The Computational Beauty of Nature – Spring 2020

Lab 3: Dynamics and Chaos

In this lab you will investigate the dynamics of the logistic map and, in particular, the degree of sensitive dependence on initial conditions at different values of the control parameter R. You will then extend your studies to the sine map. You should write down your answers to each exercise as you go. You will submit your answers to the starred (**) exercises with your next homework assignment.

1. Download **logistic.nlogo** from the class web page to your Desktop, and open it in NetLogo. Then click *setup*. The window on the right shows a graph of the logistic map $x_{t+1} = R\, x_t\, (1 - x_t)$, with $R = 2$. The controls on the left allow you to iterate the equation starting from two different initial conditions $x0$ and $x0'$, so you can watch their trajectories simultaneously. You can also change the value of R.

   Use the slider controls to adjust the values of $x0$ and $x0'$, and then click *setup*. <u>IMPORTANT: you should always click *setup* after modifying any parameter settings.</u> The two initial values will appear as colored dots on the graph (blue is $x0$, red is $x0'$). Now click *go* repeatedly to iterate the values. You will notice that both trajectories quickly converge to 0.5, where they remain fixed thereafter. This is because 0.5 is a *fixed point* of the equation when $R = 2$. Try starting from several other initial values of $x0$ and $x0'$ for comparison (remember to click *setup* each time), including initial values close to 0.5 and far away from 0.5.

2. You can also change a parameter setting by typing the command **set *parameterName newValue*** at the observer> prompt at the bottom of the control panel, substituting the actual name and value for *parameterName* and *newValue*. This is useful for setting a parameter to a precise value. For example, to change the value of R to 2.5, type **set R 2.5** and hit Return, and then click *setup*. Notice that the height of the parabola changes (in fact, you can think of R as a "knob" that controls the parabola's shape). What happens when you iterate with $R = 2.5$, starting from several different initial values of $x0$ and $x0'$? Do you reach the same fixed point as before? How quickly do the trajectories converge?

3. Now repeat the experiment for $R = 2.7$, 2.8, and 2.9. As R gets closer to 3.0, what happens to the value of the fixed point? What about the average time it takes to completely converge to the fixed point?

4. Above $R = 3.0$, the behavior of the equation changes. Set R to 3.2 and repeat the experiment. This time, instead of converging to a fixed point, the trajectories converge to a limit cycle of period 2, oscillating back and forth between two values. What are those values? Do you always end up in the same limit cycle if you start from different initial values of $x0$ and $x0'$? Try a few values to find out.

5. As you have already seen, when $R < 3.0$, there is a single fixed point that pulls all trajectories into it, no matter their starting value. We call such a fixed point *stable* or *attracting*. But *unstable* or *repelling* fixed points can also exist. For example, when $R = 3.2$, the value 0.6875 is an unstable fixed point. To see this, first set the *plot-x0'?* switch to Off, so we can focus just on the $x0$ trajectory. Then set $x0$ to 0.6875 and iterate, to verify that it is indeed a fixed point. Next, try starting from 0.67 or 0.69 to see what happens. Also try starting from 0.68751, which is just 0.00001 away from the unstable fixed point. How long does this trajectory "linger" near the fixed point before eventually wandering away?

6. If we turn up R to 3.3, is 0.6875 still a fixed point? If so, is it still unstable? If not, what happens to its trajectory, compared to the case when $R = 3.2$?

7. As it turns out, an unstable fixed point does indeed exist when $R = 3.3$, but its value is slightly larger than before. In fact, its value is precisely 0.696969696969.... Unfortunately, we cannot represent this value exactly in our simulator, since we have only a finite number of decimal places available. Try setting *x0* to this value anyway, using a large number of decimal places. How long does the trajectory remain "fixed", before eventually wandering away to join the period-2 limit cycle attractor?

8. Now turn up R to 3.5 and iterate from a few different starting points. Is the behavior of the system still periodic? If so, what is the period? Try the same thing with $R = 3.56$.

9. (**) Above an R value of about 3.6, the behavior of the system becomes *chaotic*. One of the hallmarks of chaos is extreme sensitivity to initial conditions, meaning that the system, when started from two very similar initial values, will nevertheless produce very different behavior in each case. To investigate this, first set the *plot-x0'?* switch back to On, so that we can easily compare the behavior of two trajectories. Then, for each of the values of $R = 3.6$, 3.7, 3.8, 3.9, and 4.0 (five different values), create plots showing x versus time, as follows:

   - Set *x0* to 0.2
   - Set *x0'* to 0.20000001
   - Set R to the appropriate value
   - Click *setup*
   - Repeatedly click *go* until the plots corresponding to the two initial conditions have separated. You can mouse over the plot to see the coordinates of each point.
   - Save your plot by right-clicking (or control-clicking) on it and choosing *Copy Image*, then pasting it into a Word or OpenOffice document. Don't forget to record the value of R used to create the plot.

   For each of the five R values, record the time at which the plots corresponding to the two initial conditions just begin to separate (this is the "time to separation"). Once you have recorded these times for the five R values, plot them on a single graph, showing the separation time as a function of R.

10. (**) Now download and open the file **sine.nlogo**, which implements the *sine map*: $x_{t+1} = (R/4)\sin(\pi x_t)$, where x is between 0 and 1 and R is between 0 and 4. Redo the steps of the previous exercise using the sine map model. To what extent is the behavior of the sine map similar to or different from the logistic map?

11. Strangely, deep inside the chaotic region between $R = 3.6$ and $R = 4.0$, "islands of order" can suddenly appear. One such region for the logistic map is around an R value of 3.84. Reload the logistic map model, and try starting with *x0* = 0.2 and *x0'* = 0.199 to see what happens. What is the resulting behavior of the system? How sensitive is the system to initial conditions? What happens if you turn R down to 3.82, or up to 3.86?

12. The sine map also exhibits a similar type of unexpectedly predictable behavior around $R = 3.76$. What happens to sine map trajectories at this value of R? Experiment with other nearby values of R to determine the approximate width of this "window of stability".