Course: CS30A1570 Complex Systems

Assignment 2: Dynamics and Chaos

Completion Date: 18 March 2024

Submitted by: Turzo Ahsan Sami

Abstract
This report explores the dynamics of population growth using both mathematical models and agent-based simulations. Task 1 investigates the behavior of the logistic map, a mathematical representation of population growth. Task 2 analyzes a population growth model based on an iterative equation. Task 3 builds and analyzes an agent-based model in NetLogo that simulates logistic population growth. The results demonstrate that the agent-based model exhibits qualitatively similar dynamics to the logistic population growth model, despite their different approaches.

Introduction
To understand population growth of an ecological system, the logistic model is used. It is a widely used mathematical model to capture the dynamics of population growth. It considers both intrinsic growth rate and limitations of the environment. This report investigates population growth through three tasks:

1. Exploring the behavior of the logistic map, a mathematical representation of population growth.
2. Analyzing a population growth model based on an iterative equation.
3. Building and analyzing an agent-based model in NetLogo that simulates logistic population growth.

Research Question
Does an agent-based model of population growth exhibit qualitatively similar dynamics to the logistic population growth model, despite their different approaches (equation-based vs. agent-based)?

Task 1: Logistic Map Analysis

Task 1 explored the behavior of the logistic map, revealing that the parameter R (growth rate) must be between 0 and 4 for the map to be a valid model. When R = 1, the population converges to 0.

Exercise 1.1: In the logistic map, is always between 0 and 4. Why is this the case? What happens if is set to be greater than 4?

Answer: The logistic map equation is x_{t+1} = R * x_t * (1- x_t), the parameter R represents the growth rate. From the *LogisticMap.nlogo* simulation, after varying the value of R between 0 and 4 the following observations are found:

i. R is set at 0: The population quickly becomes 0.
ii. R is set between 0 and 1: The population gradually becomes 0.
iii. R is set at 1: The population gradually becomes 0.
iv. R is set between 1 and 2: The population will stabilize at a single, non-zero value.
v. R is set at 2: The population will stabilize at a single, non-zero value.
vi. R is set between 2 and 3: The population will oscillate between two values.

vii. R is set at 3: The population will oscillate between two values.
viii. R is between 3 and 4: The population will oscillate between two values.
ix. R is set at 4: The population quickly becomes 0.

For the logistic map to remain a valid and useful model, (R) is kept within the range of 0 to 4.

Exercise 1.2: Consider the logistic map: x_{t+1} = R * x_t * (1- x_t) If R = 1, what fixed point does always go to?

Answer: From the *LogisticMap.nlogo* simulation it is evident that when R = 1, the population eventually becomes 0 (Appendix: Image 1).

Exercise 1.3: Modify the *SimplePopulationGrowth.nlogo* to change shape of individuals and the patches color.

Answer: Modified the model, set patch color as "blue" and set the turtle shape as "face happy" (Appendix: Image 2, Attachment: 1).

Exercise 1.4: Modify *SensitiveDependence.nlogo* to change the background color and add labels to the dots.

Answer: Modified the model, set background color as "green" and set label of the turtles as their coordinates. (Appendix: Image 3, Attachment: 2).

Task 2: Iterative Population Growth Model

Task 2 analyzed a population growth model based on an iterative equation. It demonstrated that the number of generations required for the population to double depends on the initial population size and the growth rate.

Study question:
Suppose that a population of organisms grows according to the following rule n_{t+1} = 1.1 n_t where, n_t is the population size at generation t and n_{t+1} is the population size at the next generation. If the population starts at n_0 = 100 individuals, how many generations will it take until the population has more than doubled (that is, is greater than 200)? (Assume that if n_t is not an integer, the actual population size is rounded off).

Answer:
Given the population growth function n_{t+1} = 1.1 n_t, where n_t, is the population at generation t, we can find the number of generations required for the population to reach more than 200 starting from an initial population of n_0 = 100. We can solve this iteratively:

i. At generation t = 0, population n_0 = 100
ii. At generation t = 1, population n_1 = 1.1 * n_0 = 1.1 * 100 = 110
iii. At generation t = 2, population n_2 = 1.1 * n_1 = 1.1 * 110 = 121
iv. At generation t = 3, population n_3 = 1.1 * n_2 = 1.1 * 121 = 133.1
v. At generation t = 4, population n_4 = 1.1 * n_3 = 1.1 * 133 = 146.4

vi. At generation t = 5, population n_5 = 1.1 * n_4 = 1.1 * 146 = 160
vii. At generation t = 6, population n_6 = 1.1 * n_5 = 1.1 * 160 = 176
viii. At generation t = 7, population n_7 = 1.1 * n_6 = 1.1 * 176 = 193.6
ix. At generation t = 8, population n_8 = 1.1 * n_7 = 1.1 * 193 = 212.3

In this scenario, it takes 8 generations for the population to reach 212 individuals (rounded at each step).

Exercise 2.1: Prove algebraically that, x_{t+1} = 2(x_t -$x_t{}^2$) has fixed point 0.5

Answer: A fixed point of a function $f(x)$ is a point x such that $f(\,x_0)$ = $x/_0$ (Reference 1)
By substituting x = 0.5, the given equation gives us, x_{t+1} = 2*(0.5 – (0.5) ^2) = 0.5
Since $f(x) = 0.5\ and\ x = 0.5$ we can confirm that 0.5 is indeed a fixed point of the function, as substituting it into the function returns the same value.

Exercise 2.2: Modify *SensitiveDependence.nlogo* as follows: add a third initial condition, x0''.

Answer: Modified the model, set background color as "green" and set label of the turtles as their coordinates. (Appendix: Image 4, Attachment: 3).

Task 3: Agent-Based Model in NetLogo

Task 3 built an agent-based model in NetLogo that simulates logistic population growth. The model incorporated key features like:

- Reproduction rate for turtles (similar to intrinsic growth rate)
- Carrying capacity to limit population growth
- Energy level for each turtle representing resource limitations
- Stochastic reproduction for added realism
- Look-around procedure to decrease energy and increase age
- Expire procedure for turtle death due to low energy or old age

The model exhibited population growth dynamics qualitatively similar to the logistic population growth model. Initially, the population grew rapidly due to low population and high energy levels. As the population increased, limited resources and energy led to slower growth and eventually a stable equilibrium around the carrying capacity.

Answer: An agent-based logistic population growth model is created with Netlogo (Appendix: Image 5, Attachment: 4). The model exhibits population growth dynamics that are qualitatively similar to the Logistic population growth model (equation based) due to several key factors:

- The turtles have a reproduction-rate that determines the number of offspring hatched per step. This rate is similar to the intrinsic growth rate (r) in the equation-based model.
- The model includes a carrying-capacity variable, which acts as an environmental limit to population growth. This is similar to the carrying capacity (K) in the equation-based model, where the population cannot grow indefinitely.
- The model uses an "energy-level" variable for each turtle. This shows that there are limited resources in the environment. Turtles lose energy over time. Reproduction requires a minimum

energy level. The reproduction is also limited by the reproduction-rate and the carrying capacity. This ensures that the population growth rate decreases as the population size approaches the carrying capacity.

- The model uses a random chance (random-float 1.0 < birthrate) to determine if a turtle reproduces, adding a stochastic element.
- The model's look-around procedure decreases each turtle's energy-level and increases the age at each tick. In the "expire" procedure, turtles die if their energy-level drops to 0 or their age reaches a certain limit set by max-age-slider. This feedback mechanism helps regulate the population size. This is similar to how the equation-based model's growth rate decreases as the population size nears the carrying capacity.

From the model simulation it is observed (Attachment: 5) that, in the beginning, low population and high energy levels lead to rapid growth (exponential phase). As the population increases and energy levels drop, reproduction is limited, and mortality is increasing (slowing growth phase). Eventually, the population reaches a dynamic equilibrium near the carrying capacity. Here, births and deaths balance each other (stable phase). This behavior is similar to the logistic population growth model, where the growth rate slows down as resources become limited (carrying capacity is approached).

Conclusion

The results demonstrate that the agent-based model, despite its different approach compared to the logistic equation, captures the essence of logistic population growth. Both models show how limited resources and environmental constraints regulate population size. This highlights the power of agent-based modeling for simulating complex ecological dynamics.

Appendices:

Image 1: The LogisticMap.nlogo simulation for R = 1

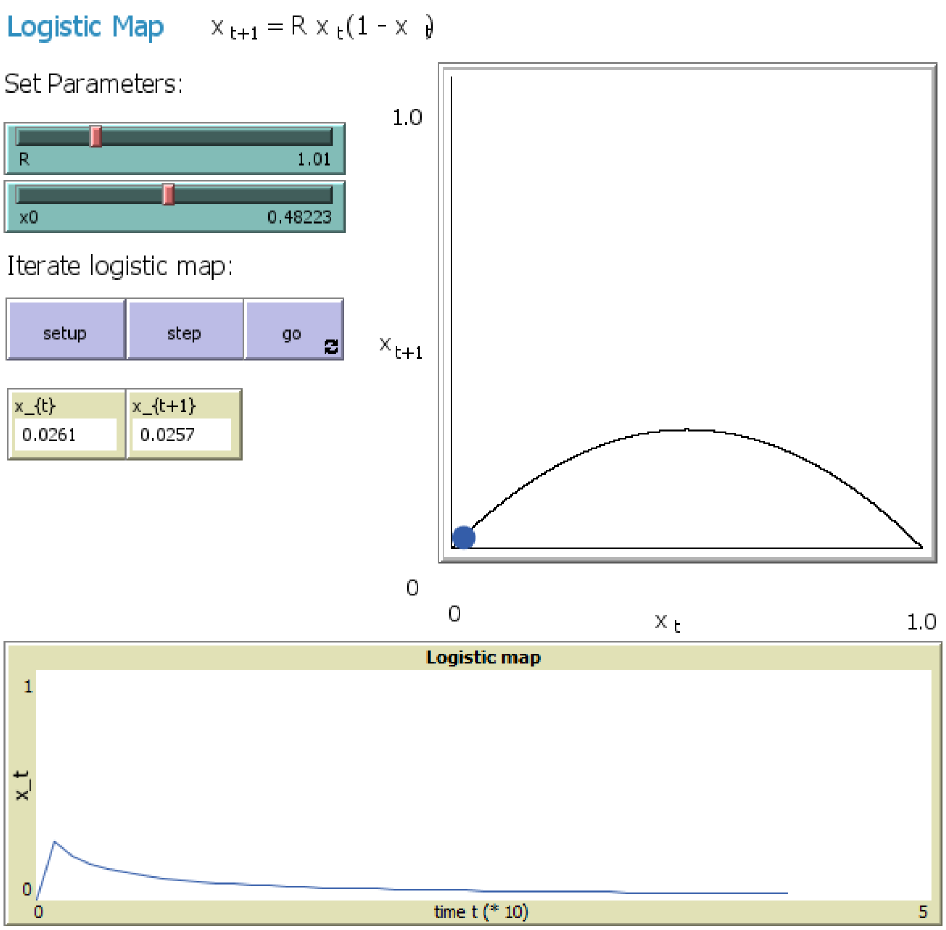


Image 2: Modified SimplePopulationGrowth.nlogo simulation

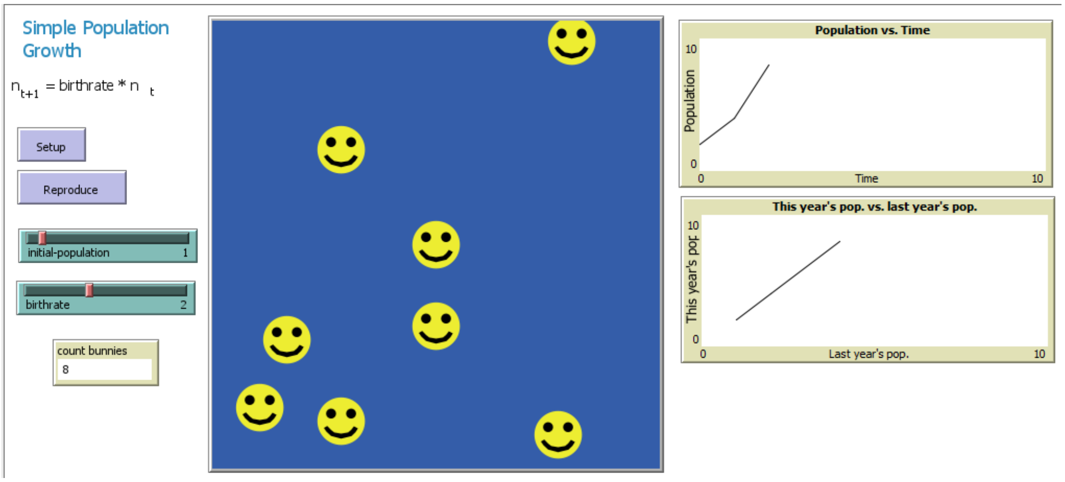


Image 3: Modified SensitiveDependence.nlogo simulation

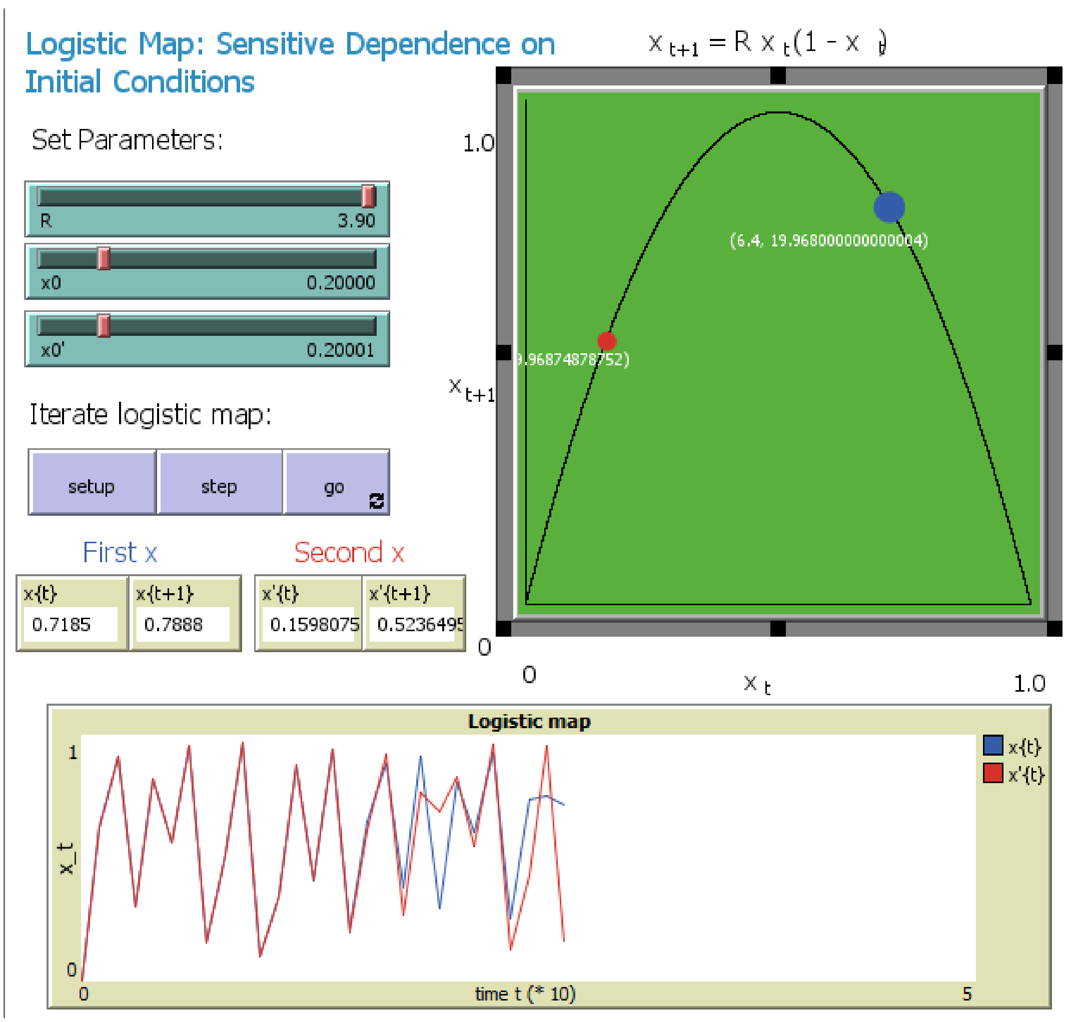


Image 4: Modified SensitiveDependence.nlogo simulation

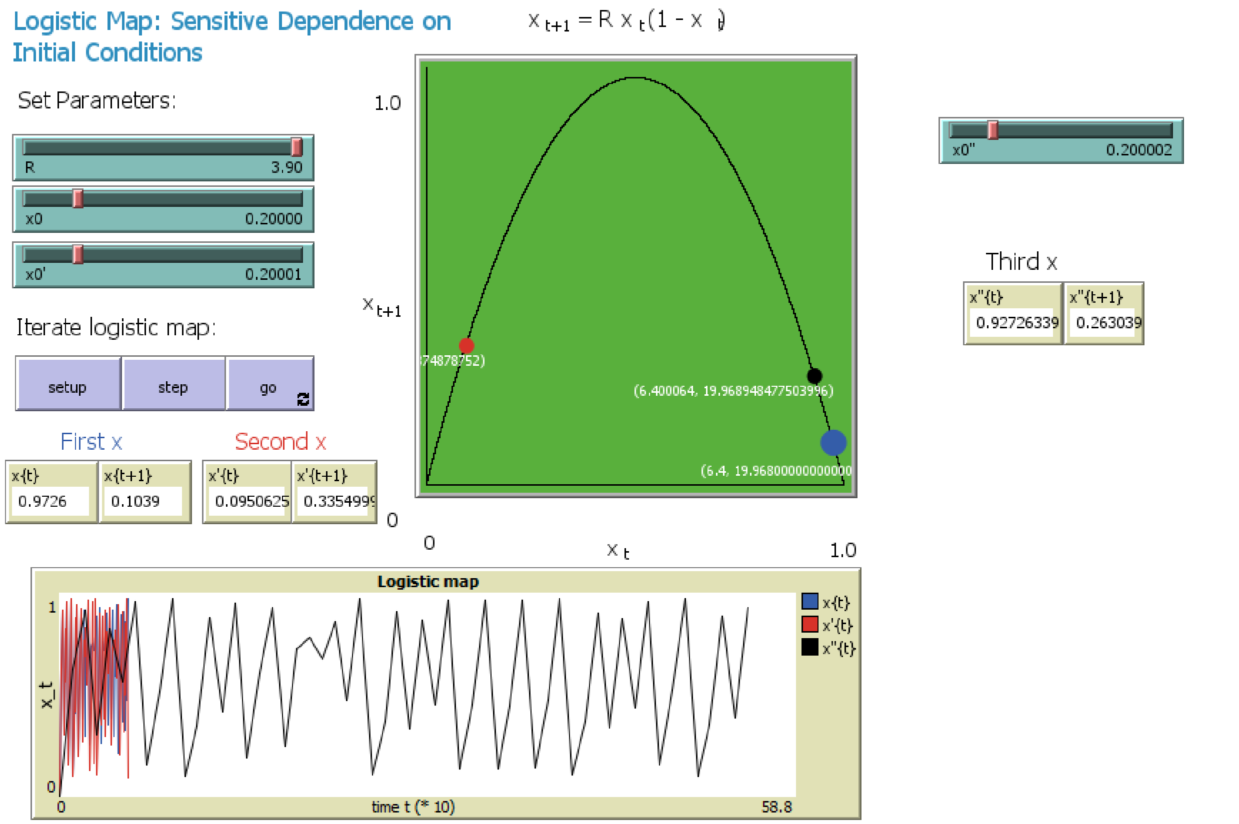
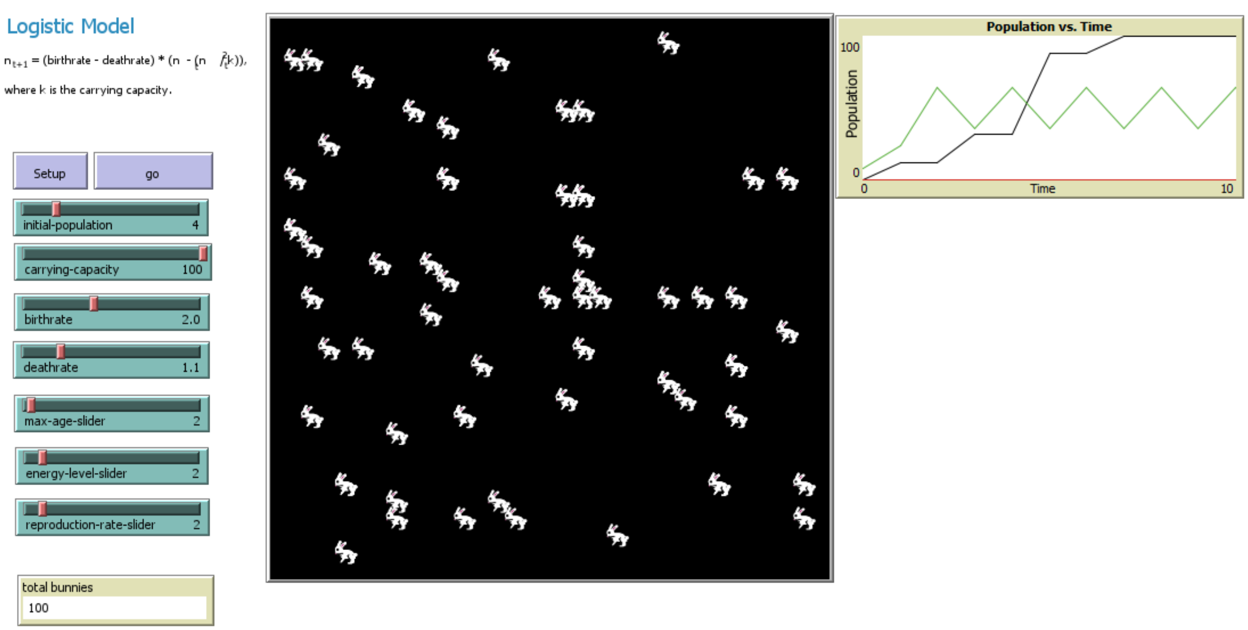


Image 5: Agent-based Population Growth Model



Attachments:

1. complex-system-code/CS_Assignment_2/SimplePopulationGrowth_v_6.1.1_modified.nlogo at main · ahsan-sami-turzo/complex-system-code (github.com)
2. complex-system-code/CS_Assignment_2/SensitiveDependence_v_6.1.1_modified.nlogo at main · ahsan-sami-turzo/complex-system-code (github.com)
3. complex-system-code/CS_Assignment_2/SensitiveDependence_v_6.1.1_modified_2.nlogo at main · ahsan-sami-turzo/complex-system-code (github.com)
4. complex-system-code/CS_Assignment_2/LogisticModel_v_6.1.1_agent_based.nlogo at main · ahsan-sami-turzo/complex-system-code (github.com)
5. complex-system-code/CS_Assignment_2/LogisticMap_v_6.1.1_agent_based.xlsx at main · ahsan-sami-turzo/complex-system-code (github.com)

References:

1. https://mathworld.wolfram.com/FixedPoint.html