Introduction to Complexity

Homework 2

Graded Part: This part you will submit on the course webpage, by doing the exercises, and then clicking on the “Homework 2” link in Unit 2.8, filling out the answers, and then clicking on the “Submit” button at the bottom of the page.

1. Download SimpleLogisticMap2.nlogo from the Course Materials page. Set R = 3.7, x0 = .2000000 (by right-clicking on the slider, selecting “edit”, and filling in the Value box), and x0’ = .2000001. Note that x0 has 6 zeros following the .2, and x0' has five zeros and a 1 following the .2. Click on “setup”, and then “go”. Continue clicking on “go” until the red and blue dot are completely separate (i.e., not overlapping at all). How many ticks does this take? Select the number below that is closest.

   A. 15

   B. 20

   C. 35

   D. 45

   E. 55

2. Repeat the steps in Question 2, but with R = 3.8. How many ticks does it take until the red and blue dot are completely separate (not overlapping)? Select the number below that is the closest.

   A. 15

   B. 26

   C. 40

   D. 47

   E. 55

3. Repeat the steps in Question 3, but with R = 3.9. How many ticks does it take until the red and blue dot are completely separate (not overlapping)? Select the number below that is the closest.

A. 15

B. 17

C. 23

D. 41

E. 45

4. Finally, repeat the steps in Question 4, but with R = 4.0. How many ticks does it take until the red and blue dot are completely separate (not overlapping)? Select the number below that is the closest.

A. 10

B. 11

C. 18

D. 28

E. 41

5. Using the results of the questions above, answer the following: As the growth rate *R* is increased, the Logistic Map's sensitivity to initial conditions (i.e., the rate at which nearby initial conditions diverge under the map) tends to

A. increase

B. decrease

C. stay the same

Ungraded Part: This part is **optional** and ungraded; you won't turn it in. It's for people who want to go a bit more deeply into the ideas covered and get more practice with NetLogo. You can choose among the Beginner, Intermediate, or Advanced options, depending on your level of familiarity and comfort with NetLogo and with the ideas presented in the lectures. Feel free to discuss this part of the homework with anyone, or to ask questions about it on the Class Forum.

Beginner Options:

1. Using a calculator (not the NetLogo Logistic map model), iterate the Logistic map with initial condition $x_0 = .35$ and $R = 1.6$. What is the fixed-point attractor for the Logistic map with these parameters?

2. Download SimplePopulationGrowth.nlogo from the Course Materials page. Do the following modifications of the code, and see if they work!

- Change the shape of the reproducing individuals from bunnies to another shape. (To see the list of possible shapes, go to the "observer >" box below the Command Center window, and type "show shapes", followed by a carriage return, or see the list of "default shapes" at http://ccl.northwestern.edu/netlogo/docs/index2.html
- Change the color of the patches, using "set pcolor green" or whatever color you prefer. The list of color names you can use includes: black, gray, white, red, orange, brown, yellow, green, lime, turquoise, cyan, sky, blue, violet, magenta, pink.
- If you're even more ambitious, add a "Monitor" box to the interface, similar to the "count bunnies" box, but call it "cumulative bunnies", and have the program use it to count how many total bunnies have been born so far in a given run.

Intermediate Options:

1. Prove algebraically that $x_{t+1} = 2\,(x_t - x_t^2)$ has fixed point 0.5 .

2. Download and modify SimpleLogisticMap2.nlogo (available on the Course Materials page) as follows: Add a third initial condition, x_0'', that can be set by a slider and that is updated and plotted by the program in the same way that x_0' is updated and plotted, but with a color different from red or blue.

3. Further modify SimpleLogisticMap2.nlogo to color the background green, and have each of the three dots show a label with its coordinates (x_t, x_{t+1}). (Hint: use the "set label" command in an "ask turtles" section.)

Advanced Option (for people who finished the intermediate part and want to go deeper into NetLogo programming):

Build a model that has the following features:

- A slider to set the initial population.
- Turtles that reproduce (no mating necessary) with a reproduction rate set be a slider on the interface.
- Some aspect of the environment (food, space, etc.) that either limits reproduction or increases the death-rate.
- A plot that shows the population as it changes over time.
- Your code should contain as least three procedures/reporters.

Hints:

If you are new to programming and/or new to NetLogo creating this model may be quite challenging. Use the NetLogo dictionary and look at samples for the model library to get ideas. Persevere. Building a model that works is very rewarding.

It is often useful to start with pseudo-code to "outline" the situation. Most models have a high level "Go" button that runs other procedures. Perhaps your Go procedure will look like this

```
to go
ask turtles [
        look-around
        reproduce
        expire
]
end
```

Now the challenge is to just create the three simpler procedures.

You will likely want to "customize" the turtles with *turtles-own* variables. You do this be starting off the code with a block that looks something like this:

```
turtles-own [ energy-level]
```

After you do this, any turtle that gets created will have this variable. You can now say things like, "ask turtles [if energy-level = 0 [die]]" that is, to ask turtles to die if their energy-level gets to zero.

Finally, know that when you use the *create-turtles* command OR the *hatch* command, you have an opportunity to set variables in the newly created turtles. Hatched turtles will have all the attributes of the parents, including turtles-own variables. Here's an example that assumes that a *turtles-own* "age" variable has been created.

```
ask turtles with [color = green and age = 24 ] [ hatch 1 [set color red]]
```

I have asked a subset of the turtles (an "*agentset*" in NetLogo parlance) to each have one offspring. Green, 24 year old turtles will have red offspring. Unless you add additional code, all of these newborns will also have an *age* variable set at 24. Probably not what we intended.

Good luck with this assignment. Try to use online resources and the NetLogo dictionary and models library (File -> Models) when you get stuck. Getting stuck is inevitable but it is what makes getting unstuck fun. Only as a last resort, email us with code questions--pasting the *entire* code in the email with problem area in bold. We cannot guarantee that we will have the time to answer each and every question. In a week or two we will post a video that explains an example program that would satisfy this assignment.