MATH 934
TOPICS IN DIFFERENTIAL EQUATIONS
~VISIONS OF CHAOS~

INSTRUCTOR: DR. ADAM LARIOS

"Chaos is the score upon which reality is written."
-Henry Miller

"Chaos is a friend of mine."
-Bob Dylan

Note. This is a little thing for you to try out after you finish the Matlab introduction worksheet. It should go quickly, but it is fun to try out.

Discrete Dynamical Systems

Differential equations are "continuous" systems, but there are also "discrete" systems, which are called "discrete dynamical systems." Consider the familiar exponential growth population model with growth rate r. If we let P_n denote the population at the n^{th} time step, then P_{n+1}, the population at the $(n+1)^{\text{st}}$ time step is given by

$$P_{n+1} = rP_n$$

Warm up

Suppose $r = 0.17$, and the population starts at the "seed value" given by $P_0 = 3$. In Matlab, compute P_{10} as follows.

```
P = 3;
for n = 1:10
  P = 0.17*P;
end
P
```

Of course, this will not store any of the values, so if we want to store them (so we can use them to plot, for example), we can do this:

```
P(1) = 3;
for n = 1:10
  P(n+1) = 0.17*P(n);
end
plot(P);
```

(If we only pass one vector `P` to `plot`, then Matlab uses just the usual counting numbers for the x-axis, `[1 2 3 ... length(P)]`). Notice that, since Matlab can't start the index at 0, we have to start it at 1. Also, it is better to preallocate `P` by setting `P=zeros(1,10)` before the loop. This makes Matlab run faster, since it sets aside the space it needs beforehand.

Recall the logistic equation mentioned in class:

$$\frac{dP}{dt} = r \cdot P \cdot \left(1 - \frac{P}{K}\right)$$

where r is the effective growth rate, and K is the carrying capacity. We can consider a discrete version of this equation (setting $K = 1$ for simplicity), namely

$$P_{n+1} = r \cdot P_n \cdot (1 - P_n)$$

This seemingly simple dynamical system exhibits chaotic behavior, in the sense that tiny changes in the parameters can lead to very different long-term outcomes. This means that, if we want to predict the behavior, even if our measurements of these parameters is very precise, but not perfect, the long-term behavior is essentially unpredictable. We explore this below.

Things to try

(1) To get started, choose the seed value $P_0 = 0.5$, and set $r = 1.61$, and repeat this process, say, 250 times to find P_{250}. Remember to only output P_{250}, since you don't want a mess showing up. (There is nothing special about the 250 here, we just need a large number.)

(2) Next, let's try varying r. Try the above exercise plugging in $r = 2.6$, $r = 2.61$, $r = 2.611$, $r = 2.6111$, $r = 2.61111$, $r = 2.611111$, outputting only P_{250} each time. Pay attention to (and record) the values as you go. We are only slightly varying r, and the output results are not very surprising.

(3) Try the above exercise again, but this time with $r = 3.6$, $r = 3.61$, $r = 3.611$, $r = 3.6111$, $r = 3.61111$, $r = 3.611111$. What the heck just happened?

(4) OK, that was weird. Let's try to get a better picture of things by looping over a wide range of r values. Calculate P_{250} for 1000 different r values, ranging between $3 \leq r \leq 4$ (`linspace` would be a good tool to use here), and save them as you go in a big vector. Plot your r values against your P_{250} values. What do you see? How are the points changing as r increases?

(5) Try changing your seed value $P_0 = 0.5$ a little. Do you see different behavior? If not, try $P_0 = 0.7$ and $P_0 = 0.9$. What do you notice?

(6) OK, it's time to sort everything out. Let's make a big loop that runs over everything, looping from seed P_0 from 0.1 to 0.9 using, about 1000 values or so. Put this all on the same plot by declaring "`hold on`" somewhere near the top of your document. If you want to watch it draw each plot, put `pause(0.1)` in your outer loop. **Don't forget to go full screen and zoom in!**

Matlab automatically connects the dots between plotted points. Try plotting with points, to see things easier, like this:

```
plot(r_vals,P_vals,'.');
```

What you just built is called "The orbit diagram for the logistic family." It illustrates that the long-term behavior of this system is independent of initial conditions, and chaotically dependent on the parameters.