Project 1: Logistic Map and Chaotic Dynamics

Jason Loyd, Amanda Minnich, and Oscar Mondragon
Computer Science Department
University of New Mexico
{jloyd, aminnich, omondrag}@cs.unm.edu

February 4, 2013

1 Author Contributions

Mondragon wrote the code for problems 1-4, Minnich wrote the code for problems 5 and 7, and Loyd wrote the code for problem 6. Mondragon, Minnich, and Loyd wrote the Methods and Results sections and edited the report together. Mondragon wrote the ReadMe and formatted the report and figures, Loyd wrote the Introduction, and Minnich wrote the Conclusion.

2 Introduction

Our goal for this report is to discuss the logistic model and how well it does or does not describe population data sets and their corresponding dynamics. In general, a logistic model predicts population values over time, x_t, as a function of the initial population, x_0; birth and death rates, encapsulated by R; and the carrying capacity, K, which is the maximum population supported by an environment. The logistic model is generated using the expression $x_{i+1} = Rx_i(1 - x_i)$.

One interesting characteristic of this model is that it defines behavior for both stable and chaotic systems, depending on the value of R. Chaotic systems are defined as systems with infinite periodicity and completely deterministic behavior, with periods of embedded stability and ergodicity. The logistic map does not have strong predictive power for chaotic systems over long time periods due to their sensitive dependence on initial conditions and sentivity to the precision of the measurement of the population at each time step. However, logistic map simulations do allow us to study and learn more about the behavior of chaotic systems, as well as to examine the aforementioned periods of embedded stability and ergodicity.

The logistic model is considered to be a reasonable model for complex adaptive systems with well-defined initial conditions. A complex adaptive system is generally defined to be a system in which many components, or even collections of components, with no hierarchical governance and non-complex individual behaviors demonstrate complex aggregate behavior, highly developed mechanisms for information processing, and adaptation as the result of learning or evolving [4]. In this report we employ the logistic model to more completely understand complex population phenomena.

3 Methods

For questions 1 to 3, in order to analyze the logistic map we wrote a function that calculates a population vector at discrete time steps using the logistic map equation. With the goal of finding synchronization points, behavior at different inital conditions, and sensitive dependence of numerical precision calculations, we varied input parameters to generate multiple plots. For many of the figures we drew multiple population vectors in the same figure using MATLAB's hold function to compare population dynamics when varying a single parameter. The code to round to different number of precision digits was adapted from [2].

To generate the bifurcation diagram for question 4 we used the *arrayfun* function of MATLAB. This function takes a function and an array as arguments and applies the function to each element of the array. In this case, the array contained values of R from 2.6 to 4 in 0.0005 increments, and the function applied to that array computes and plots the logistic map for each value of R. This generates a vector of x_t values for each entry in the array. To produce a clearer figure, we plotted the second half of the population vector for each R value, so as to only include asymptotic population values.

To see doubling regions in detail in the bifurcation diagram we enlarged the boxed region in Figure 4 in order to get Figure 5, and similarly enlarged the boxed region in Figure 5 to get Figure 6. The limits of Figure 5 were chosen to highlight the transition from 4 periodicity to chaos, and the limits of Figure 6 were chosen to highlight the transition from 8 periodicity to chaos. To generate Figure 7, which shows the chaotic region of the bifurcation diagram, we focused on a different region of the diagram. To highlight periods of embedded stability, we plotted multiple red vertical lines using Microsoft Paint and specified the period for each.

Figure 8 was generated using MATLAB's plot and line functions.

To obtain estimates for x_0, R, and K for question 6, we iterated over a range of values for each variable using MATLAB. At every iteration these values were passed to our *logmap* function. We rounded the resulting vector using MATLAB's built-in *round* function and compared it to the datasets obtained from X.csv and Y.csv, respectively. During each iteration generated values were compared to the values from the dataset. If these estimated values more closely resembled the values from the dataset than the previous best estimate, the baseline values were updated. Once our function generated optimal values, we plotted the dynamics of this estimated system alongside the time series data, generating Figures 9 and 10.

For dataset X, x_0 ranged from 0.44 to 0.52, R ranged from 3.44 to 3.52, and K ranged from 700 to 1000. For dataset Y, x_0 ranged from 0.08 to 0.15, R ranged from 3.8 to 4.1, and K ranged from 400 to 700. We chose these ranges based on visual inspection of the graphs of the datasets. We asserted minimum carrying capacities based on knowledge that the maximum population divided by carrying capacity could not be greater than 1. The maximum values were picked to be reasonably larger than the minimum values. The X time series was clearly 4 periodic, so according to the bifurcation diagram R must be restricted to values between 3.44 and 3.52. The Y data was chaotic, so the R values had to be at least 3.8, and 4.1 seemed like a sufficient upper bound. We chose the x_0 ranges to correspond to the carrying capacity ranges; the minimum x_0 value was the initial population value divided by the maximum carrying capacity, and the maximum x_0 was the initial population value divided by the minimum carrying capacity.

Figure 11 was created using MATLAB's *fft* function, followed by normalization. Instructions on this normalization method came from MATLAB's website [3].

4 Results

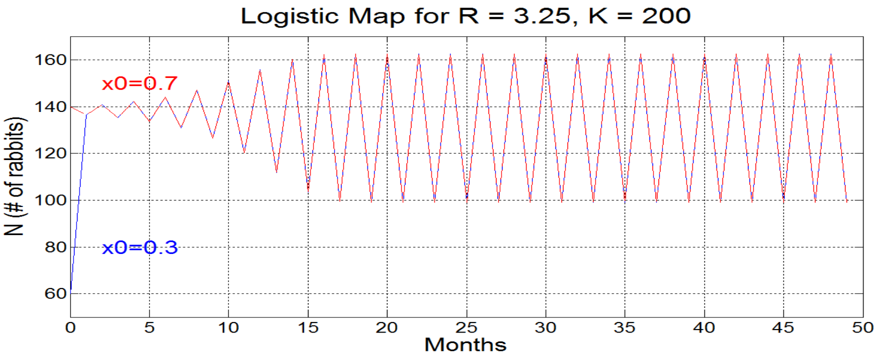

Figure 1: Logistic Map for R=3.25 and K=200

Figure 1 shows systems that are not sensitive to initial conditions. With R = 3.25, populations will become synchronized no matter what the initial population values are. In this case, with an x_{0a} of 0.3 and an x_{0b} of 0.7, Population A and B become synchronized after 2 months. Both populations settle into a period-2 cycle at 16 months. The population values then oscillate between 100 rabbits and just over 160 rabbits.

As with any model, the logistic map works based on certain assumptions, and it is only accurate as long as these assumptions hold true. One assumption we make is that the population has a constant carrying capacity. Another is that the birth and death rates of the population, and therefore R, remain constant. Both of these assumptions completely ignore changes in the environment that can affect population dynamics. The environment includes things like weather as well as interactions with other species. This model assumes that the species in question exists in a static environment and that there are no changing cross-species interactions affecting population dynamics. A third assumption is that the initial population is measured accurately to infinite precision. Depending on the R value of the system, if there is even a very small error in the measurement of the initial population the entire model of the system could be completely inaccurate.

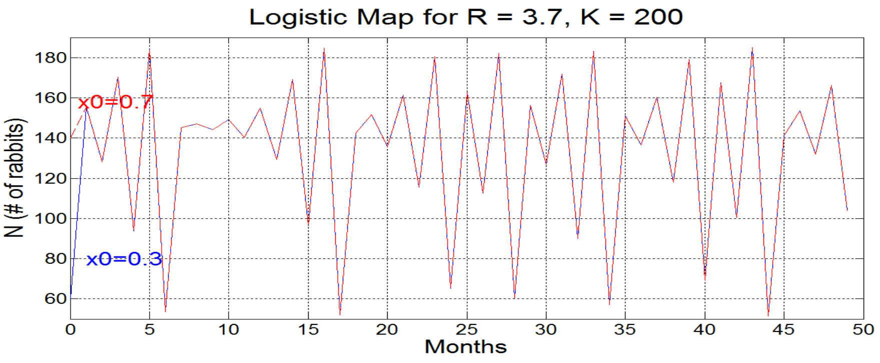

Figure 2: Logistic Map for R=3.7 and K=200

Figure 2 shows logistic maps with R = 3.7. These systems have infinite periods, so although there are visible spots of ergodicity and embedded stability, these populations are chaotic. Normally chaotic systems are very sensitive to even minute differences in initial conditions. With x_{0a} of 0.3 and x_{0b} of 0.7, it seems like these two systems should never become synchronized, but clearly they do. This is because the logistic equation is of the form $x_{i+1} = Rx_i(1 - x_i)$. If x_0 equals 0.3, then $1 - x_0$ equals 0.7. Since x_{0a} and x_{0b} are reciprocals of each other, they become synchronized. In real populations, it is unlikely that chaotic systems could become synchronized in this way because they would have to have reciprocal initial population values calculated with infinite precision. If they differ even slightly from being the reciprocals of each other, they will quickly diverge.

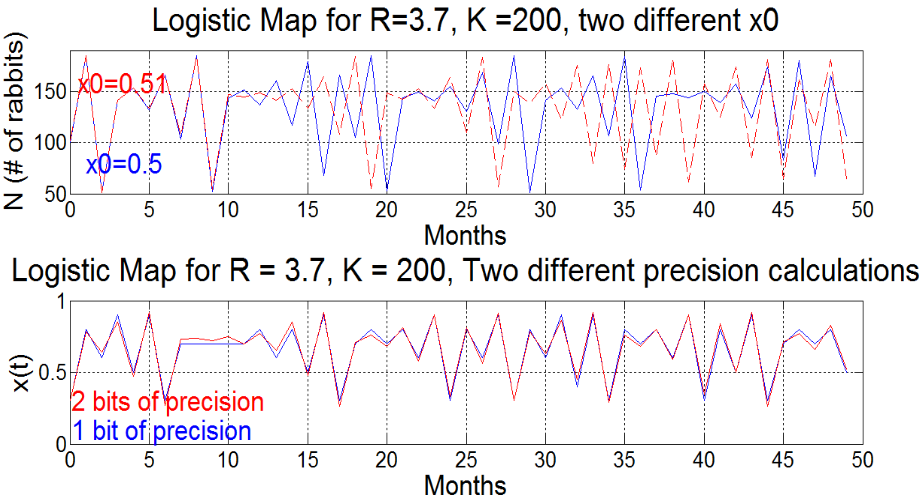

Figure 3: Logistic Maps for R=3.7, K =200

Figure 3a illustrates sensitive dependence on initial conditions. Population A and Population B both have R = 3.7 and a K = 200. However, Population A, illustrated by a blue line, has an x_{0a} of 0.5, while Population B, illustrated by the red line, has an x_{0b} of .51. Although these initial population values are very close, these chaotic systems diverge at t=10 because of this minor difference.

Figure 3b illustrates the sensitivity of chaotic systems to differences in precision. Two systems are graphed, both with an R value of 3.7, a K value of 200, and an x_0 value of 0.3. However, Population A, illustrated by the blue line, rounds to one decimal place at every step, and Population B, illustrated by the red line, rounds to two. Although these two populations have similar dynamics initially, they diverge as time passes. In other words, small differences in population size amplify over time in chaotic systems.

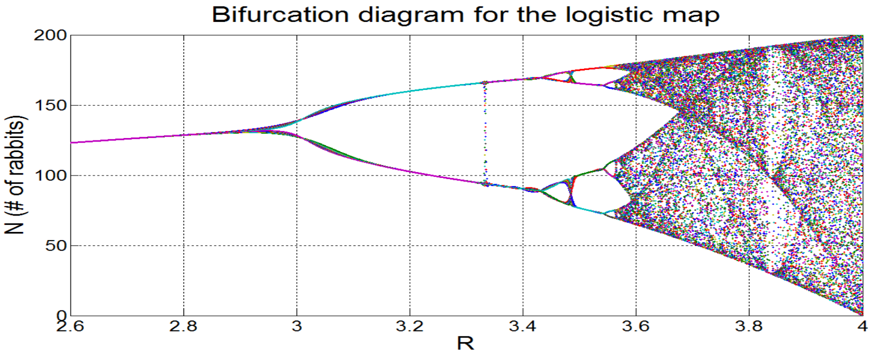

Figure 4: Bifurcation Diagram for R = 2.6 to 4

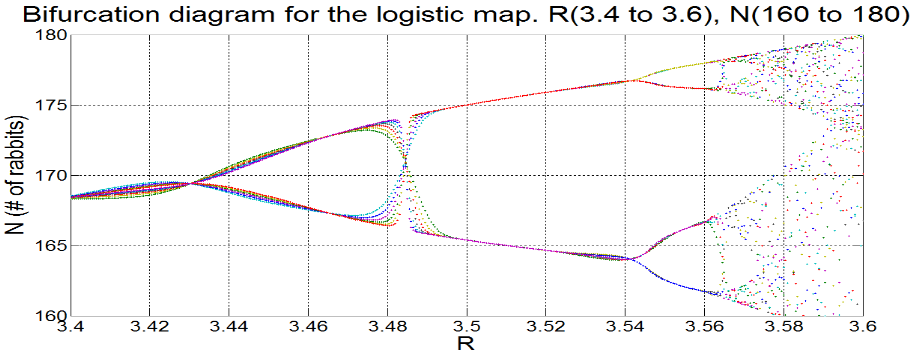

Figure 5: Bifurcation diagram for R = 3.4 to 3.6

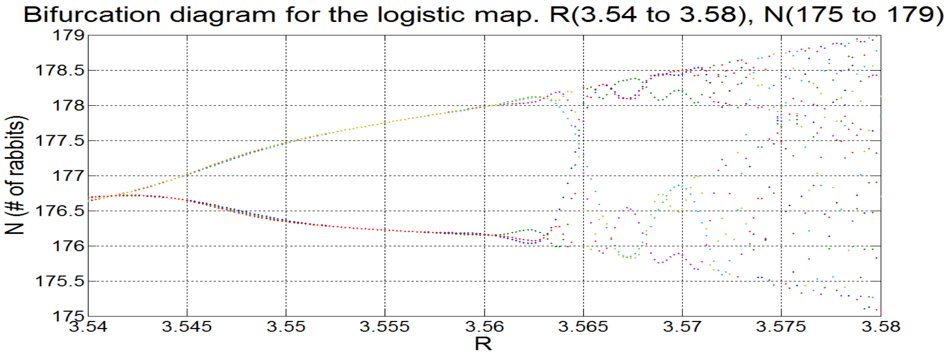

Figure 6: Bifurcation diagram for R = 3.54 to 3.58

Figures 4 and 5 show the transition to chaos in the logistic map. Bifurcations in the period happen repeatedly as R increases. Eventually the period goes to infinity, which is called chaos. A comparison of Figures 4, 5, and 6 illustrates the fractal nature of the bifurcation diagram. This self-similarity hints that there may be a constant that defines how the rate of bifurcation changes over time. This fact was verified by Mitchell Feigenbaum, who found this rate of change in the rate of bifurcation to be ≈ 4.6692 [1] . He calculated this value using the formula $d_k = (a_k - a_{k-1})/(a_{k+1} - a_k)$, taking the limit as k goes to ∞, where d_k represents the rate at which the next bifurcation will occur relative to the previous bifurcation[1]. This rate turns out to hold for all unimodel maps[4]. This means that for any chaotic system that is governed by this kind of map, by measuring when the first bifurcations occur you can get an estimate of the point when the system will become chaotic, which is quite powerful[1].

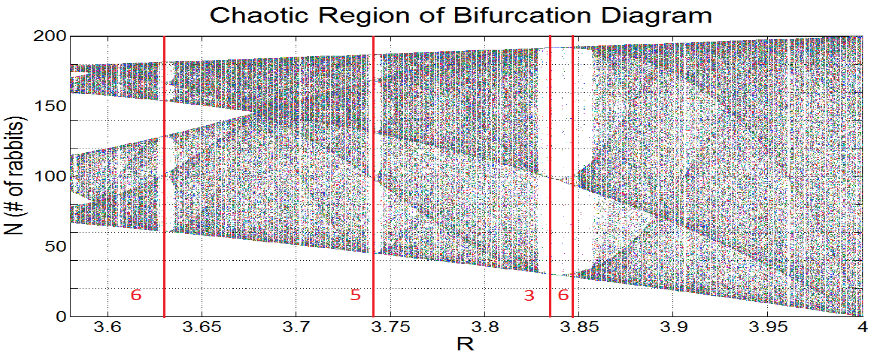

Figure 7: Bifurcation diagram highlighting periods of stability in chaos

Figure 7 highlights periods of embedded stability that occur within chaotic dynamics. This means that for certain values of R which fall within the chaotic range, a system will actually have unstable periodicity. Mathematically, this happens anywhere that the function $f^m(x)$ intersects the identity line $y = x$[1]. This embedded stability is a fundamental propery of chaotic systems.

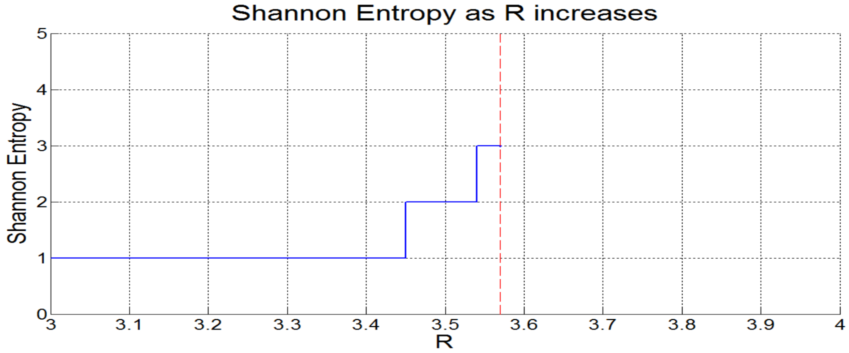

Figure 8: Shannon Entropy as R increases

Shannon Entropy gives an informative measure of how much uncertainty or randomness there is in a dataset. Entropy is defined by $H_x = -\sum p_x log(p_x)$, where p_x equals the probability of being in state x. As R increases, the period increases as well. By plugging in values corresponding to number of bifurcations in the period per step, we can clearly see how the entropy changes as a system moves towards chaos. In chaos, a system is defined to have an infinite period. As Figure 8 shows, Shannon's

Entropy goes to infinity as we move into chaotic dynamics.

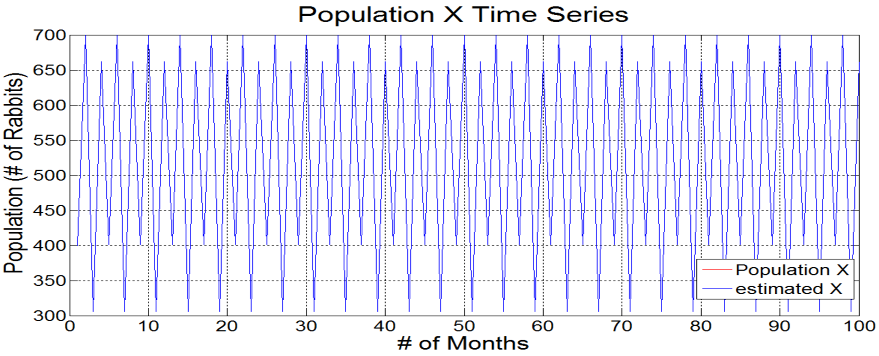

Figure 9: Population X Time Series

For Population X, R = 3.5, K = 800, and x_0 = 400. Population X has a period of 4 and was clearly generated by a logistic map. As shown in Figure 9, our estimated parameters match this population exactly. We believe we are able to accurately predict Population X at time step 105 and time step 120. At time step 105, Population X is 401, and at time step 120, Population X is 662.

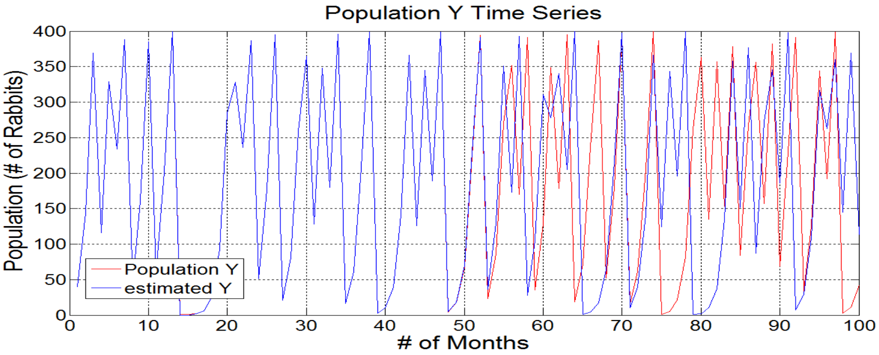

Figure 10: Population Y Time Series

For Population Y, R = 4, K = 400, and x_0 = 40. Population Y is a chaotic system, so small differences in precision can cause differences in behavior over time. Figure 10 shows this divergence; at first our estimated parameters track Population Y accurately, but at around t = 53 they diverge. As a result, we are unable to predict Population Y at future time steps with high confidence. Our estimate of Population Y at time step 105 is 240, and our estimate of Population Y at time step 120 is 392.

At first glance, Population Z looks like it has a period of 4. However, upon closer inspection it becomes clear that there are no constant threshold values for each set of peaks.

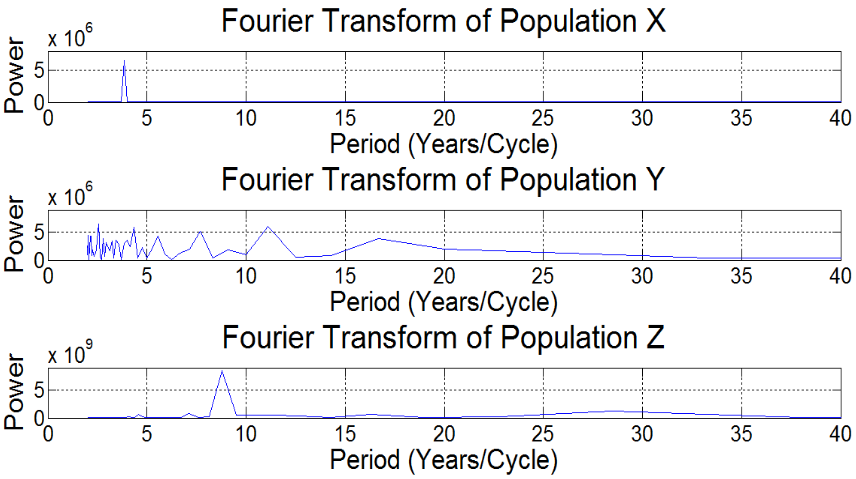

Figure 11: Fourier Transfer for population X, Y and Z

To better judge the period, we performed a Discrete Fourier transform of the data. As shown in Figure 11c, there is a clear peak at around 9.4. This indicates that the period of the graph is 9.4.

There are several indications that this time series could not have been generated by a logistic map. The first is that the highest peak increases in height each time, so there is no clear carrying capacity. A second is that, as stated before, although the data looks periodic, it is not. A third is that it is clearly not chaotic either. Figure 11 compares the Fourier transforms of Populations X, Y, and Z. Population X has only 1 peak at 4; it is periodic. Population Y has an apparently random distribution; it is chaotic. While Population Z does have a peak at 9.4, it also has smaller peaks at other values. This is neither a random chaotic distribution nor a purely periodic function. This means that this series is neither periodic nor chaotic and therefore could not have been generated by a logistic map. Furthermore, even if we ignore the noise and assume it has a period of 9.4 and it is truly periodic, this is still not an allowable period for the logistic map. The period of any function generated by a logistic map must be a power of 2.

5 Conclusion

Our results indicate that as R increases from 3 to 4, population systems that are governed by the logistic map go from periodicity to chaos. Shannon's entropy also increases as R increases, with the uncertainty of the dataset going to infinity as the system becomes chaotic.

Our results illustrate the positives and negatives of using a logistic map to predict population dynamics. For a population in a neutral environment with non-chaotic dynamics, the logistic map has strong predictive power. However, if a system has chaotic dynamics, the logistic map fails long-term. While it may initially make accurate predictions, over time even very small differences between prediction values and actual population values can cause complete divergence in population estimates. Thus, while the logistic map can shed light on chaotic system dynamics, it is not accurate as a long-term predictor of population values for chaotic systems. For a population whose dynamics are neither 2^n periodic nor chaotic, as in time series Z, the logistic map has no predictive power at all.

One important caveat of using the logistic map as a model of population dynamics is that it completely ignores any changes in the environment. When the model is set up, a population has a certain birth rate, death rate, and carrying capacity, and these values are then set for the entire model. In real environments, these parameters are constantly shifting, so this is a major shortcoming of the logistic map. However, in an environment that has fairly stable species relations and weather patterns, it can be a useful tool in understanding how populations change over time.

References

[1] G. W. Flake. *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation.* A Bradford Book, 2000.

[2] Matlab. Decimal precision. `http://www.mathworks.com/matlabcentral/newsreader/view_thread/169196`, Last accessed 01/31/2013.

[3] Matlab. Using fft. `http://www.mathworks.com/products/matlab/examples.html?file=/products/demos/shipping/matlab/sunspots.html`, Last accessed 01/31/2013.

[4] M. Mitchel. *Complexity: A Guided Tour.* Oxford University Press, 2011.