

# Final Project: The Euler-Lotka Equation

EN 250: Quantitative Biology

James Lepore

14 December 2021

# Table of Contents

<b>Part 1: General Component</b>	<b>3</b>
<b>1 History of the Model</b>	<b>3</b>
1.1 Assumptions	4
1.2 Objectives	4
1.3 Background	4
1.3.1 Leonhard Euler	5
1.3.2 Alfred J. Lotka	5
<b>2 Applications of the Model</b>	<b>6</b>
2.1 Demography	6
2.2 Conservation Ecology	6
<b>3 Mathematical Properties of the Model</b>	<b>7</b>
3.1 Solvability by Software	7
3.2 Model Outcomes	8
<b>4 Example Research</b>	<b>9</b>
<b>5 Optimal Experimental Design</b>	<b>9</b>
<b>Part 2: Original Research Component</b>	<b>11</b>
<b>6 Introduction</b>	<b>11</b>
<b>7 Materials and Methods</b>	<b>11</b>
7.1 Materials	11
7.2 Methods	12
<b>8 Calculations</b>	<b>12</b>
<b>9 Results</b>	<b>16</b>
<b>11 Conclusions</b>	<b>18</b>
<b>Works Cited</b>	<b>19</b>

# Part 1: General Component

## 1 History of the Model

The history behind population growth models is rich and expansive. When it comes to population growth models, many first instincts tend to focus on the impact of the environment on a population. If there are no limiting factors in the environment, the simplest way to think about population growth is through an exponential model. There are more organisms to reproduce over time, and therefore the population grows more quickly over time. Figure 1 demonstrates this simple population growth model through *Paramecia* (Vandermeer 1).

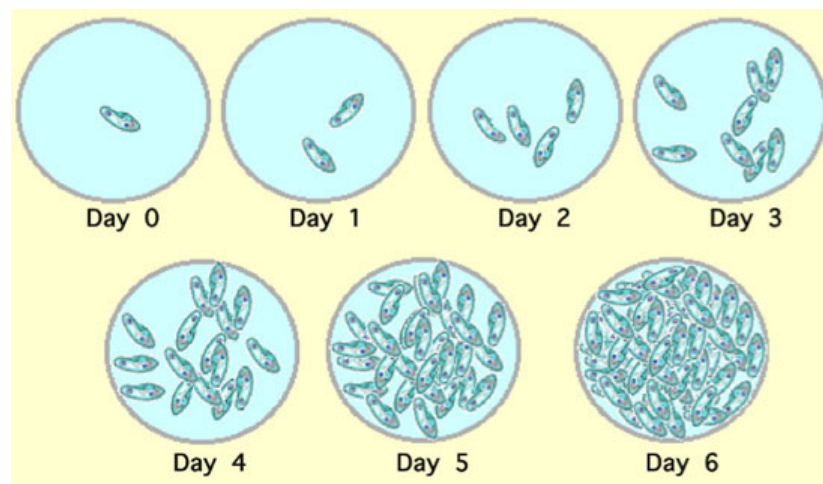


Figure 1: Simple exponential growth for a *Paramecium* population

Even when discussing limiting factors, the discourse tends to focus on external factors, such as space, food, and water. The Euler-Lotka equation, however, focuses particularly on the age-structured population growth of animals. This equation allows for an estimation of population growth based on the age demographic of females in the population; this is an internal factor. Of course, the age demographics may be impacted by external factors, but this is out of the scope of the model. The inputs for the model are raw population data, and the model is not concerned with what external factors caused those numbers to come about.

The Euler-Lotka equation is perhaps the most important equation in the study of age-structured population growth. In this Section 1, the assumptions, objectives, and background of the model and its creators will be discussed in detail.

## 1.1 Assumptions

There are several necessary assumptions to simplify the model. The first assumption is that each individual's chance of survival and reproduction depends only on its age. As mentioned in the introduction to Section 1, external factors are not considered in the calculations of the model, only raw age and population data. A related assumption is that there are no effects of population density. This specifically addresses the potential limiting factor of space. If a population grows to be rather crowded, this will not deter further reproduction according to the model.

The model additionally assumes that there is no shortage of males. Females in the population can carry one pregnancy at a time, but males can impregnate as many females as they can within any given time period. Due to this natural occurrence of females being more limited in their ability to reproduce, the model does not bother counting the number of males in a population; rather, it is solely based on the age demographic of females in the population and female births.

A post-breeding census is also assumed to be used for the count. For simplicity of the model, all births occur at around the same time immediately before the population is censused, and those newborn individuals are counted as age 0 (Okuyama). Despite these assumptions, the model nonetheless serves as an accurate and trusted estimation for population growth.

## 1.2 Objectives

The objective of the Euler-Lotka equation is to give an estimation of how a population is growing. This is an important indicator of the viability of a population. With this information gained, a population viability analysis can be performed in order to determine the probability that a population will go extinct within a given number of years. Clearly, this sort of information is useful for several reasons and can be used by conservationists, ecologists, and biologists, among others (Amarasekare).

## 1.3 Background

As the name implies, the Euler-Lotka equation is attributed to two originators. Swiss mathematician Leonhard Euler derived a special form of the equation back in 1760, and Alfred J. Lotka built upon the work of Euler to derive a more continuous version of the equation in 1911.

### 1.3.1 Leonhard Euler

Euler was the most prolific mathematician of his time. His work has had several implications in a vast array of topics within mathematics and mechanics. He also contributed to work in demography, which is a less traditionally known aspect of his wide range within math and science. In 1748, he wrote *Introduction to Analysis of the Infinite*, in which he used exponential growth to describe a population in four examples. In 1760, he published an article combining this exponential growth with an age structure for the population; this work was pivotal to the field of demography. The next year, he also helped Süssmilch with the second edition of his treatise on demography, in which he worked out a model; however, he did not publish his detailed analysis (Bacaër).

### 1.3.2 Alfred J. Lotka

Lotka was a pioneer in the field of biophysics, famous for his work in population dynamics and his proposal of the predator-prey model. He also built off of the work of Euler by developing a continuous model of population dynamics.

For this model, the number of births per time,  $B(t)$ , is a function of the fraction of individuals surviving to age  $a$ ,  $l(a)$ , and the birth rate per capita for mothers of age  $a$ ,  $b(a)$ . This yields the following equation:

$$B(t) = \int_0^t B(t - a) l(a) b(a) da$$

Since Euler determined births to follow exponential growth, Lotka substituted the general form for  $B(t)$ .

$$Qe^{rt} = \int_0^t Qe^{r(t-a)} l(a) b(a) da$$

Writing this as a discrete sum results in:

$$1 = \sum_{a=\alpha}^{\beta} e^{-ra} l(a) b(a)$$

Finally, defining the discrete growth rate  $\lambda = e^r$  yields the common form of the Euler-Lotka equation.  $r = \ln \lambda$  is known as Lotka's intrinsic rate of natural increase.

$$1 = \sum_{a=1}^{\omega} \lambda^{-a} l(a) b(a)$$

(“Euler–Lotka Equation.”)

## 2 Applications of the Model

The most widely used application of the Euler-Lotka equation is in the field of demography. Demography is the statistical study of human populations. Demographers use census data, surveys, and statistical models to analyze the size, movement, and structure of populations. There is also large interest in applying the Euler-Lotka equation in the field of Conservation Ecology. In order to protect species that could potentially go extinct, scientists need a way to determine the growth rate of the population.

### 2.1 Demography

In terms of demography of the human population, governments would be interested in the results of an analysis using the Euler-Lotka equation. You need people to sustain the population; this is obvious. If the population is shrinking, this might be cause for concern, and the government may be able to implement some sort of policy to increase the rate of birth. The alternate scenario is a population that is growing far too quickly for the resources of a country to sustain. China’s previously enforced One Child Policy, for example, was placed into effect to control the size of the country’s rapidly growing population. This brought about its own set of problems in terms of the demographics of the population, but the policy and its implications are irrelevant to the overall message that a demographic analysis is certainly of interest to governments around the world, and the Euler-Lotka equation is the premier equation for calculating age-structured population growth (National Geographic).

### 2.2 Conservation Ecology

There is also a very significant application in the field of conservation ecology. This field is a discipline that is emerging rapidly as a result of the accelerating deterioration of natural systems and the worldwide epidemic of species extinctions. Its goal is to find ways to conserve species, habitats, landscapes, and ecosystems as quickly, as efficiently, and as economically as possible. In order to determine what population to protect, there needs to be some sort of warning or indication that a population is declining in the first place. The Euler-Lotka equation can serve as an excellent model to analyze the decline of a population attributed to age distribution of that population. Performing an analysis

of the model outcome will provide that aforementioned warning or indication so that conservationists can start focusing more attention and resources towards the populations that need them the most (Malepeai).

### 3 Mathematical Properties of the Model

A thorough mathematical analysis of the Euler-Lotka equation demonstrates some interesting properties that it possesses.

The Leslie matrix is another very well known way to describe the growth of populations and their projected age distribution, in which a population is closed to migration, growing in an unlimited environment, and where only one sex, usually the female, is considered. Given the similarities between this model and the Euler-Lotka model, it should come as no surprise that the Euler-Lotka equation is the characteristic polynomial of the Leslie matrix. In linear algebra, the characteristic polynomial of a square matrix is a polynomial which is invariant under matrix similarity and has the eigenvalues as roots. So, the solutions of the Euler-Lotka equation will give us information about the eigenvalues of the Leslie Matrix, which can provide more information about population stability.

Another interesting property occurs when Lotka's intrinsic rate of natural increase ( $\lambda$ ) is set to 1. In this case, the discrete formula yields the replacement rate of the population. Replacement fertility is the total fertility rate at which women give birth to enough babies to sustain population levels. According to the UN Population Division, a total fertility rate (TFR) of about 2.1 children per woman is considered replacement-level fertility (Euler-Lotka).

#### 3.1 Solvability by Software

The equation itself is a simple summation over the range of viable ages of female reproduction for a particular population. This is very easy to implement in code, making the Euler-Lotka equation solvable by software. In fact, it is much easier to perform the necessary calculations with software than by hand. Using loops or advanced list comprehension, the repeated multiplications and additions are abstracted away, resulting in a very simple function. An example implementation in Python is shown below (Hill).

```

import numpy as np
from scipy.optimize import brentq

# The data, from Table 6 of:
# P. H. Leslie and R. M. Ranson, J. Anim. Ecol. 9, 27 (1940)
x = np.linspace(8, 72, 9)
m = np.array( [0.6504, 2.3939, 2.9727, 2.4662, 1.7043,
               1.0815, 0.6683, 0.4286, 0.3000] )
P = np.array( [0.83349, 0.73132, 0.58809, 0.43343, 0.29277,
               0.18126, 0.10285, 0.05348, 0.02549] )

# Calculate the product sequence f and R0, the ratio between the number of
# female births in successive generations.
f = P * m
R0 = np.sum(f)
if R0 > 1:
    msg = 'R0 > 1: population grows'
else:
    msg = 'Population does not grow'

# The Euler-Lotka equation: we seek the one real root in r
def func(r):
    return np.sum(f * np.exp(-r * x)) - 1

# Bracket the root and solve with scipy.optimize.brentq
a, b = 0, 10
r = brentq(func, a, b)
print('R0 = {:.3f} ({} )'.format(R0, msg))
print('r = {:.5f} (lambda = {:.5f})'.format(r, np.exp(r)))

```

Figure 2: Python program to calculate the growth rate of a population using the Euler-Lotka Equation

### 3.2 Model Outcomes

The standard equation in discrete time is given by the following:

$$1 = \sum_{a=1}^{\omega} \lambda^{-a} l(a) b(a).$$

The 1 on the left-hand side of the equation demonstrates a perfectly balanced population growth, in which there is exactly enough reproduction to keep the current population sustained. However, it is important to note that the result of the expression on the right-hand side of the equation will not always result in 1. To understand the implications of the model, it is necessary to first understand the different outcomes of the calculation.

The result is a summation of positive values, so the final result will always be greater than zero. If the result is between 0 and 1, this indicates that the population is shrinking. A conservationist who performs an analysis using the Euler-Lotka equation would be concerned with a result less than 1. Using the rate of population decline, they can estimate how long a particular population is able to remain alive at that rate. On the other hand, a result above the value of 1 indicates that the population is growing. If this growth reaches startlingly high levels, this may also be a concern. A population may be projected to grow too



quickly for an environment to handle. That would act as a warning signal to indicate that it is time to consider other potential external growth factors before making a determination about whether the population may overproduce, resulting in damage to their own species and others in the environment.

## 4 Example Research

When thinking about the Euler-Lotka equation, the typical usage is on the macro scale, looking at entire populations of creatures. A recent interesting study describes applying the Euler-Lotka equation to cell divisions in microscopic organism populations. The research paper explains that there are fluctuations in cell division times which significantly affect the growth rate. If those fluctuations are uncorrelated, he explains that the growth rate can be accurately predicted by the Euler-Lotka equation. However, due to inherited physical properties of the cell, there can be correlations between cell division times. The researcher derives a generalized form of the Euler-Lotka equation which is valid for the correlated cell divisions.

$$\lim_{\Delta \rightarrow \infty} \frac{1}{\Delta} \ln \left\langle e^{-\Lambda \sum_{i=1}^{\Delta} \tau_i} \right\rangle_{\{\tau_i\}} = -\ln 2,$$

Figure 3: Generalized form of Euler-Lotka equation from recent cell division study

If the cell divisions are uncorrelated, the left hand side of the equation reduces and becomes equivalent to the traditional Euler-Lotka equation. To generate this equation, the researchers “used a result in queuing theory that was recently applied in stochastic thermodynamics and to study enzyme replicating information.” This research is still quite novel, but they expect this generalized form to be helpful to perform further analysis of population growth in single-cell experiments (Pigolotti).

## 5 Optimal Experimental Design

It would be rather difficult to design and conduct an experiment to test the Euler-Lotka equation. It would likely be unethical to risk viability of a population in order to test the Euler-Lotka equation; rather, an observational study would be optimal for studying the model.

The distinction between the two types of studies becomes clear with some explanation. In an experiment, researchers randomly assign a treatment to a group so that the researchers can draw the cause and effect (causal) conclusion. In an observational study,

however, the researchers measure or survey members of a sample without trying to affect the members or manipulating the variables. This means that while the evidence provided by an observational study may generally be less strong than the experimental study, it is also non-invasive. For a team of researchers to go into a population and begin manually changing the population to see if the growth rate would differ would cause more harm to the population than is justifiable for the extra evidence provided by the experimental conditions (Khanna).

So, it is clear that an observational study would be an appropriate setting to perform an analysis using the Euler-Lotka equation. Taking a census of the population over time and keeping track of when females give birth and die will provide all data necessary for the equation, and long-term tracking of overall population numbers will determine whether the estimate provided by the Euler-Lotka equation served as an accurate model for growth of the overall population.

# **Part 2: Original Research Component**

## **6 Introduction**

The objective of this original research component is to enhance my understanding of population growth predictions by developing a growth simulation model from scratch, and applying the Euler-Lotka equation to the simulation in order to compare different types of growth. In particular, I was inspired by the Pigolotti paper, explained in section 4. In short, this paper describes the difference in applying the Euler-Lotka equation to microscopic organisms, and how the fluctuations in cell division times being correlated or uncorrelated significantly affect the accuracy of the Euler-Lotka model's prediction for growth. Specifically, the Euler-Lotka model should be able to accurately predict uncorrelated fluctuations, but is less effective when it comes to correlated fluctuations. My simulation creates a model for both correlated and uncorrelated cell growth, and then I extract the necessary data from each model to compare how well the Euler-Lotka equation would predict the actual growth for each scenario.

## **7 Materials and Methods**

### **7.1 Materials**

I created the simulation using the Python programming language. Specifically, Python Version 3.6.9 was used for the writing and interpretation of the code. The development platform used was Visual Studio Code, but for the purposes of reproduction, any code editor will suffice. There were several libraries that were used for this program. The first one is the Python random library. This library was used to introduce some element of random variability into the specific age at which each cell would reproduce. This is necessary because random events and natural variability within a population must be accounted for to receive genuine results. I also included the abc (Abstract Base Classes) library to incorporate abstract classes within my implementation. More detail about where this is used can be found in Section 7.2, Methods. In order to provide a graphical representation of the results, I also included the matplotlib library. Finally, I imported a couple of mathematical helper libraries to ease the computational load; specifically, I imported the reduce function and the numpy library. These are all the necessary materials for creating this simulation program.

## 7.2 Methods

In this section, I will simply be explaining the necessity of the different classes and methods that were used in the program; the technical details will be touched upon more closely in Section 8, Calculations. The most important piece of this puzzle was figuring out how to represent a cell programmatically. I wanted to be able to implement two different types of cells: those whose growth fluctuations are correlated and those whose are uncorrelated. This seemed like an appropriate place to utilize the computer science concept of inheritance (not to be confused with the genetic concept). In the object-oriented programming portion of computer science, inheritance refers to the ability of an object to take on one or more characteristics from other classes of objects. Because of this special relationship, I created three classes. The base class is called `Cell`, and the two other classes that inherit the attributes and methods of `Cell` are called `UncorrelatedCell` and `CorrelatedCell`. When a new `Cell` is created, it is initialized at age 0 and given a predetermined, random lifespan. Cells also have the ability to grow, where its age increases, and indicate whether it is ready to divide.

It is also necessary to set up the simulated environment in which the cells will grow. This is done in the `uncorrelated_sim()` function for uncorrelated cells and in the `correlated_sim()` function for correlated cells. Each function follows the same structure, and these functions make up the simulation portion of the program. Additionally, since we are trying to compare the simulated results to the result of the Euler-Lotka equation, I incorporated an algorithm into the simulation that would generate the necessary input parameters for the Euler-Lotka equation, so that we could also calculate an estimation for the population growth.

## 8 Calculations

The first development made towards this simulation was the abstract `Cell` class. This class was created to be abstract because it cannot be instantiated on its own; rather, the two subclasses will simply use its attributes and methods. The benefit of this inheritance hierarchy is that the two different types of cells can have similar functionalities without the code being redundant. There is a static variable in the `Cell` class called “`id`” that gives unique names to each created cell. The constructor method will instantiate a `Cell` object at age 0. The other method of note is the `grow()` method, which will increment a cell’s age and determine whether it is time for the cell to split based on the cell’s determined division time. This value is set up differently in each of the two subclasses.

```

12 class Cell(ABC):
13     id = 1
14
15     @abstractmethod
16     def __init__(self):
17         self.id = Cell.id
18         Cell.id += 1
19         self.age = 0
20
21     def getAge(self):
22         return self.age
23
24     def getDivisionTime(self):
25         return self.division_time
26
27     '''
28     Increment age by 1
29     If age is >= division time, cell divides (return False)
30     and indicates 2 new cells should be created
31     '''
32     def grow(self):
33         self.age += 1
34         return self.age <= self.getDivisionTime() or self.getDivisionTime() <= 0
35
36     def __str__(self):
37         return f"Cell_{self.id}:\t{self.type}\tAge={self.age}\tDivision_Time={self.division_time}"

```

Figure 4: The Cell class

Each subclass only contains a constructor method, and the difference is how the division time is initialized. For the uncorrelated cell, the division time is randomly selected to be any integer from 2 through 8. We know that this is a simplified version of uncorrelated cell division, because fluctuations in cell division times would be completely independent of its parent cells or any other cells in the population. For the correlated cell, the division time is randomly selected to be any integer within 3 units of time of its parent cell's division time. So, its cell division time is at least somewhat influenced by its parent's division time, meaning that this is an implementation of correlated cell divisions.

```

40 class UncorrelatedCell(Cell):
41     def __init__(self):
42         super().__init__()
43         self.type = "Uncorrelated"
44         self.division_time = random.randint(2, 8)
45
46
47 class CorrelatedCell(Cell):
48     def __init__(self, parent_div_time = 5):
49         super().__init__()
50         self.type = "Correlated"
51         self.division_time = random.randint(parent_div_time - 3, parent_div_time + 3)

```

Figure 5: The UncorrelatedCell and CorrelatedCell subclasses

The next important step to describe is how the simulation runs for each cell type. For each period of time, each cell in the population is aged up. If any cell has reached its division time, it will be replaced by two new Cell instances. Throughout the process, the population at time is recorded, as well as the ratio of

births to potential “mothers.” This data is recorded so that it can be used in the euler\_lotka\_estimation method. This method performs the discrete summation of the number of individuals surviving to a particular age multiplied by the birth rate per capita for mothers of that same age. The code for these sections is shown below.

```

54 def uncorrelated_sim():
55     P = [0 for _ in range(100)]
56     num_potential_mothers = [0 for _ in range(100)]
57     num_mothers = [0 for _ in range(100)]
58     uncorrelated_cells = [UncorrelatedCell() for _ in range(5)]
59     population = []
60     total_num = 5
61
62     for i in range(T):
63         population.append(len(uncorrelated_cells))
64         new_cells = []
65         for cell in uncorrelated_cells:
66             num_potential_mothers[cell.getAge()]+=1
67             P[cell.getAge()]+=1
68             if(not cell.grow()):
69                 # cell divides
70                 num_mothers[cell.getAge()-1]+=1
71                 del cell
72                 new_cells.append(UncorrelatedCell())
73                 new_cells.append(UncorrelatedCell())
74                 total_num+=2
75         uncorrelated_cells+=new_cells
76         print(f"Time {i}: Population: {population[i]}", end='\r')
77
78     # clean up
79     for cell in uncorrelated_cells:
80         del cell
81
82     plt.plot(range(1,len(population)+1), population, '+', label="Uncorrelated")
83     m = (np.divide(np.array(num_mothers), np.array(num_potential_mothers))).tolist()
84     return [(p/total_num) for p in P], m, population

```

Figure 6: The function to simulate uncorrelated cell growth

```

87 def correlated_sim():
88     P = [0 for _ in range(100)]
89     num_potential_mothers = [0 for _ in range(100)]
90     num_mothers = [0 for _ in range(100)]
91     correlated_cells = [CorrelatedCell() for _ in range(5)]
92     population = []
93     total_num = 5
94
95     for i in range(T):
96         population.append(len(correlated_cells))
97         new_cells = []
98         for cell in correlated_cells:
99             num_potential_mothers[cell.getAge()]+=1
100             P[cell.getAge()]+=1
101             if(not cell.grow()):
102                 # cell divides
103                 num_mothers[cell.getAge()-1]+=1
104                 parent_div_time = cell.getDivisionTime()
105                 del cell
106                 new_cells.append(CorrelatedCell(parent_div_time))
107                 new_cells.append(CorrelatedCell(parent_div_time))
108                 total_num+=2
109         correlated_cells+=new_cells
110         print(f"Time {i}: Population: {population[i]}", end='\r')
111
112     # clean up
113     for cell in correlated_cells:
114         del cell
115
116     plt.plot(range(1,len(population)+1), population, 'x', label='Correlated')
117     m = (np.true_divide(np.array(num_mothers), np.array(num_potential_mothers))).tolist()
118     return [(p/total_num) for p in P], m, population

```

Figure 7: The function to simulate correlated cell growth

```

121 def euler_lotka_estimation(P, m):
122     result = 0
123     for i in range(len(P)):
124         if(not np.isnan(P[i])) and (not np.isnan(m[i])):
125             result+=(P[i]*m[i])
126     return result
127

```

Figure 8: Calculation of the Euler-Lotka growth rate estimation

Finally, the last step is to put all the components together and run the simulation from the main function. First, both simulations are run, and the resulting population growth charts are plotted and saved. Then, the outputs are used to model the growth using the Euler-Lotka equation, which will provide a theoretical estimate for how fast the population should be growing. The results will then be printed to the screen.

```

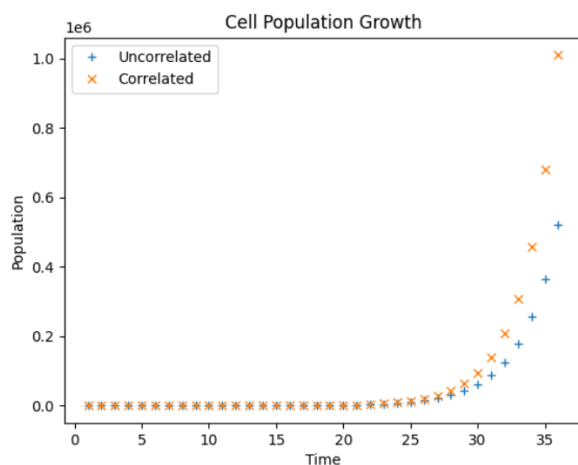
136 if __name__ == "__main__":
137     # Part 1: Simulation of Growth
138     print("\033[1m")
139     print("." * 50)
140     print("Starting Simulation\033[0m\n")
141
142     print("Uncorrelated:")
143     P_uncorr, m_uncorr, pop_uncorr = uncorrelated_sim()
144     print("\nCorrelated:")
145     P_corr, m_corr, pop_corr = correlated_sim()
146
147     plt.title("Cell Population Growth")
148     plt.xlabel("Time")
149     plt.ylabel("Population")
150     plt.legend()
151     plt.savefig("cell_divisions.png")
152
153     print("\n\n\033[1mSimulation Complete\n")
154     print("." * 50)
155     # Part 2: Comparison to Euler-Lotka Model predictions
156     print("Starting Model Checking\033[0m\n")
157
158     R0_uncorr = euler_lotka_estimation(P_uncorr, m_uncorr)
159     R0_corr = euler_lotka_estimation(P_corr, m_corr)
160
161     ratios = list(filter((1).__ne__, [pop_uncorr[i]/pop_uncorr[i-1] for i in range(1,len(pop_uncorr))]))
162     actual_uncorr = reduce(lambda a, b: a + b, ratios) / len(ratios)
163     ratios = list(filter((1).__ne__, [pop_corr[i]/pop_corr[i-1] for i in range(1,len(pop_corr))]))
164     actual_corr = reduce(lambda a, b: a + b, ratios) / len(ratios)
165
166     print(f"Uncorrelated Growth Rate\n\tEstimate:\t{R0_uncorr}\n\tActual:\t{actual_uncorr}")
167     print(f"Correlated Growth Rate\n\tEstimate:\t{R0_corr}\n\tActual:\t{actual_corr}")
168
169     print("\n\n\033[1mModel Checking Complete\n")
170     print("." * 50)
171     print("\033[0m")

```

Figure 9: The main function

## 9 Results

I ran the simulation multiple times and recorded the results. Below I included several example outputs. The outputs consist of the graph to visualize population growth from the samples with uncorrelated divisions as well as those with correlated divisions. The printed parts of the output are the numerical data for the Euler-Lotka estimate of population growth rate vs. the actual observed growth rate.



```

.....
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$ python3 cell_divisions.py

.....
Starting Simulation

Uncorrelated:
Time 35: Population: 522039
Correlated:
Time 35: Population: 1010697

Simulation Complete

.....
Starting Model Checking

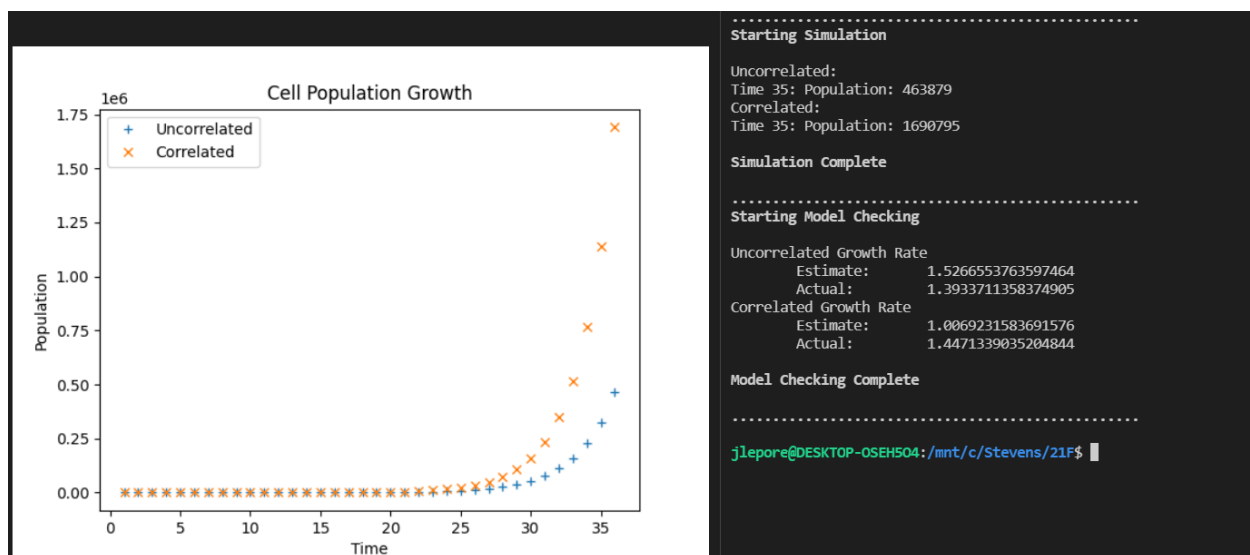
Uncorrelated Growth Rate
  Estimate: 1.5245669565662456
  Actual: 1.398558043215652
Correlated Growth Rate
  Estimate: 1.035022648260575
  Actual: 1.4259194987703234

Model Checking Complete

.....

```

Figure 10: Sample output 1



```

.....
Starting Simulation

Uncorrelated:
Time 35: Population: 463879
Correlated:
Time 35: Population: 1690795

Simulation Complete

.....
Starting Model Checking

Uncorrelated Growth Rate
  Estimate: 1.5266553763597464
  Actual: 1.3933711358374905
Correlated Growth Rate
  Estimate: 1.0069231583691576
  Actual: 1.4471339035204844

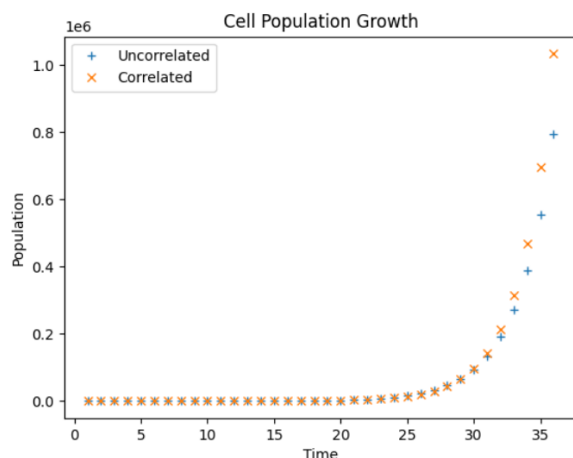
Model Checking Complete

.....
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$ 

```

Figure 11: Sample output 2





```
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$ python3 cell_divisions.py

.....
Starting Simulation

Unrelated:
Time 35: Population: 794315
Correlated:
Time 35: Population: 1033203

Simulation Complete

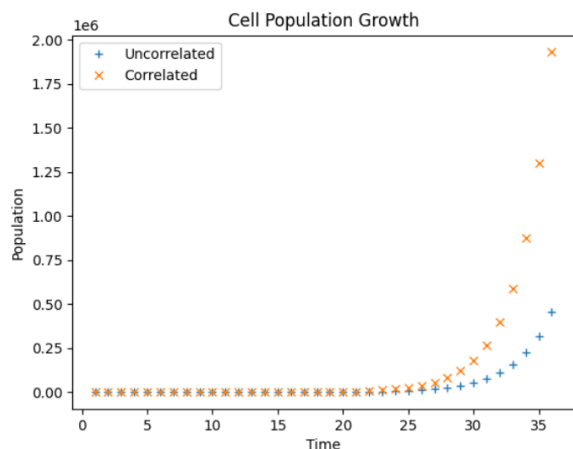
.....
Starting Model Checking

Unrelated Growth Rate
Estimate: 1.5244958882851325
Actual: 1.414290534569052
Correlated Growth Rate
Estimate: 1.0230033697926246
Actual: 1.4297433630638807

Model Checking Complete

.....
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$
```

Figure 12: Sample output 3



```
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$ python3 cell_divisions.py

.....
Starting Simulation

Unrelated:
Time 35: Population: 457279
Correlated:
Time 35: Population: 1929043

Simulation Complete

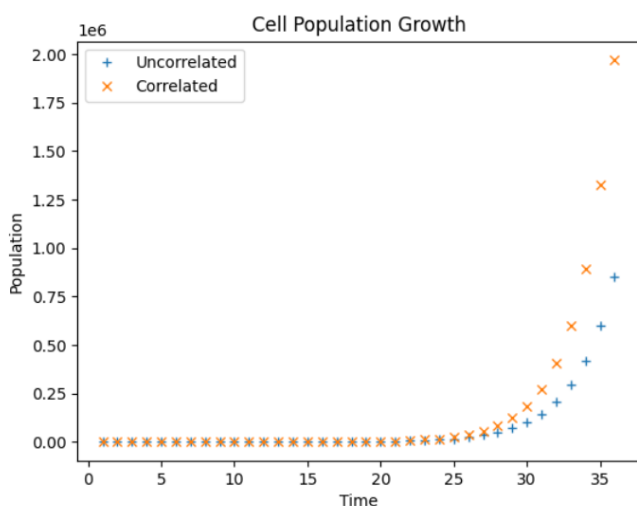
.....
Starting Model Checking

Unrelated Growth Rate
Estimate: 1.524535194651352
Actual: 1.3915552139830503
Correlated Growth Rate
Estimate: 0.9893452201091002
Actual: 1.4499888369758422

Model Checking Complete

.....
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$
```

Figure 13: Sample output 4



```
.....
Starting Simulation

Unrelated:
Time 35: Population: 854113
Correlated:
Time 35: Population: 1968871

Simulation Complete

.....
Starting Model Checking

Unrelated Growth Rate
Estimate: 1.525289675904367
Actual: 1.4161675848262079
Correlated Growth Rate
Estimate: 1.0228278642392663
Actual: 1.451657011579808

Model Checking Complete

.....
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$
jlepore@DESKTOP-OSEH504:/mnt/c/Stevens/21F$
```

Figure 14: Sample output 5

## 10 Discussion

The results of this simulation were very interesting and helpful in enhancing my understanding of the Euler-Lotka equation and the findings of the Pigolotti paper. There are a few intuitive results that we can immediately observe. First, the growth in both uncorrelated and correlated cases is exponential in nature. This very clearly aligns with the assumptions and expectations of a simple age-based population growth model. There are no limiting factors in the environment, and there are more organisms to reproduce over time; therefore the population grows more and more quickly over time.

Another observation from the results lies in the numerical estimation. The Pigolotti paper claims that the Euler-Lotka equation is relatively accurate in estimating cell growth in the uncorrelated case, but not in the correlated case. My findings here support that claim. If we take the data from sample output 1, the growth rate observed for the uncorrelated simulation was about 1.399, whereas the expected value from the Euler-Lotka equation was about 1.525. Intuitively, these numbers are quite close to one another, and with the estimation being greater than one, we know that we should (correctly) expect the population to continue to grow, rather than shrink or remain the same. We can confirm this intuition by calculating the percent error as follows:

$$\% \text{ Error} = \left| \frac{1.399 - 1.525}{1.525} \right| * 100 = 8.26\%$$

This percent error is below 10%, which demonstrates clearly that the actual and expected values were rather close to each other. We can replicate this calculator with the correlated numbers from the same sample output to reveal a much larger percent error:

$$\% \text{ Error} = \left| \frac{1.426 - 1.035}{1.035} \right| * 100 = 37.78\%$$

Once again, this large number intuitively makes sense, as the Euler-Lotka estimation being very close to one barely shows any growth, while, in fact, the growth was even larger than the uncorrelated counterpart. Therefore, we can see the large discrepancy and support Pigolotti's observations with our own simulated results.

## 11 Conclusions

Overall, the study demonstrated that the Euler-Lotka equation can relatively accurately be applied to populations of uncorrelated cell growth, but should not be used for populations in which there are correlated fluctuations in cell growth. In future work, I would like to also programmatically implement the generalized Euler-Lotka formula from the Pigolotti paper to see if the estimations are more consistent for both types of cell divisions.

## Works Cited

- Amarasekare, Priyanga, and Renato M. Coutinho. "The Intrinsic Growth Rate as a Predictor of Population Viability under Climate Warming." *Journal of Animal Ecology*, vol. 82, no. 6, 2013, pp. 1240–1253.,  
<https://doi.org/10.1111/1365-2656.12112>.
- Bacaër, Nicolas. "Euler and the Geometric Growth of Populations (1748–1761)." *A Short History of Mathematical Population Dynamics*, 2011, pp. 11–20.,  
[https://doi.org/10.1007/978-0-85729-115-8\\_3](https://doi.org/10.1007/978-0-85729-115-8_3).
- "Euler–Lotka Equation." Wikipedia, Wikimedia Foundation, 3 Apr. 2021,  
[https://en.wikipedia.org/wiki/Euler%E2%80%93Lotka\\_equation#Analysis\\_of\\_expression](https://en.wikipedia.org/wiki/Euler%E2%80%93Lotka_equation#Analysis_of_expression).
- Hill, Christian. "Solving the Euler-Lotka Equation." *Learning Scientific Programming with Python*, Cambridge University, Dec. 2020,  
<https://scipython.com/book/chapter-8-scipy/examples/solving-the-euler-lotka-equation/>.
- Khanna, Chetna. "Observational vs Experimental Study." *Medium, Towards Data Science*, 5 Dec. 2020,  
<https://towardsdatascience.com/observational-vs-experimental-study-543425a3b3c8>.
- Malepeai, Matthew. "Conservation Ecology." *Graduate Group in Ecology*, University of California, Davis, 5 Apr. 2021,  
<https://ecology.ucdavis.edu/conservation-ecology>.
- National Geographic Society. "Demography." *National Geographic Society*, 23 Mar. 2020, <https://www.nationalgeographic.org/encyclopedia/demography/>.

Okuyama, Toshinori. "Census Timing Alters Stage Duration Distributions in Matrix Population Models." *Ecology and Evolution*, vol. 9, no. 15, 2019, pp. 8500–8508., <https://doi.org/10.1002/ece3.5315>.

Pigolotti, Simone. "Generalized Euler-Lotka Equation for Correlated Cell Divisions." *Physical Review*, vol. 103, no. 6, 30 June 2021, <https://doi.org/10.1101/2021.01.11.426278>.

Rees, Mark, and Stephen P. Ellner. "Age-Structured and Stage-Structured Population Dynamics." Santa Fe Institute, Santa Fe Institute, 2009, [https://centre.santafe.edu/complextime/w/images/complextime/4/43/Rees\\_and\\_Ellner\\_2009.pdf](https://centre.santafe.edu/complextime/w/images/complextime/4/43/Rees_and_Ellner_2009.pdf).

Vandermeer, John. "How Populations Grow: The Exponential and Logistic Equations." *Nature News*, Nature Publishing Group, 2010, <https://www.nature.com/scitable/knowledge/library/how-populations-grow-the-exponential-and-logistic-13240157/>.