

Introduction to Chaos and the Logistic Map

CS523 Assignment One - Spring 2013

The University of New Mexico

Karl Stolleis
stolleis@unm.edu

John Ericksen
johncarl@unm.edu

Introduction

The following report is an introduction to the concepts of the logistic map function and its application to complex systems and chaos theory. The initial equation for the logistic map is as follows :

$$P_{(t+1)} = R(P_t - P_t^2/k)$$

where the population P at a given time step t is used to calculate the population at the next time step

$P_{(t+1)}$ in the sequence. The value R is the the birth rate of the population minus the death rate. The constant k is the “carrying capacity,” or the number of individual organisms, that the organism's environment can support. [1] With some algebraic maneuvering the equation can be rewritten in the form:

$$P_{(t+1)} = R P_t (1 - P_t)$$

where the value P is now the population given as a percentage of the total carrying capacity. [1] We will use both equations in this paper to demonstrate various principles.

The logistic map is used as a basic demonstration of a mathematical equation that, despite its simplicity, can give rise to chaotic behavior. Chaos, in this context, simply means that the values generated by the logistic map, at a given R-value, will become unpredictable. Unlike the common definition of chaos, the chaos described by dynamical systems is really unpredictability arising from a deterministic structure, such as a mathematical equation. [2] The logistic map also demonstrates a hallmark of chaotic systems, namely high sensitivity to initial conditions. Each of these principles will be discussed in the results section.

The connection of the logistic map to complex, adaptive systems has a number of points. The equation demonstrates that some deterministic systems can exhibit behavior that, under no circumstances, can be predicted. Many such systems, such as weather, biological populations, fluid turbulence and economic systems exhibit classic, chaotic behavior, the principles of which can be observed in the logistic map. [3] We will demonstrate such hallmarks as the Feigenbaum constant, self-similarity and how these chaotic behaviors can be

described in terms of Shannon entropy. Lastly we will give an example of how, when compared to an actual, biological system, the logistic map is a simple and naive method to model populations.

Methods

Each of the data plots in the results section were generated using the software package Matlab. A script was written for each of the plots shown.

In part one, population predictions for a theoretical rabbit population were generated using using a set of two different data inputs. The script `cs523_1_1.m` uses both the `logisticMap.m` and `logisticPopulation.m` functions to generate the data for questions one, two and three of part one. The resulting data sets are created by simply varying the input parameters for the listed functions, namely the R-value, the carrying capacity, the initial population and the number of time steps to be generated. The actual functions were simply the two listed in the introduction of this paper.

Question four was generated with `cs523_1_4a.m`, `cs523_1_4b.m`, `cs523_1_4c.m`, and `bifurcation.m` scripts. The bifurcation diagram displays the behavior of the logistic map at a large time value threshold. The script executes the logistic map function until the threshold is met, then samples the next one hundred data points. If the logistic map has reached a stable state (cycling through a set of values) then the resulting data points contains less than the sample amount of unique values. If the logistic map has not reached stability, all of the data points are unique and represents chaotic behavior. These points are plotted over each R value, from -2 to 4.

Question five was generated with the `cs523_1_5.m` script. The method for calculating the Shannon entropy was to generate a set of population predictions using a series of R-values between 2.5 and 4.0 and then create a histogram of the resulting values, for each R-value. The values were rounded to five decimal places to compensate for small rounding errors in the floating point representation. The histogram values were normalized, to create a set of probabilities and then the Shannon entropy equation :

$$H(x) = - \sum p(x) * \log_2(p(x))$$

where $H(x)$ is the calculated entropy, for each R-value and $p(x)$ is the probability of the expected values

in the population estimate. The process was repeated for each R-value, and the resulting H-values were plotted.
[4]

Questions six(a,b) of part two were generated with the scripts `cs523_2_1.m` and `cs523_2_2.m`, respectively. Both scripts, in turn, used the `X.csv` and `Y.csv` data sets and `logisticMap.m` and `logisticPopulation.m` functions to first plot the initial data set and then plot the calculated population estimate. The question asked us to estimate R-value, initial population and carrying capacity such that the initial data set could be reproduced. Analysis of the initial plot of X allowed for only a single, possible R-value and initial population. Analysis also showed a small range of possible carrying capacities which were iterated over until the correct value was found. Correctness was simply determined by how well the resulting plot matched the initial plot. A similar process for the Y data set was used with only a few more steps because of three additional, possible R-values.

Question seven was plotted first from given data set `Z.csv`, and using script `523_2_3.m` a discrete Fourier series was generated to calculate the exact periodicity of the data set. [5][6] An additional script `cs523_2_3a.m` was created to try and use the logistic map equation to recreate the data but the resulting plot was omitted for brevity.

Included in the directory, with the source scripts, is a read me file that outlines the detailed instructions for running the above mentioned scripts. Additional details can be found in the comments included in the individual script files.

Author Contributions

Initial plots for 1.1, 1.2, 1.3, 2.1, 2.2, 2.3 were generated by Karl S. while 1.4(a,b,c,d) and 1.5 were generated by John E. Both authors then independently confirmed the work of the other and compared results. Report draft was written by Karl S., subsequently modified by both authors and final draft was completed by John E.. On-line collaboration was used extensively in comparing data and results.

Results

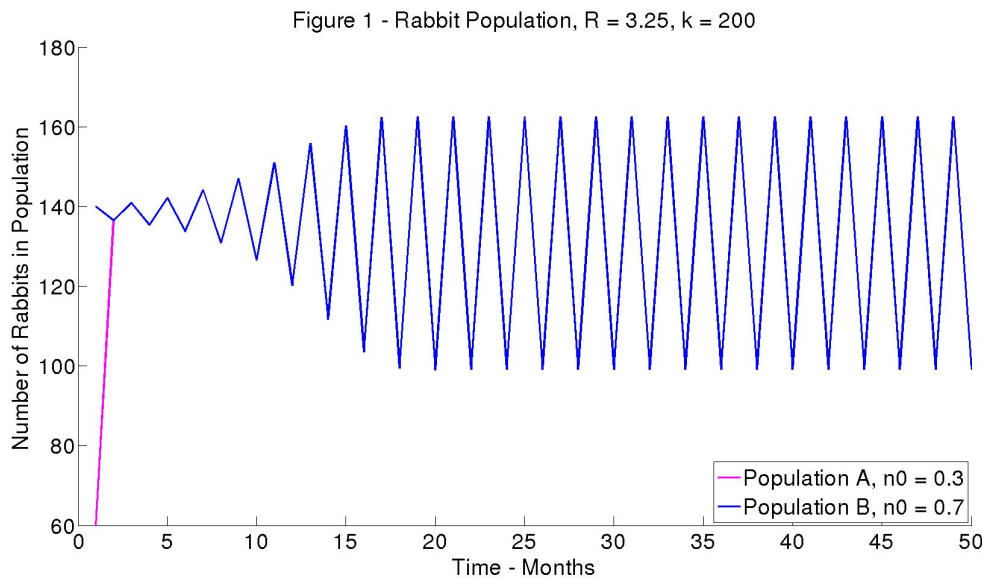


Figure 1 : Plot of two theoretical, rabbit populations with an R -value set below the chaotic threshold. Both populations synchronize quickly despite having different initial values.

Figure 1 shows the values plotted for the two, proposed rabbit populations for part one, question one.

The R -value is 3.25 which is below the chaotic threshold of ~ 3.569 [2] so the expected result was that the two populations would stabilize with a two attractor period and the values would synchronize after a large number of time steps. The plot clearly shows the resulting two attractor state and the population values synchronized after a single time step, despite not having reached the final attractor values. The assumptions underlying our predictions stem from a R -value below the chaotic threshold, the carrying capacity being identical for both populations and the initial values chosen that converge immediately to the same population cycles.

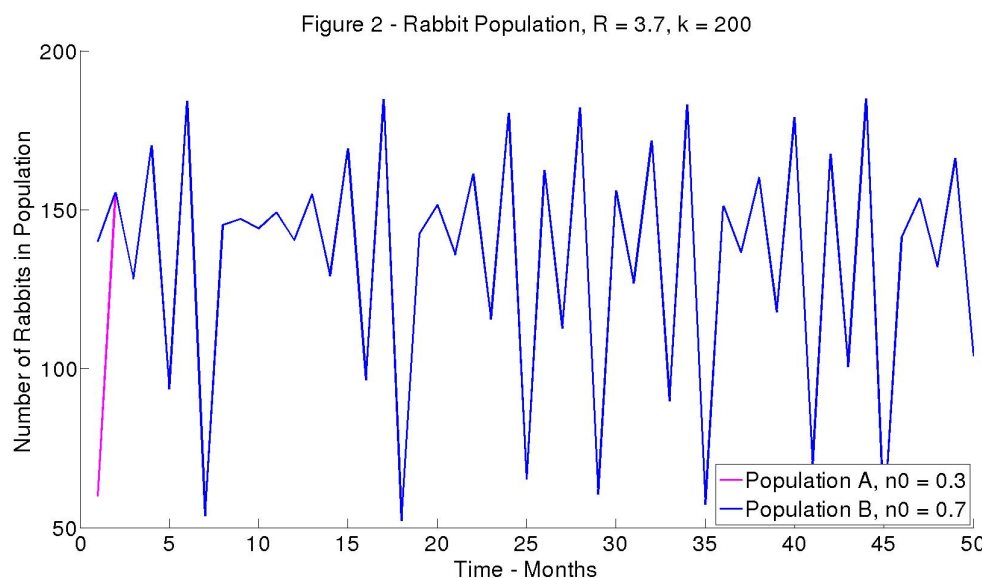


Figure 2 : Plot of two theoretical, rabbit populations with an R -value set above the chaotic threshold. Both populations unexpectedly synchronize despite having unique initial values. This behavior is explained by the sum of the initial values being equal to one.

Figure 2 shows the same process as for Figure 1 but having an R-value of 3.7, above the chaotic threshold of the logistic map equation. From the plot it is easy to see that the two populations do not reach a steady attractor state, despite what appears to be quasi-periodic (ergodic) regions of the plot. The surprising result was that the two populations again became synchronized after a single time step, belying the sensitivity of the equation to initial conditions while in the chaotic region. The synchronization can be explained by the initial values, (x_{0a}, x_{0b}) being 0.3 and 0.7 respectively, summing to 1.0. Additional test plots, with other analogous values confirmed this. The logistic map equation where the current population value is multiplied by one minus the current value is the ultimate explanation for this behavior. The actual population dynamics show the classic unpredictability of having an R-value which exceeds the chaos boundary for this equation. It would be highly improbable that this particular condition, of synchronization, would occur in a real population.

Figure 3 - Rabbit Population, R = 3.7, k = 200

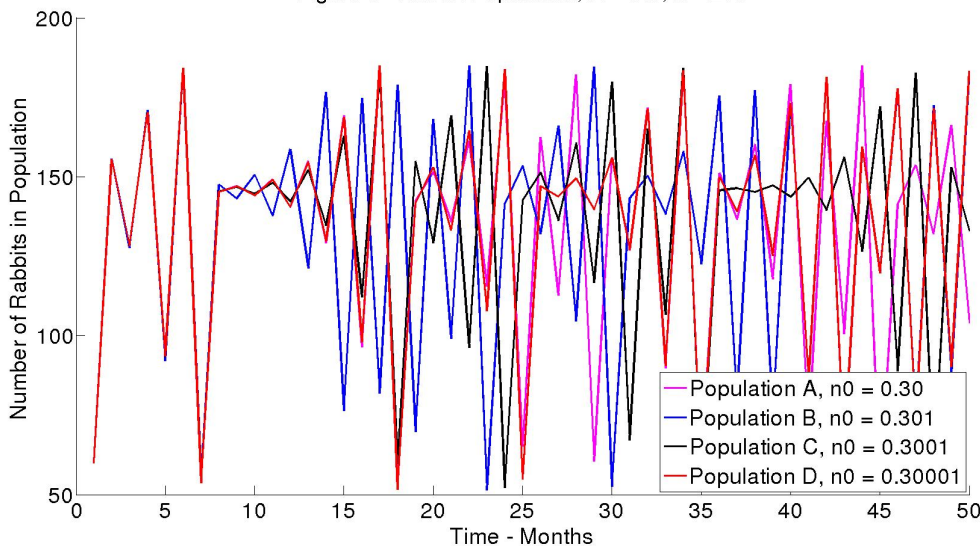


Figure 3 : Plot of four theoretical, rabbit populations with an R-value set above the chaotic threshold. The sensitivity to initial conditions is demonstrated by the divergence of the four predictions despite the initial values being quite similar in value.

Figure 3 again uses a similar process as Figures 1 and 2 with four estimated populations. The R-value, in this case is set above the chaotic threshold but the initial populations are perturbed by a small amount. The result shows the four populations shadowing one another for approximately the first seven time steps but then diverging into individual, chaotic paths that show no overlap or any signs of synchronization across the populations. The sensitivity to initial conditions is exacerbated by the inability of a computer to represent numbers with infinite precision. The rounding errors created by floating point representations, in conventional computer architecture, can also create the above behavior.

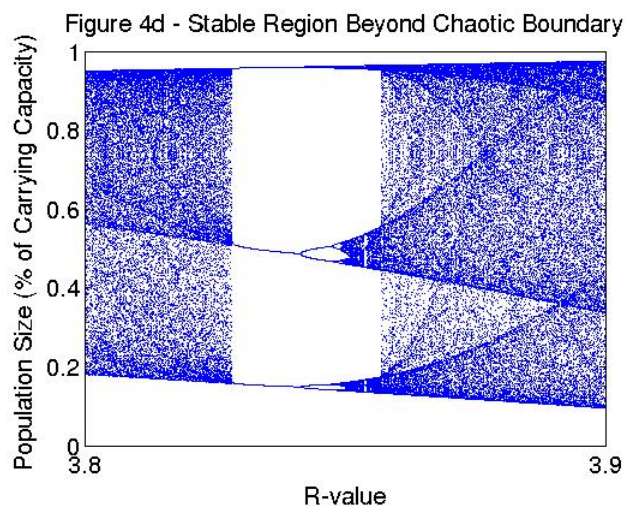
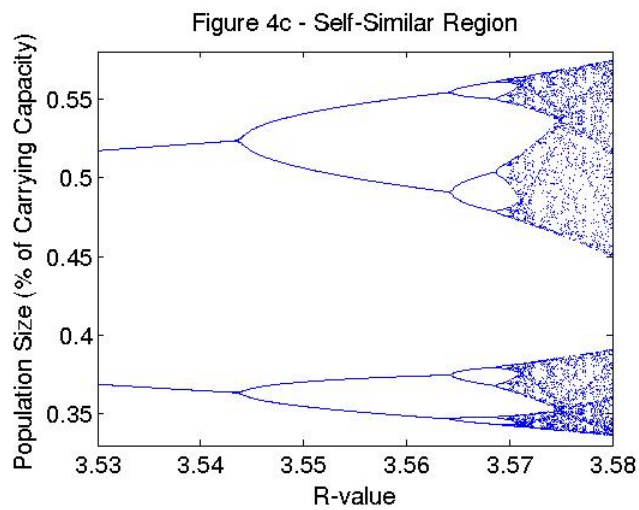
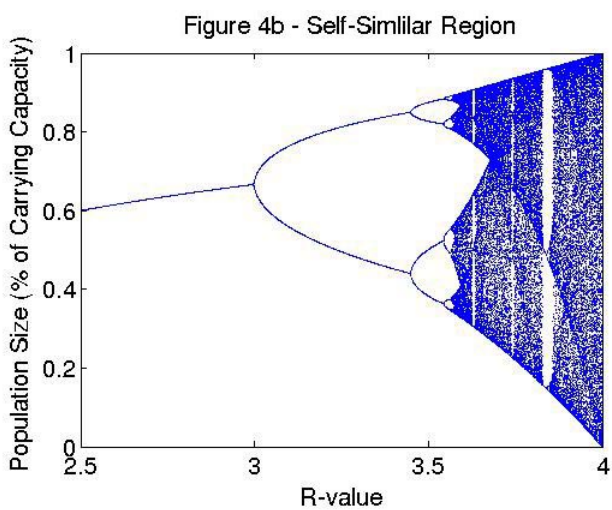
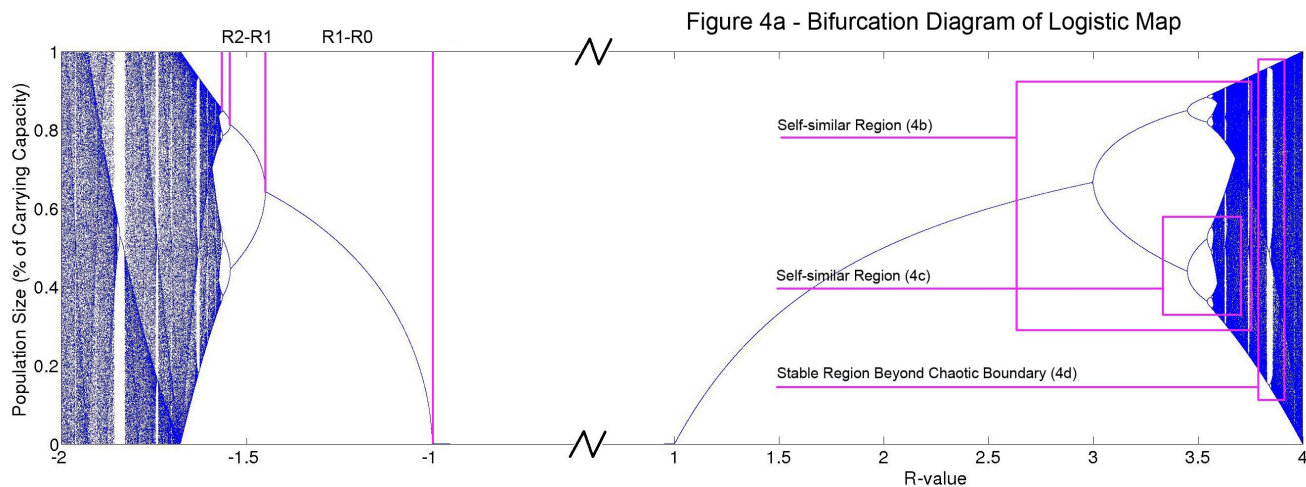


Figure 4a (top) : Plot of bifurcation diagram showing areas of self-similarity, stable regions and regions contributing to calculation of Feigenbaum constant.

Figures 4b, 4c (middle) : Areas of "fractal" self-similarity within the diagram

Figure 4d (left) : Region of apparent stability beyond chaotic boundary.

$$\lim_{n \rightarrow \infty} \frac{(R_{(n-1)} - R_{(n-2)})}{(R_{(n)} - R_{(n-1)})}$$

Formula for Feigenbaum constant.

Figure 4a shows the bifurcation diagram generated from the logistic map function ranging from -2 to 4. This plot shows the areas of stability and instability over values of R, including the negative region. [10] The

positive region of the map shows the stability and chaotic behavior of the logistic map for the various values of R . R , in the positive case represents a birth rate higher than the death rate of the population. When R is negative the population is stable at 0 between $-1 < R < 0$, then for $R < -1$ the population stability splits and ranges between values $-0.5 < P < 1.5$. Obviously this does not represent a physical population as there is a hard lower bound on the population size, but theoretically if a negative population was possible there would be a similar logistic map splitting into a chaotic region as in the positive direction. What is noteworthy about this plot is the self-similarity. Each plot split (from 1 to 2, 2 to 4, 4 to 8) has a similar branch visible in Figure 4c. Each split happens a rate of ~ 4.669 as fast as the previous split, Feignbaum's constant. [9]

Another noteworthy feature of the bifurcation diagram is the region of stability between $3.8 < R < 3.9$ seen in Figure 4d. As highlighted, the bifurcation diagram shows that the behavior of the logistic map suddenly becomes stable at two values. These values split in a similar way as the stable regions below ~ 3.569 and proceed to fall into another chaotic region at about $R > 3.87$.

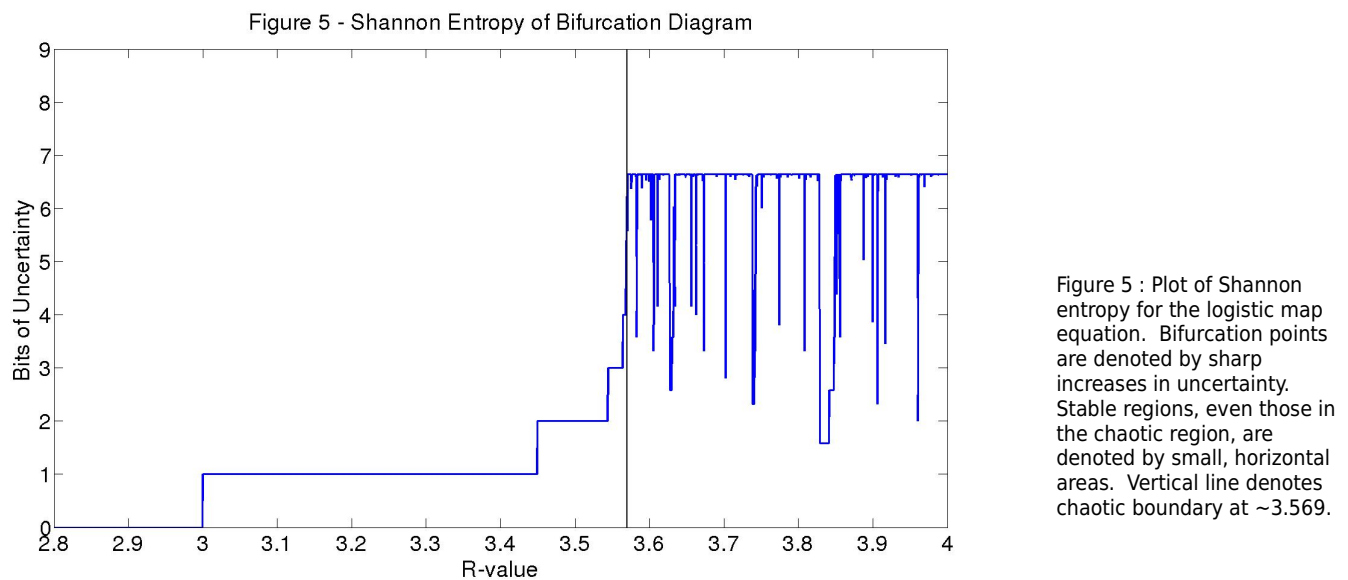


Figure 5 represents a plot of Shannon entropy for the bifurcation diagram discussed in Figures 4. The assumptions underlying the calculation of $H(x)$ are that the population being modeled has been in existence long enough to reach the stable attractor state. The predicted behavior of this graph was that entropy would start equal to zero in the region, of the bifurcation diagram, before an R -value equal to three. In the region of R less than three, the map shows a single attractor which leaves no uncertainty as to the attractor's value, thus having an

entropy of zero. With each bifurcation a sharp increase in entropy is shown followed by the entropy remaining at that level for some period of R . Again, this behavior is due to the attractors acquiring a steady state with two values. The pattern repeats with a “stair step,” corresponding with additional bifurcation points. When the chaos value for R is reached the plot become correspondingly chaotic. While in the chaotic region the probability of any value repeating would appear to be zero so entropy should be infinite but the this condition only exists as the number of time steps, in the prediction, tends to infinity. The fact that there are a finite number of time steps in the calculation means there is an apparent upper bound on entropy or uncertainty. Of note, in the chaotic region, is the appearance of regions that have a set of non-increasing, finite entropy. These regions correspond to the stable regions, which appear periodically, in the bifurcation diagram. Note also that the width of the regions of stable entropy, prior to the chaotic boundary, can be shown to follow the ratio outlined by Feigenbaum's constant.

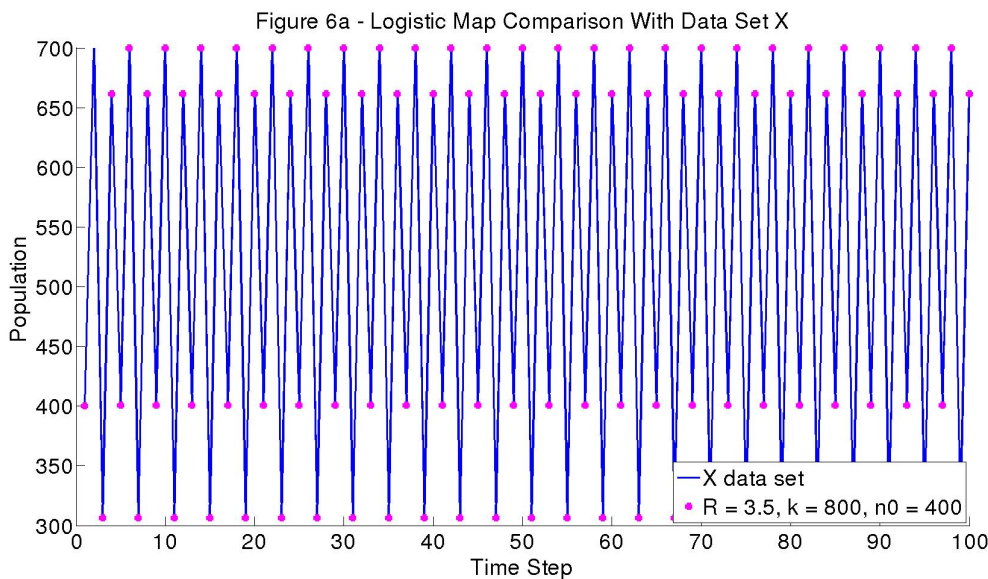


Figure 6a : Plot of given data set 'X' versus a second population estimate calculated with the shown input parameters. Note that the two estimates overlap completely.

Figure 6a shows a plot of the given data set “X,” assumed to have been generated using the logistic map equation and an estimate derived by the authors, respectively. The plot clearly shows, using methods previously outlined, that the original data set can be reproduced by correctly determining the input parameters used for the initial calculation.

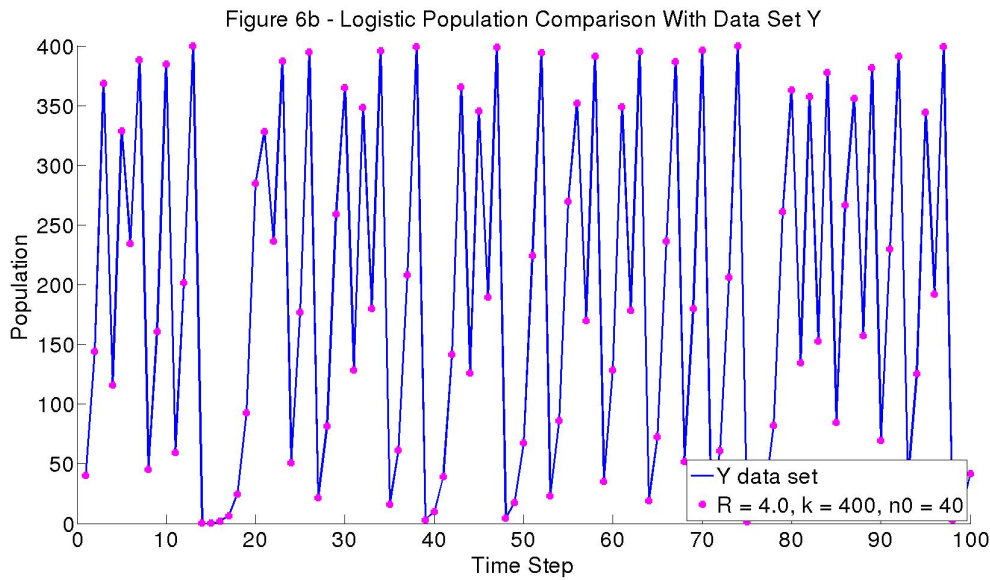


Figure 6b : Plot of given data set 'Y' and calculated parameter set showing complete overlap of data points despite having an R-value beyond the chaotic threshold.

Figure 6b shows a plot of the given data set “Y,” assumed to have been generated using the logistic map equation and a derived estimate. Again, despite having a chaotic R-value of 3.7, use of the exact, initial conditions were able to reproduce the original data plot.

Using the estimated parameters, for each data set, extrapolated values for the populations were calculated at time steps 105 and 120. The values for X were ~ 400.7074 and ~ 662.5526 and Y values were ~ 310.4546 and ~ 312.8679 , respectively. Given the behavior of the generated plots, with respect to the initial data sets, the confidence in these extrapolated values is high. For X the fact that R is below the chaotic threshold increases confidence even further, if not infinitely. For Y the fact that $R > \sim 3.569$ leaves some uncertainty about any extrapolation, especially at ever larger time step values.

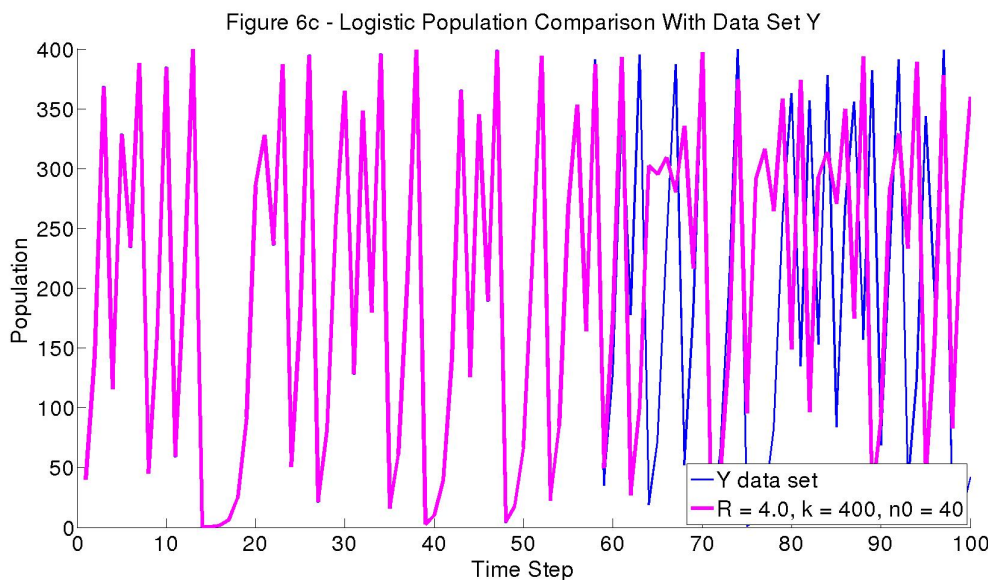


Figure 6c : Plot of given data set 'Y' and calculated parameter set showing divergence in chaotic region. Data used logistic population function with integer inputs demonstrating sensitivity to internal precision of computing system.

Figure 6c is included to show results when the estimated population used the correct parameters of the initial data set but divergence from the original data set appears around time step 55. The estimate in question was generated using the logistic population equation, rather than the logistic map equation. The difference being that initial condition and carrying capacity are integers, in this case, thus giving another example of internal computing precision having an effect on population calculations with a chaotic R-value.

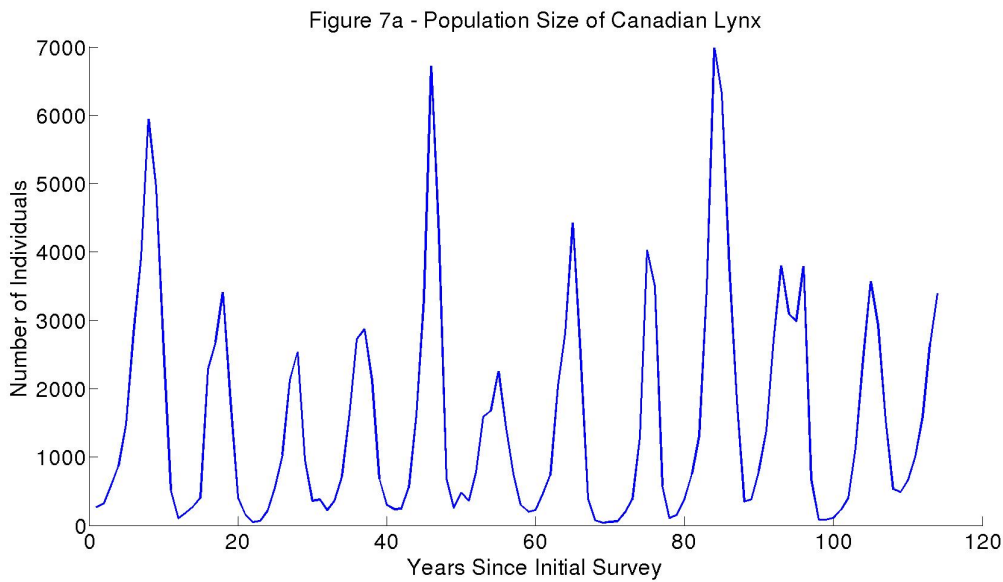


Figure 7a : Plot of data set "Z." The data set is of real, observed values of Canadian Lynx populations in the wild. Note the obvious periods of approximately 10 years.

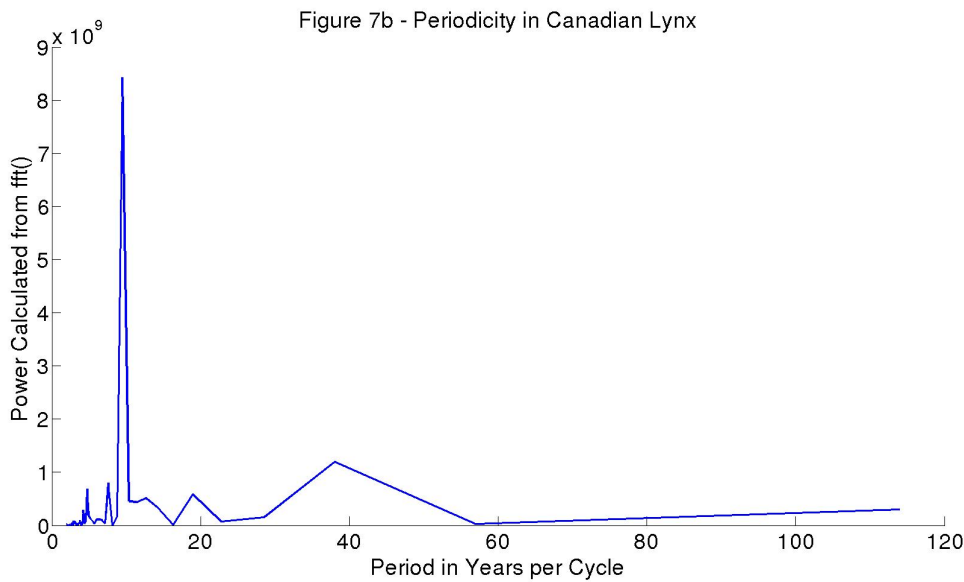


Figure 7b : Figure 7b : Discrete Fourier transform of Canadian Lynx data showing periodicity in data of 9.5 year cycle. Secondary period of 38 years is not as obvious but is present in plot.

Figures 7a and 7b show plots of actual population data of wild, Canadian Lynx and the results of a discrete Fourier transform of the data set. The initial data shows clear peaks in the population, above a value of 2000, with several peaks with a longer period above a threshold of 5000. The average period, of these peaks,

would appear to be 10 years and 40 years respectively. The Fourier series, clearly shows the higher frequency period that has an exact value of 9.50 years. A second, lower peak of approximately 38 years appears but given its value, in relation to the noise floor it is hard to say definitively that this is an actual, second frequency.

This data set, unlike the previous two, is taken from a real population. It demonstrates why the logistic map is a naive approximation of a real biological population. The Lynx data shows no clear, stable value that the upper values approach, which would indicate the existence of an attractor or population limit. One can also note that the logistic map produces, in stable regions, a plot that has upper and lower attractor values that are symmetric about a line drawn from a single y-axis value. The Lynx data could only have this symmetry if the values were allowed to become negative, indicating a negative population number. Another hint that the Lynx data is not representative of a logistic map is the high number of population numbers less than 1000, and low number of population numbers above 1000. This indicates that the Lynx population booms at intervals not based on the previous population, or the total carrying capacity of the population, but on external factors. Furthermore, the year by year Lynx population is relatively smooth. Again, this indicates that the Lynx are surviving and reproducing not because of a population maximum or reproductive rate, but on external factors.

Upon further analysis it has been shown that the wild Lynx population is tightly coupled with the population of the Snowshoe Hare, which is turn coupled with vegetation volume, in the habitat. Additional factors include hunting and trapping of Lynx by humans. [7][8] The additional complexity, stemming from these external, dynamic pressures, would indicate that the logistic map simply does not have enough variables to account for these factors.

Conclusions

This report covered a wide variety of complexity topics centered around the logistic map. We have highlighted the basic chaotic characteristics of a simple logistic map, and how that may be used to simulate a biological population. We have also highlighted where a logistic map fails to model a population accurately in the case of the Lynx. Logistic maps have a deep relationship with chaos and demonstrate the fascinating combination of stable, predictable behavior, and unstable chaotic behavior. Logistic maps and the associated bifurcation diagram demonstrate Feignbaum's constant, which is used to describe the self-similarity of the bifurcation plot.

References

- [1] M. Moses, "Logistic Map Lecture," Jan., 2013.
- [2] M. Mitchell, "Dynamics, Chaos and Prediction," in *Complexity, A Guided Tour*, 1st ed., New York: Oxford, 2011, pp. 15-39.
- [3] J. Gleick, "The Butterfly Effect," in *Chaos: Making a New Science*, New York: Open Road Integrated Media.
- [4] R. Yeung, "Shannon's Information Measures," in *A First Course in Information Theory*, 1st ed., New York: Springer, 2002, pp. 10-15.
- [5] M. Roberts, "The Discrete-Time Fourier Transform," in *Fundamentals of Signals & Systems*, 2nd ed., New York: McGraw-Hill, 2008, pp. 391-431.
- [6] et.al. (12/28/13), *Using FFT*, [Online], Available :<http://www.mathworks.com/products/matlab/examples.html>
- [7] N. Stenseth, et.al., "Population Regulation in Snowshoe Hare and Canadian Lynx," in *Proceedings of the National Academy of Sciences*, Oxford, Oxford, United Kingdom, 1997, pp. 5147-5152.
- [8] M. McCarthy, "The Lynx and the Snowshoe Hare," in *Winter Ecology*, Boulder, CO, 2005.
- [9] et.al. (12/29/13), *Feigenbaum Constants*, [On-line], Available: <http://en.wikipedia.org/wiki/>
- [10] et.al. (12/29/13), *Bifurcation Diagrams*, [On-line], Available: <http://en.wikipedia.org/wiki/>

Appendices

Attached

```
cs523_1_1.m
```

```
% Calculation of rabbit population using logistic map function. Output is  
% number of rabbits over time.  
% Karl Stolleis - CS523 - Spring 2013
```

```
clear, clc
```

```
hold on
```

```
% initial values
```

```
R = 3.7;
```

```
k = 200;
```

```
count = 50;
```

```
n0a = 0.30 * k;
```

```
n0b = 0.301 * k;
```

```
n0c = 0.3001 * k;
```

```
n0d = 0.30001 * k;
```

```
% calculate population estimates
```

```
% four values for part 3
```

```
P1 = logisticPopulation(R, k, n0a, count);
```

```
P2 = logisticPopulation(R, k, n0b, count);
```

```
P3 = logisticPopulation(R, k, n0c, count);
```

```
P4 = logisticPopulation(R, k, n0d, count);
```

```
% plot
```

```
plot(P1, 'r');
```

```
plot(P2, 'k');
```

```
plot(P3, 'b');
```

```
plot(P4, 'g');
```

```
set(gca, 'FontSize', 30);
```

```
title('Figure 3 - Rabbit Population, R = 3.7, k = 200');
```

```
xlabel('Time - Months');
```

```
ylabel('Number of Rabbits in Population');
```

```
% use this for four values
```

```
% legend('Population A, n0 = 0.30', 'Population B, n0 = 0.301', ...
```

```
%       'Population C, n0 = 0.3001', 'Population D, n0 = 0.30001', ...
```

```
%       'Location', 'SouthEast');
```

```
legend('Population A, n0 = 0.3', 'Population B, n0 = 0.7', 'Location', ...  
      'Southeast');
```

```
cs523_2_1.m
```

```
% Script calculates population using logistic map in attempt to match data  
% set X for Assignment One Part Two  
% CS523 - Karl Stolleis - Spring 2013
```

```
clear, clc
```

```
hold on
```

```
% initial values
```

```
R = 3.5;
```

```
k = 800;
```

```
n0 = 400;
```

```
count = 100;
```

```
X = csvread('X.csv');
```

```
plot(X, 'r');
```

```
% title('Figure 6a - Data Set X');
```

```
P = logisticPopulation(R,k,n0,count);
```

```
plot(P, 'k');
```

```
set(gca, 'FontSize', 30);
```

```
title('Figure 6b - Logistic Map Comparison With Data Set X');
```

```
xlabel('Time Step');
```

```
ylabel('Population');
```

```
legend('X data set', 'R = 3.5, k = 800, n0 = 400', 'Location', 'SouthEast');
```

```
cs523_2_2.m
```

```
% Script calculates population using logistic map in attempt to match data  
% set Y for Assignment One Part Two  
% CS523 - Karl Stolleis - Spring 2013
```

```
clear, clc
```

```
hold on
```

```
% initial values
```

```
R = 4.0;
```

```
k = 400;
```

```
n0 = 40;
```

```
x0 = .1;
```

```
count = 120;
```

```
Y = csvread('Y.csv');
```

```
plot(Y, 'r');
```

```
%title('Figure 6c - Data Set Y');
```

```
% note that logPop and logMap functions give different values
```

```
%P = logisticPopulation(R,k,n0,count);
```

```
P = logisticMap(R, x0, count);
```

```
P = P * 400;
```

```
plot(P, 'k');
```

```
set(gca, 'FontSize', 30);
```

```
title('Figure 6e - Logistic Population Comparison With Data Set Y');
```

```
xlabel('Time Step');
```

```
ylabel('Population');
```

```
legend('R = 4.0, k = 400, n0 = 40', 'Y data set', 'Location', 'SouthEast');
```


cs523_2_3.m

```
% Calculation of periodicity of Canadian Lynx data using Fourier Transform.
% The following code is modified from a Mathworks tutorial on data set
% analysis using the fft() function in Matlab.
% http://www.mathworks.com/products/matlab/examples.html?
file=/products/demos/shipping/matlab/sunspots.html
% Karl Stolleis - CS523 - Spring 2013
```

```
clear, clc
```

```
hold on
```

```
Z = csvread('Z.csv');
```

```
% This section is to plot the initial population data
%
% plot(Z,'k');
%
% set(gca, 'FontSize', 30);
% title('Figure 7a - Population Size of Canadian Lynx');
% ylabel('Number of Individuals');
% xlabel('Years Since Initial Survey');
```

```
% Fourier series
```

```
F = fft(Z);
n = length(F);
```

```
F(1) = [];
```

```
% change back to time domain
power = abs(F(1 : n/2)).^2;
nyquistRate = 1/2;
frequency = (1 : n/2) / (n/2) * nyquistRate;
```

```
% invert to get period from frequency
period = 1./frequency;
plot(period, power, 'k');
```

```
set(gca, 'FontSize', 30);
title('Figure 7b - Periodicity in Canadian Lynx');
ylabel('Power Calculated from fft()');
xlabel('Period in Years per Cycle');
```

```
[maxPower, index] = max(power);
maxCycle = period(index);
```

```
fprintf('Years/Cycle of Canadian Lynx Population = %.2f\n\n', maxCycle);
```

logisticPopulation.m

```
function [P] = logisticPopulation(R, k, n0, count)

%logisticPopulation() calculates population based on logistic map equation
% logisticPopulation() uses the logistic map equation to calculate the
% number of individuals in a population at over a given period of time.
% The function takes the R-value (births - deaths), a k-value (carrying
% capacity of environment), n0 (the initial count of individuals) and
% count (the number of iterations to perform on population).
% The function returns a vector of the actual number of individuals, in
% the population, at time step i.
% R - float value between 0.0 and 4.0
% k = int value between 0 and maxInt
% n0 = float value between 0.0 and inf
% count = int value between 0 and maxInt
% Karl Stolleis - CS523 - Spring 2013

P = zeros(1, count);
P(1) = n0;

for i = 1 : count - 1
    P(i+1) = R * (P(i) - (P(i)^2 / k));
end

end
```

logisticMap.m

```
function [X] = logisticMap(R, x0, count)

%logisticMap() calculates percentage of carrying capacity based on logistic map equation
% logisticMap() uses the logistic map equation to calculate the
% percentage of individuals in a population, as a percentage of carrying
% capacity, over time.
% The function takes the R-value (births - deaths),n0 (the initial count of individuals) and
% count (the number of iterations to perform on population).
% The function returns a vector of the population size, as a percentage
% of carrying capacity for time step i.
% R - float value between 0.0 and 4.0
% n0 = float value between 0.0 and inf
% count = int value between 0 and maxInt
% Karl Stolleis - CS523 - Spring 2013

X = zeros(1, count);
X(1) = x0;

for i = 1 : count - 1
    X(i+1) = R * X(i) * (1 - X(i));
end

end
```

```
cs523_1_4(a,b,c,d).m
```

```
% Script to display the bifurcation diagram plot using the bifurcation.m function.  
% John Ericksen - CS523 - Spring 2013
```

```
% Large view of bifurcation diagram, including negative region  
[X, R] = bifurcation(-2, 4, 0.0001);
```

```
plot(X,R, 'b.', 'MarkerSize',0.1)
```

```
set(gca, 'FontSize', 30);  
title('Figure 4a - Bifurcation Diagram of Logistic Map');  
xlabel('R-value');  
ylabel('Population Size (% of Carrying Capacity)');  
axis([-2,4,0,1]);
```

```
% Script to display the bifurcation diagram plot using the bifurcation.m function.  
% John Ericksen - CS523 - Spring 2013
```

```
% Zoom in to the splitting from 1 -> 2 branches  
[X, R] = bifurcation(2.85, 3.1, 0.0001);
```

```
plot(X,R, 'b.', 'MarkerSize',0.1)
```

```
set(gca, 'FontSize', 16);  
set(gca, 'XTick', [2.8:.1:4]);  
title('Figure 4b - Self-Similar Region');  
xlabel('R-value');  
ylabel('Population Size (% of Carrying Capacity)');  
axis([2.85,3.1,.5,.78]);
```

```
% Script to display the bifurcation diagram plot using the bifurcation.m function.  
% John Ericksen - CS523 - Spring 2013
```

```
% Zoom in to the splitting from 2 branches to the chaotic state  
[X, R] = bifurcation(3.53, 3.58, 0.0001);
```

```
plot(X,R, 'b.', 'MarkerSize',0.1)
```

```
set(gca, 'FontSize', 16);  
set(gca, 'XTick', [2.8:.01:4]);  
title('Figure 4c - Self-Similar Region');  
xlabel('R-value');  
ylabel('Population Size (% of Carrying Capacity)');  
axis([3.53,3.58,.33,.58]);
```

```
% Script to display the bifurcation diagram plot using the bifurcation.m function.  
% John Ericksen - CS523 - Spring 2013
```

```
% Zoom in to the splitting from 2 branches to the chaotic state  
[X, R] = bifurcation(3.8, 3.9, 0.0001);
```

```
plot(X,R, 'b.', 'MarkerSize',0.1)
```

```
set(gca, 'FontSize', 16);  
set(gca, 'XTick', [2.8:.1:4]);  
title('Figure 4d - Stable Region Beyond Chaotic Boundary');  
xlabel('R-value');  
ylabel('Population Size (% of Carrying Capacity)');  
axis([3.8,3.9,0,1]);
```

cs523_1_5.m

```
% Calculates and displays the bifurcation plot shannon entropy
% John Ericksen - CS523 - Spring 2013

threshold = 50000;
sampleSize = 100;
iter = 2.5:0.0005:4;

E = zeros(size(iter));
ni = 1;

% Iterates over 2.5 - 4, displaying a sample of points at high time
% values > threshold.
for n = iter
    X = zeros(0,sampleSize);
    sample = 0.1;
    i = 1;
    for t = 1:threshold+sampleSize
        lastsample = sample;
        sample = n * lastsample * (1 - lastsample);

        if t > threshold
            X(i) = sample;
            i = i+1;
        end
    end

    % Round, so we may associate like points
    X = round(X.*(10^5))/(10^5);
    V = hist(X, unique(X));
    Vp = V / sampleSize;

    % Shannon entropy  $H(x) = -\sum P(x) \log_2(p(x))$ 
    E(ni) = -sum(Vp.*log2(Vp));
    ni = ni + 1;
end

% Plot bifurcation diagram entropy
plot(iter,E,'b', 'LineWidth', 2)
line([3.569945672,3.569945672],[0,10], 'Color', 'k', 'LineWidth', 2);

set(gca, 'FontSize', 30);
set(gca, 'XTick', [2.8:.1:4]);
title('Figure 5 - Shannon Entropy of Bifurcation Diagram');
xlabel('R-value');
ylabel('Bits of Uncertainty');
axis([2.8,4.0,0,9]);
```

bifurcation.m

```
function [X,R] = bifurcation(Xmin, Xmax, resolution)

% Function to calculate the bifurcation values between Xmin and Xmax R values,
% with the provided resolution.
% CS523 - John Ericksen - Spring 2013

threshold = 700;
sampleSize = 100;

i = 1;

X = 0;
R = 0;

for n = Xmin:resolution:Xmax
    sample = 0.1;
    for t = 1:threshold+sampleSize
        lastsample = sample;
        sample = n * lastsample * (1 - lastsample);
        if lastsample == sample
            X(i) = n;
            R(i) = sample;
            i = i+1;
            break;
        end
        if t > threshold
            % Collect the sample after the threshold is met. This allows the
            % logistic map to reach stability.
            X(i) = n;
            R(i) = sample;
            i = i+1;
        end
    end
end

end

end
```