how close together you start the initial conditions!