Stony Brook University

Analysis of Bacteria Growth

Taking E.coli Bacteria As an Example

AMS 333

Mathematical Biology

Homework 2

Wenhan Gao

State University of New York at Stony Brook

October 2020

1 Introduction

The growth of E.coli bacteria is one of the most simplistic population growth systems, so it has been studied heavily in both biological and mathematical fields. In this report, we will also take E.coli bacteria as an example to delve the growth of bacteria with different mathematical models. Also, in this report, Matlab, a computer programming language and tool, will be utilized in order to compute the mathematics involved in simulating such systems, and create plots to show trends in the growth of bacteria in different models to help us analyze the data. There are various ways of modeling the dynamics of how populations of organisms vary over time, two models, exponential growth and logistic growth, will be compared and discussed in this report.

2 Mathematical Modeling

2.1 Simple Exponential Growth Model

Bacteria reproduce through asexual cell division that a single parent cell duplicate all major cellular constituents, and then divide into two nearly identical offspring cells. Hence, it is intuitive to think of such growth as exponential growth with a constant doubling time, where doubling time is the amount of time the population takes to double in size. In other words, the amount of bacteria doubles after a certain period of time.

We can use a mathematical model to describe this growth in terms of population at a given time:

$$N(t) = N_0 2^{\frac{t}{\tau}} \ , \ t \geq 0 \tag{2.1}$$

where $N(t)$ represents the population size at a given time t, N_0 is the initial condition: population size at time $t = 0$, and τ is the doubling time.

Assume that we begin with a single E.coli bacterium, and grow it under optimal conditions for 1 day, the doubling time of E.coli bacteria under ideal conditions is about 20 minutes, i.e., 1/3 hour. The Matlab portion below displays the growth of E.coli bacteria under simple exponential model over the course of 24 hours.

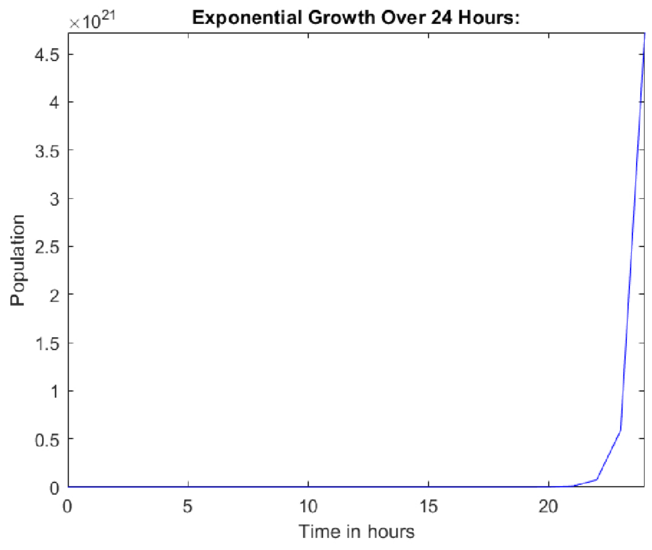
Matlab Implementation

Exponential Growth Start with a Single Bacterium Over 24 Hours:

```
N0 = 1;
doubling_time = 1/3;   % doubling time 20 minites = 1/3 hour
time_list = 0:24;
exponent_list = time_list/doubling_time;
Nt = N0*pow2(exponent_list);   % Nt is a vector of #bacteria over time
N25 = Nt(25)
```

```
N25 = 4.7224e+21
```

```
% Nt(25) is the amount of bacteria when time is 24 hours, aka at end of
one day
plot(time_list,Nt,'b');
xlabel('Time in hours');
ylabel('Population');
axis([0 24 0 Inf]);
title('Exponential Growth Over 24 Hours:');
```



```
volume_cubic_meter = (2e-6)*(1e-6).^2*N25;
%%% bacterial volume, in liters, at the end of one day
volume_liter = 1000*volume_cubic_meter
```

```
volume_liter = 9.4447e+06
```

```
N_L = N25/volume_liter % number of bacteria in 1L
```

```
N_L = 5.0000e+14
```

```
%Time needed to develop 1L bacteria with initial N0 = 100
time_in_hours = log2(N_L/100)*(1/3)
```

```
time_in_hours = 14.0617
```

End of Matlab Implementation

Not surprisingly, with this model, the number of bacteria grows extremely rapidly. At the end of one day, there will be 4.7224e+21, or 4.7224 sextillion(billion trillion), bacteria present. Each bacterium has roughly a dimension of $(2\cdot10^{6}m)(1\cdot10^{6}m)^{2} = 2\cdot10^{-18}m^{3}$; therefore, by a simple calculation as shown below, the volume of bacteria would approximately be 9444700 liters after one day. It's going to take 14.0617 hours or about an overnight for a experimental biologist to grow E.coli bacteria with a liter of medium in her culture beginning with 100 bacteria selected from an agar plate. We can also multiply the $N(t)$ vector by 100 in the above code to observe the time needed. As a remainder, $N(t)$ vector starts from a single bacterium.

```
Nt(11:16)*100 %Nt*100, start from 100 bactiria, first one time = 10

ans = 1×6(Time = 10 hours through Time = 15 hours)
10^15 ×
    0.0001     0.0009     0.0069     0.0550     0.4398     3.5184

%0.5e+15 falls between N15 and N16, t = 14 and t = 15
```

As we can see in the above code, $5\times10^{14} = 0.5\times10^{15}$, which is the number of bacteria in 1L, falls between time = 14 hours and time = 15 hours, which means the biologist need to wait between 14 to 15 hours, we have a consistent answer as the one we actually calculated.

2.2 Logistic Equation Model

Although simple exponential growth model seems to well-imitate the population dynamics of bacteria, it does not. Exponential growth can be observed at relatively small populations; however, as population grows, more resources, such as nutrition, physical space, energy, etc., are depleted, then the growth rate slows. The environment has limited resources that organisms need in order to populate. In other words, the environment has a limited capacity; there is a maximum population size the environment can endure. With that being said, the growth in population size will stop at some point.Therefore, we need a different model that takes these factors into account.

Let's consider the experimental system described above, a experimental biologist to place 100 E.coli bacteria in a liter of medium in her culture and grow up to one day, by the logistic model:

$$\frac{dN}{dt} = R_0 N(t)\left[1 - \frac{N(t)}{K}\right] \tag{2.2}$$

Similarly as the exponential equation, this equation describes the population growth after a specific time t.

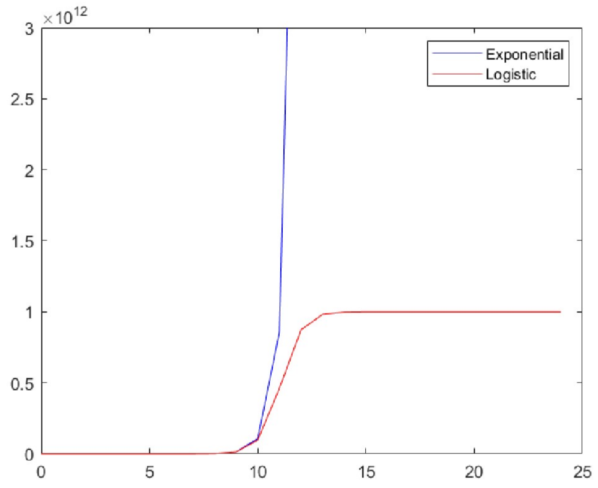
2.2.1 Analytical Form

Equation (2.2) integrates to:

$$N(t) = \frac{KN_0e^{R_0t}}{K - N_0 + N_0e^{R_0t}} \tag{2.3}$$

where $N(t)$ is the population size after a specific time t, N_0 is the initial population size at time $t = 0$, and in this case, $N_0 = 100$, R_0 is the unrestricted growth rate, $R_0 = (\beta - \gamma) = (\frac{\ln 2}{\tau} - 0) = 3\ln 2$, where γ is the death rate; bacteria do not have a intrinsic life span, so the death rate γ is just 0 here. K is the carrying capacity, and it's said to be 10^9 cells per mL, that is 10^{12} cells per L. Let's implement this form with Matlab.

```
plot(time_list, Nt*100,'b');% to plot exp. model
hold on; % to keep above plot
R0 = 3*log(2);
K = 10^12;
Nt_logistic = 100*ones(size(time_list));
% to make a vector of Nt, also make N0 = 100
for t = 1:24
Nt_logistic(t+1) = (K*100*exp(R0*t))/(K-100+100*exp(R0*t));
end
plot(time_list, Nt_logistic,'r');
axis([0 25 0 3*10^12]);
legend('Exponential','Logistic');
```



```
% display Nt, the first one is N0, maximum is 1e12
display(Nt_logistic(1:25));
```

```
1.0e+11 *
Columns 1 through 10
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0003 0.0021 0.0167 0.1324
Columns 11 through 20
0.9696 4.6207 8.7297 9.8214 9.9773 9.9972 9.9996 10.0000 10.0000 10.0000
Columns 21 through 25
10.0000 10.0000 10.0000 10.0000 10.0000
```

End of Matlab Implementation

The plot above displays the growth trends of both simple exponential and logistic model. Bacteria grow infinitely fast and without a limit in size under simple exponential model, and it's fixed in the logistic model, where the growth stops at some point. From the $N(t)\_logistic$ value display above, we can see that after 15 hours, at time $t = 15$, $N(15) = 9.9972 \times 10^{11}$, which is close enough to the maximal bacterial population, 10^{12}, in her culture, so I'll consider this to be having reached the maximal population. Also, at time $= 17$ hours, it is even closer, exactly 10^{12} after rounding by Matlab.

With this model, the biologist should expect to wait 15 hours, before reaching the maximal population. Now, since we know what the environment capacity is, let's find out how long the exponential model takes to reach maximal capacity. Waiting time $T_w = \log_2(10^{12}/100) \cdot \frac{1}{3}) = 11.0731$ hours.(Calculated by Matlab use similar code as calculating time needed to develop 1L bacteria above)

We can also use the same method as we used to find the time to have 1L bacteria, this time, we will find out which time interval $K = 10^{12} = 0.1 \times 10^{13}$ falls in between.

```
%For readers to have a better view, select time =8 through 13 to display
Nt(9:14)*100
```

```
ans = 1×6(Time = 8 hours through Time = 13 hours)
10^13 ×
    0.0002      0.0013      0.0107      0.0859      0.6872      5.4976
```

As we can see $K = 10^{12} = 0.1 \times 10^{13}$ falls between Time = 11 hours and Time = 12 hours. This is different from the logistic model, which is 15 hours. Possible reasons: **the growth rate is different; under the simple exponential model, bacteria grow continuously with a fixed constant doubling time, in other words, bacteria growth rate is always fixed. However, under logistic model, the growth rate changes as population grows because resources, such as nutrition, physical space, energy, etc., are depleted by bacteria that are already present.**

2.2.2 Numerical Solution

For the case given above, we are able to solve the differential equation (2.2) analytically; however, in many cases, it is extremely hard if not impossible to solve it analytically. Therefore, we need numerical methods to analyze and solve the differential equation numerically.

Let's consider the logistic equation again with the same $R_0 = 3\ln 2$ and $K = 10^{12}$ as above:

$$\begin{aligned}\frac{dN}{dt} &= R_0 N(t)\Big[1 - \frac{N(t)}{K}\Big] \\ &= 3\ln 2 \cdot N(t) - 3\ln 2 \cdot N(t) \cdot \frac{N(t)}{10^{12}} \\ &= 3\ln 2 \cdot N(t) - \frac{3\ln 2}{10^{12}} \cdot N(t)^2\end{aligned} \tag{2.4}$$

There is a fixed point N_{fixed} in this system. We set $\frac{dN}{dt} = 0$. Thus, the equation (2.4) becomes:

$$3\ln 2 \cdot N_{fixed} - \frac{3\ln 2}{10^{12}} \cdot N_{fixed}^2 = 0 \tag{2.5}$$

To calculate this N_{fixed}, we will use Newton's root finding method with an initial guess of $0.9K$ and a tolerance threshold of 10^{-2} to solve for the value of N_{fixed}.
Newton's loot finding method:

$$N_{fixed(n+1)} = N_{fixed(n)} - \frac{f(N_{fixed(n)})}{f'(N_{fixed(n)})} \tag{2.6}$$

where N_{fixed} is root approximation, and $N_{fixed(n+1)}$ is the next iteration of $N_{fixed(n)}$.
$f(N_{fixed})$ and $f'(N_{fixed})$ given by:

$$f(N_{fixed}) = 3\ln 2 \cdot N_{fixed} - \frac{3\ln 2}{10^{12}} \cdot N_{fixed}^2 \tag{2.7}$$

$$f'(N_{fixed}) = 3\ln 2 - \frac{6\ln 2}{10^{12}} \cdot N_{fixed} \tag{2.8}$$

Now, let's use Matlab to implement this method and find N_{fixed} with an initial guess of $0.9K$ and a tolerance threshold of 10^{-2}.

Matlab Implementation

```
nr  % to call the function below
```

```
At 1-th iteration, the error value is -2.631793e+10
At 2-th iteration, the error value is -3.170364e+08
At 3-th iteration, the error value is -4.830662e+04
At 4-th iteration, the error value is -1.220703e-03
Fixed point value is 1.000000e+12 and it took 4 iterations
```

```
function nr  % Newton's method to find roots
tol = 1e-2;   %tolerance
N_fixed = 0.9e+12; % initial guess
i = 0;  % to record number of iterations
while abs(f(N_fixed)) > tol
       N_fixed = (N_fixed-(f(N_fixed)/df(N_fixed)));
       i = i + 1;
       error_value = f((N_fixed));
       fprintf('At %d-th iteration, the error value is %e \n',i,
              error_value);
end
number_of_iterations = i;
fprintf('Fixed point value is %e and it took %d iterations', N_fixed,
        number_of_iterations);
end

function y = f(x) % use x represent N_fixed
    y = 3*log(2)*x - (3*log(2)*x.^2)/1e+12;
end

function y = df(x)
    y = 3*log(2) - (6*log(2)*x)/1e+12;
end
```

End of Matlab Implementation

As above Matlab portion displays, it takes 4 iterations to get $N_{fixed} = 10^{12}$. There indeed is a fixed point of the system at $K = 10^{12}$.

2.2.3 Forward Euler Method

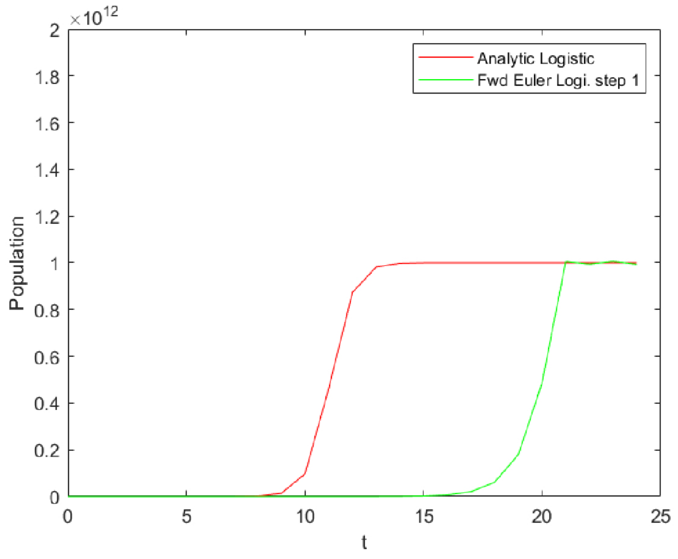
One way to interpret the above differential equation (2.4) is by a method called Forward Euler's Method.

Forward Euler's method is a numerical method for solving initial value differential equations. $\frac{dN}{dt}$ provides us with the slope of the function at a given point t. The initial value $N(0)$ provides us with the first point on the function and also the value of $\frac{dN}{dt}$, the slope of the tangent line at this point. We can approximate the function with a tangent line at the point given by the initial value. However, the tangent line is a good approximation only for a small interval. Therefore, after moving a small interval, this interval is denoted by $\Delta\tau$, we will want to construct a new tangent line.

Now, let's implement the simulation of the growth of bacterial culture described above using Forward Euler in Matlab with a time step of one hour.

Matlab Implementation(Continuation of Analytical Form above)

```
% call below function with deltatau = 1
fwdEuler(1)
```



```
function fwdEuler(deltaTau) % forward euler to solve first order ODE
hold on;
maxT = 24;
N0 = 100;  % initial value of N, N(0)
t = 0: deltaTau: maxT;  % over the course of 24 hrs
Nt = zeros(size(t));
Nt(1) = N0;

for i = 2: length(t)   % interate to find X values
Nt(i) = Nt(i-1)+deltaTau*f(Nt(i-1));  % applying formula
end

plot(t,Nt, 'g')
xlabel('t')
ylabel('Population')
axis([0 25 0 2*10^12]);
legend('Analytic Logistic', 'Fwd Euler Logi. step 1');
end

function dN_dt=f(N)
% derivative of N with respect to t, but rate of change varies as N
varies.
  dN_dt = 3*log(2)*N - (3*log(2)*N.^2)/(1e+12);
end
```

End of Matlab Implementation

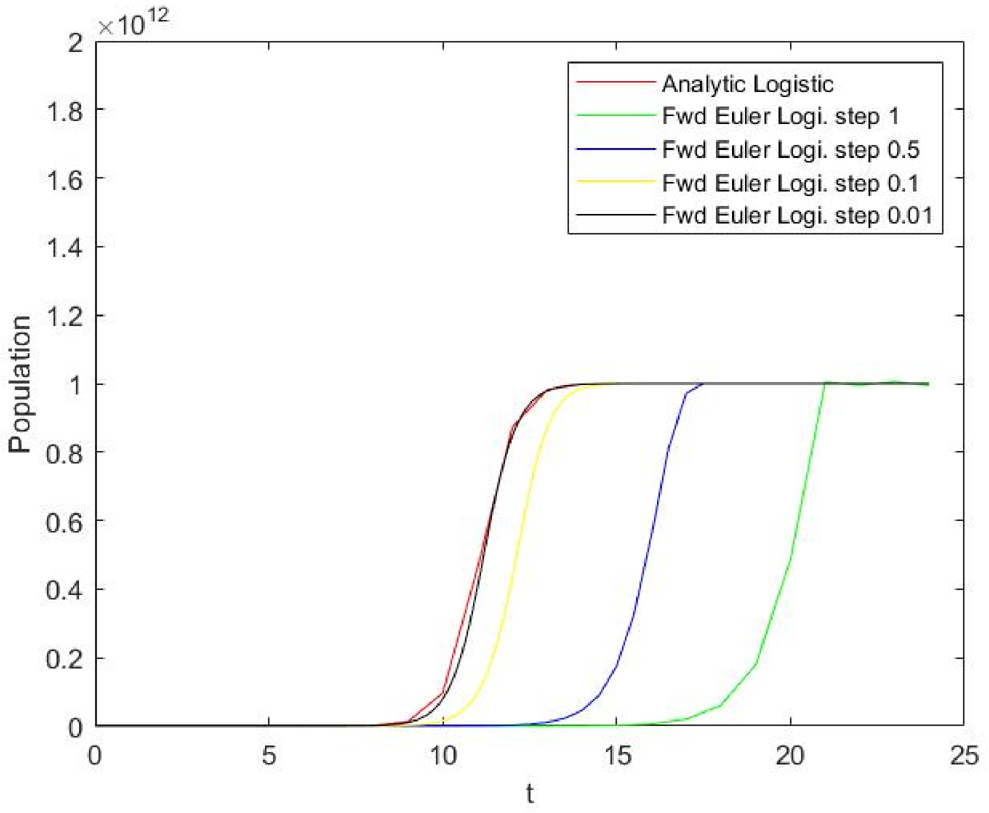
2.2.4 Discussion on the plots

The above figure shows various plots using the exponential growth model and logistics model solved analytically and numerically by Forward Euler method with a time step of an hour.

The exponential model grows exponentially without a limit and far exceeds the environment capacity, as we discussed before, it only describes the system well when the population size is small. The logistic model also grows roughly "exponentially" first, this is because N is small at the beginning, thus $\left[1-\frac{N(t)}{K}\right] \approx 1$, and then the rate of change $\frac{dN}{dt} = R_0N(t)\left[1-\frac{N(t)}{K}\right] \approx R_0N(t)$. As N increases, $\left[1-\frac{N(t)}{K}\right]$ decreases, so the rate of change decreases as well, that is why the curve is getting flattened as time goes. Finally, when N approaches the maximum capacity that is K, $N \approx K$, $\left[1-\frac{N(t)}{K}\right] \approx 1-1 = 0$; therefore, the rate of change $\frac{dN}{dt} \approx 0$, the population stays a constant, that is why the curve becomes a straight line eventually.

The curve derived by solving the DE with a step time of an hour has similar behavior as the analytical curve; they both grows exponentially first, and then the grows rate slows. At the end, the analytical curve stays a straight line, the numerical curve becomes a wave curve around the environment capacity, this is because of the nature of Forward Euler method, the approximate value of the function is given by the previous tangent line approximation, and when the population exceeds the maximum capacity, the slope becomes negative.

As discussed in 2.2.3, the tangent line is only a good approximation when the interval($\Delta\tau$) is small. Therefore, there is a discrepancy between the plots of numerical solution and analytical solution. Smaller the interval gets, more accurate the approximation is. Therefore, we want to have smaller time steps in order to get a better approximation. Let us try different time steps($\Delta\tau$) in Matlab to illustrate this.



This figure is obtained in a symmetric way as in 2.2.3.

As expected, when we minimize the time steps($\Delta\tau$) , the curve behaves more and more like the analytical solution. **When the time step is set to be 0.01, we have a plot that is close enough to the analytical plot.** So, I'll consider to choose a time step of 0.01 hour as the "best choice" to observe the behavior of bacterial growth described in this paper. However, the smaller time step is, the better and more accurate the approximation gets, but small time step also increases the time complexity, so 0.01 would be a good choice the balances between accuracy and running time.