Oregon State University
College of Engineering
ENGR 103

Project Report : Population Growth Models

Alexander Vogs

Part 1: Population Growth Simulation - Program Design

Program Statement:

The purpose of this program is to simulate population growth over time using two different models: geometric growth and logistic growth. The user will input values for birth rate, death rate, carrying capacity, and starting population. The program will simulate 100 time steps of population change and display results every 10 steps.

The program must:

1. Read birth rate, death rate, carrying capacity, and initial population from the user.
2. Validate all inputs.
3. Use the geometric and logistic equations to model population growth.
4. Print population values at every 10 time steps for both models.

Assumptions to be made:

1. Birth and death rates are floats between 0 and 1.
2. Carrying capacity and initial population are positive integers.
3. The population can't go negative.
4. Growth equations are applied repeatedly for 100 steps.

Inputs:

1. Birth rate (float, 0–1)
2. Death rate (float, 0–1)
3. Carrying capacity (positive integer)
4. Initial population (positive integer)

Outputs:

1. Population every 10 time steps for geometric model
2. Population every 10 time steps for logistic model

Sequence of Steps:

1. Ask for birth rate, death rate, carrying capacity, and starting population
2. Validate each input:
   Birth rate and death rate must be between 0 and 1
   Carrying capacity and initial population must be positive integers
3. Calculate growth rate: `r = birth rate - death rate`
4. For geometric growth: Use: `Nt+1 = Nt + r * Nt`
5. For logistic growth: Use: `Nt+1 = Nt + r * (1 - Nt / K) * Nt`

6. Loop through 100 time steps for both models
7. Print population every 10 steps

Pseudocode:

```
Prompt for birth rate
Prompt for death rate
Prompt for carrying capacity
Prompt for initial population

If any input is out of bounds or not a number:
   Display error and exit

Calculate r = birth rate - death rate

Set population = initial population

For t from 1 to 100:
   Calculate next population using geometric model
   Save value
   Update current population

Set population = initial population

For t from 1 to 100:
   Calculate next population using logistic model
   Save value
   Update current population

Every 10 steps:
   Print population values for both models
```

Test Plan

To make sure the program works correctly, I created a test plan with good, bad, and edge input cases.

Case Type	Input (birth, death, K, N_0)	Expected Output
Good Case 1	0.2, 0.1, 1000, 21	Should show population growing gradually under both models
Good Case 2	0.3, 0.2, 800, 100	Slight growth, should plateau under logistic
Edge Case 1	0.0, 0.0, 1000, 50	Population should stay constant
Edge Case 2	0.5, 0.5, 500, 200	No change in population
Bad Case 1	-0.1, 0.2, 1000, 100	Error: Birth rate out of bounds
Bad Case 2	0.3, "abc", 1000, 100	Error: Invalid input
Bad Case 3	0.2, 0.1, -100, 50	Error: Carrying capacity must be positive

How will I analyze correctness?
I'll check the math manually for a few steps and compare it with what the program outputs. I'll also use the test plan to confirm expected behavior in each case.

Part 2: Population Growth Simulation – Python Implementation

I wrote a Python program to simulate population growth using both the geometric and logistic growth models. The program takes in values for birth rate, death rate, carrying capacity, and initial population. It validates the inputs, then calculates and displays the population size at every 10 time steps for 100 total steps using both models.

Here is the final code I used: GitHub Download

Revisiting Looking Back

To test the program, I used the same cases from Part 1, including the whooping crane example with a birth rate of 0.2, death rate of 0.1, carrying capacity of 1000, and initial population of 21.

Test Case: Whooping Crane Recovery

Inputs:

- Birth rate: 0.2
- Death rate: 0.1
- Carrying capacity: 1000
- Initial population: 21

Expected Outputs:

1. Geometric growth should keep increasing rapidly.

2. Logistic growth should slow as it approaches 1000.

Actual Output (every 10 time steps):

Time Step	Geometric Model	Logistic Model
10	54.47	52.86
20	141.28	127.27
30	366.44	277.91
40	950.44	508.46
50	2465.21	740.13
60	6394.11	888.95
70	16584.69	957.93
80	43016.40	984.87
90	111573.47	994.67
100	289392.86	998.13

Revisit Test Plan Results

- **Bad Inputs:** Rejected as expected (e.g. negative birth rate, non-number death rate).
- **Edge Cases:** Flat population growth or zero change confirmed the logic worked correctly.
- **Good Inputs:** Matched mathematical expectations.

How many steps to reach ~1000 individuals?

- **Geometric model** hit 1000 individuals *around step 40.*
- **Logistic model** approached 1000 *between step 90–100,* slowing down as it got closer.

How I Analyzed the Results

I used a calculator and spot-checked the math by hand to make sure the calculations matched the formulas:

- Geometric: `Nt+1 = Nt + r * Nt`
- Logistic: `Nt+1 = Nt + r * (1 - Nt / K) * Nt`

I confirmed the loop ran for 100 steps and printed the right values at every 10-step interval. I also checked against the expected behavior described in the assignment.