Stephen Hanna

AMS 333: Mathematical Biology

HW #2

Code

```
%This program creates a plot of the number of yeast cells
every 3 hours
%given a start of a single yeast cell and calculates the
space the yeast
%cells would take up after 3 days
a = 3:3:72
%This creates a vector with values for each 3 hours, from
time 0, up until
%3 days, 72 hours later.
b = zeros(1,24);
%This initializes the vector
for i = 1:1:24
    b(i) = 2^(i*3/1.5)
end
%This loop inputs values for a vector for the number of
yeast cells after a
%given time interval
plot(a,b,'--r','LineWidth',3.0);
%This creates a plot for all values of a vs b with the
requested parameter
fig = figure(1);
%This changes the handle of the plotted figure to fig
print(fig,'Simple Growth Model Starting With One Cell','-
dpdf')
%This saves the figure as a pdf
VolumeFilled = 160*((10^-6)^3)*2^(72/1.5)
%This calculates the number of liters the yeast would
occupy after 3 days
```

Execution

```
b =

  1.0e+14 *

 Columns 1 through 10

   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000

 Columns 11 through 20
```

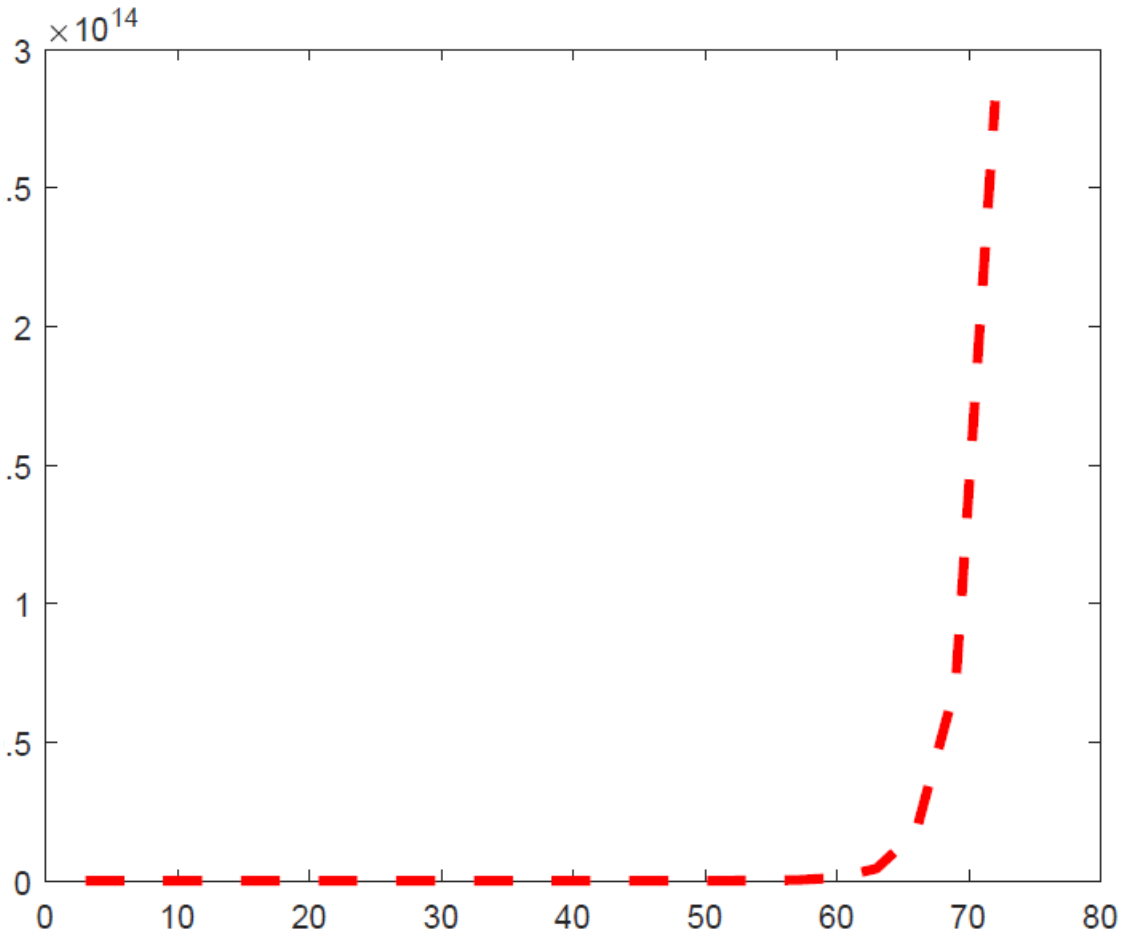
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0002 0.0007 0.0027 0.0110

Columns 21 through 24

0.0440 0.1759 0.7037 2.8147

VolumeFilled =

0.0450



Code:

```
%This program calculates the time it would take to reach
the maximum number
%of yeast cells in a 1L liquid growth medium, assuming the
maximum number
%is defined as when the yeast cells collectively take up 1L
of space. The
%initial amount of cells is 10000
NumberOfCellsInOneLiter = 1/(160*((10^-6)^3))
```

```
%This calculates the number of yeast cells required to fill
1 Liter of
%space
TimeRequiredToFillALiter =
log(NumberOfCellsInOneLiter/10000)/log(2)*1.5
%This calculates the number of hours required for yeast
cells to divide to
%the point that they fill 1 Liter of space
Days = TimeRequiredToFillALiter/24
%This calculates the number of days required for yeast
cells to divide to
%the point of occupying 1 Liter of space.
```

Execution:

NumberOfCellsInOneLiter =

6.2500e+12

TimeRequiredToFillALiter =

43.8289

Days =

1.8262

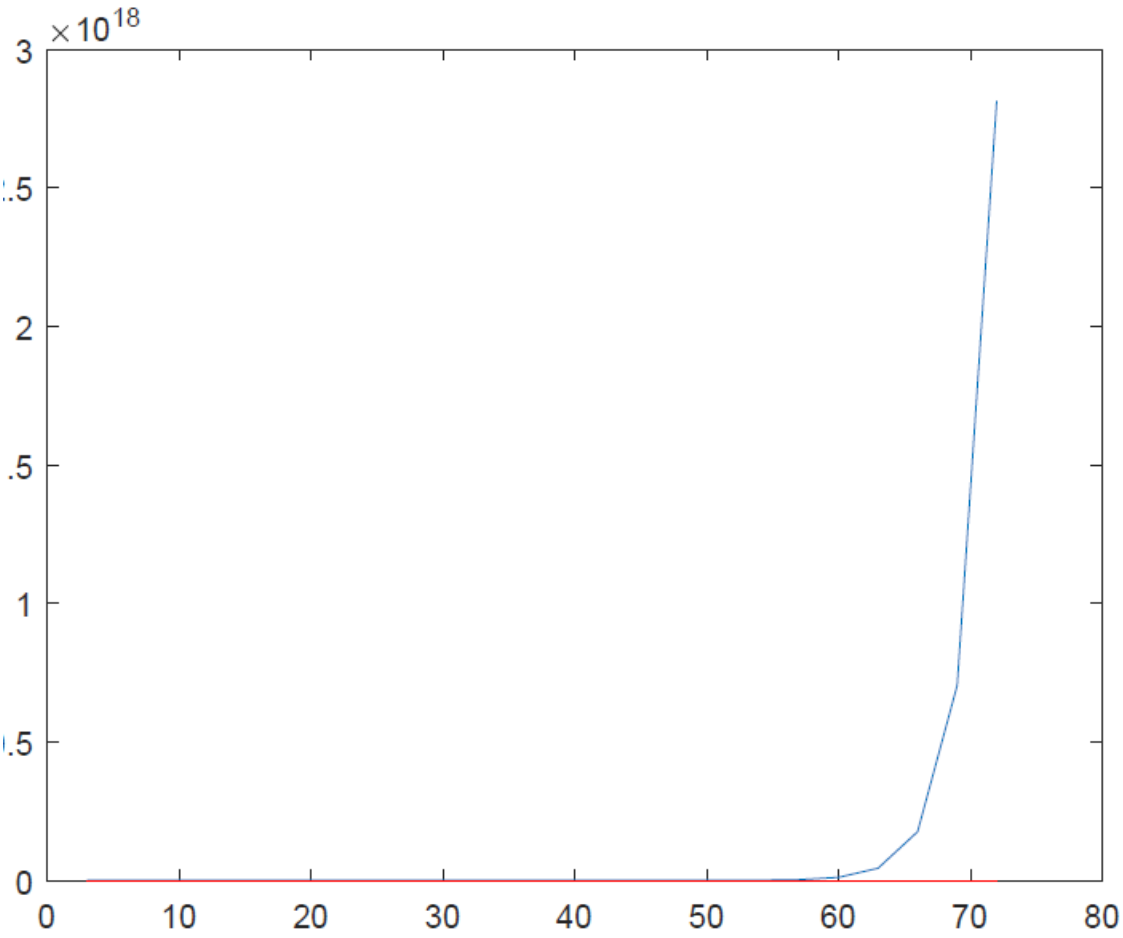
Code:

```
%This program creates a plot of the number of yeast cells
every 3 hours
%given a start of a 10000 yeast cells and calculates the
space the yeast
%cells would take up after 3 days using two different
models
a = 3:3:72;
%This creates a vector with values for each 3 hours, from
time 0, up until
%2.375 days, 57 hours later.
b = zeros(1,24);
%This initializes the vector
c = zeros(1,24);
%This initializes the vector
for i = 1:1:24
b(i) = 10000*2^(i*3/1.5)
%This creates a vector for the number of yeast cells after
a given time
%interval using the simple exponential growth model
```

```
y = exp(log(2)*i*3/1.5);
%This creates a vector for each exponential value in the
logistic equation
%model
c(i) = ((2*10^11)*10000*y)/((2*10^11)-10000+10000*y)
end
%This creates a vector for each value in the logistic
equation model
plot(a,b,a,c,'r');
%This creates a plot for all values of a vs b and a vs c
with the requested
%parameter
fig = figure(1);
%This changes the handle of the plotted figure to fig
print(fig,'Simple Growth Model vs Logistic Model','-dpdf')
%This saves the figure as a pdf
```

Execution:

```
b =

  1.0e+18 *

 Columns 1 through 10

   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000

 Columns 11 through 20

   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0002   0.0007   0.0027   0.0110

 Columns 21 through 24

   0.0440   0.1759   0.7037   2.8147

c =

  1.0e+11 *

 Columns 1 through 10

   0.0000   0.0000   0.0000   0.0000   0.0001   0.0004   0.0016   0.0065   0.0259   0.0996

 Columns 11 through 20

   0.3467   0.9124   1.5408   1.8613   1.9634   1.9907   1.9977   1.9994   1.9999   2.0000

 Columns 21 through 24

   2.0000   2.0000   2.0000   2.0000
```



Discussion:

Given the significantly different shapes of the graphs, I would revise my previous estimate from 1.83 days to 2.5 days. At 2.5 days, the entire liter would be saturated with maximal density of yeast cells. The reason there is a difference in my answers is that, as the population grows, the yeast cells will approach a greater density and be less capable of properly dividing. At 2.5 days, the maximal saturation point will be reached. The reason the saturation point was lower with the first scenario is that there were no limiting factors, whereas now the population density is a limiting factor.

Code(1):

```
%This program creates a plot of the number of yeast cells
every 1 hour
%given a start of a 10000 yeast cells and calculates the
space the yeast
%cells would take up after 3 days using the Forward Euler
method

a = 0:.25:16
```

```
%This creates a vector for the time that has passed
b = zeros(1,65)
%This initializes a vector for the population values
b(1) = 10000;
%This sets the initial population
for i = 2:65
    x = (log(2))*.25*(i-1)/1.5;
    %This determines the R0 value for each run of the loop
    y = x*(b(i-1));
    %This finds the product of the R0 value and the i-1
population
    y1 = b(i-1)/(2*10^11);
    %This solves for part of the limiting factor in the
derivative
    z = y*(1-y1)*.25;
    %This finds the product of the derivative and the
interval of time
    b(i) = (b(i-1))+z
    %This sums the i-1 value and the derivative times the
interval of time.
    %I did all the steps in different parts to analyze for
any errors that
    %may have been made at a different part of the loop

end
%This loop inputs values for a vector for the number of
yeast cells after a
%given time interval
plot(a,b,'--r','LineWidth',3.0);
%This creates a plot for all values of a vs b with the
requested parameter
fig = figure(1);
%This changes the handle of the plotted figure to fig
print(fig,'LogisticEulerForwardMethod','-dpdf')
%This saves the figure as a pdf
```

Execution:

```
b =

  1.0e+11 *

 Columns 1 through 10

   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000

 Columns 11 through 20
```

0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

Columns 21 through 30

0.0000 0.0000 0.0000 0.0001 0.0001 0.0002 0.0004 0.0007 0.0012 0.0022

Columns 31 through 40

0.0042 0.0079 0.0152 0.0295 0.0580 0.1149 0.2276 0.4431 0.8217 1.3670

Columns 41 through 50

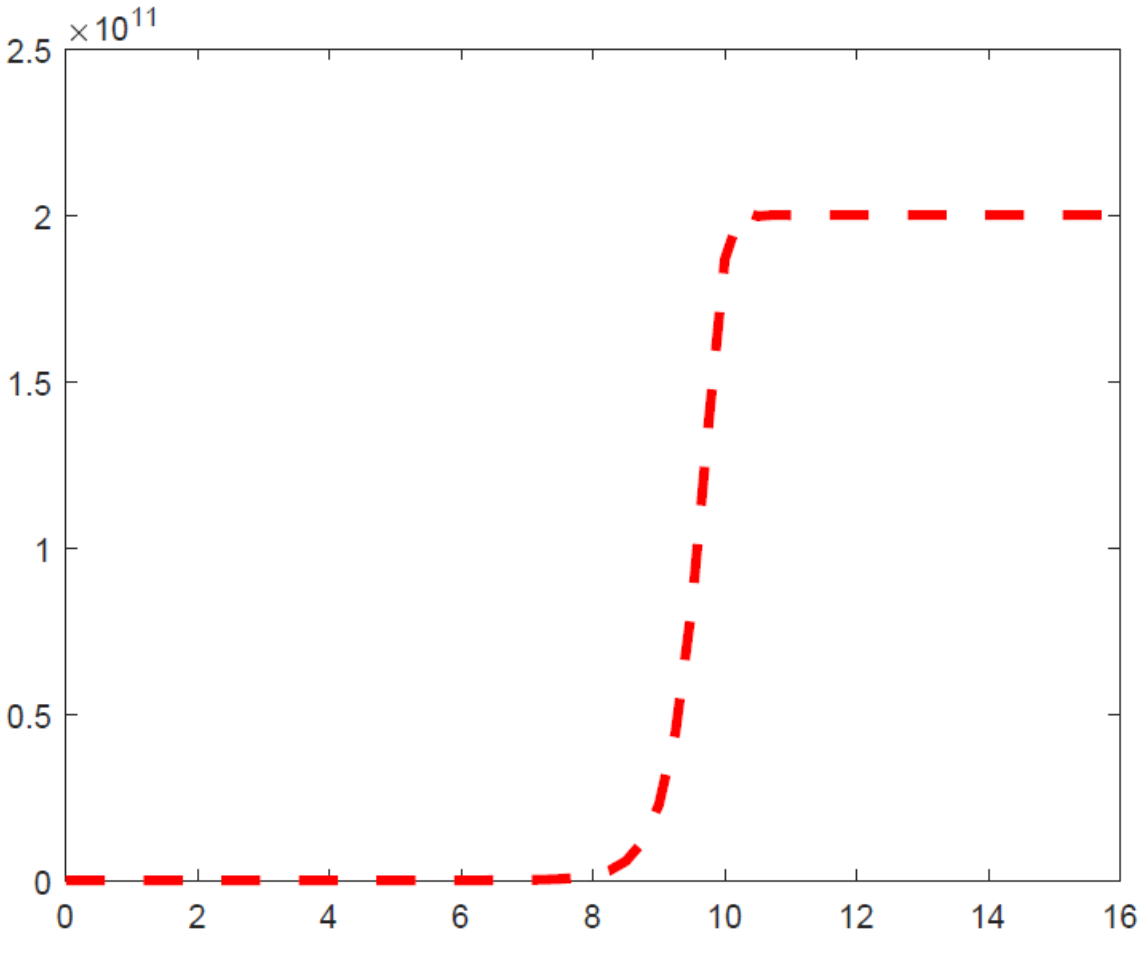
1.8668 2.0140 1.9969 2.0007 1.9998 2.0001 2.0000 2.0000 2.0000 2.0000

Columns 51 through 60

2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000

Columns 61 through 65

2.0000 2.0000 2.0000 2.0000 2.0000

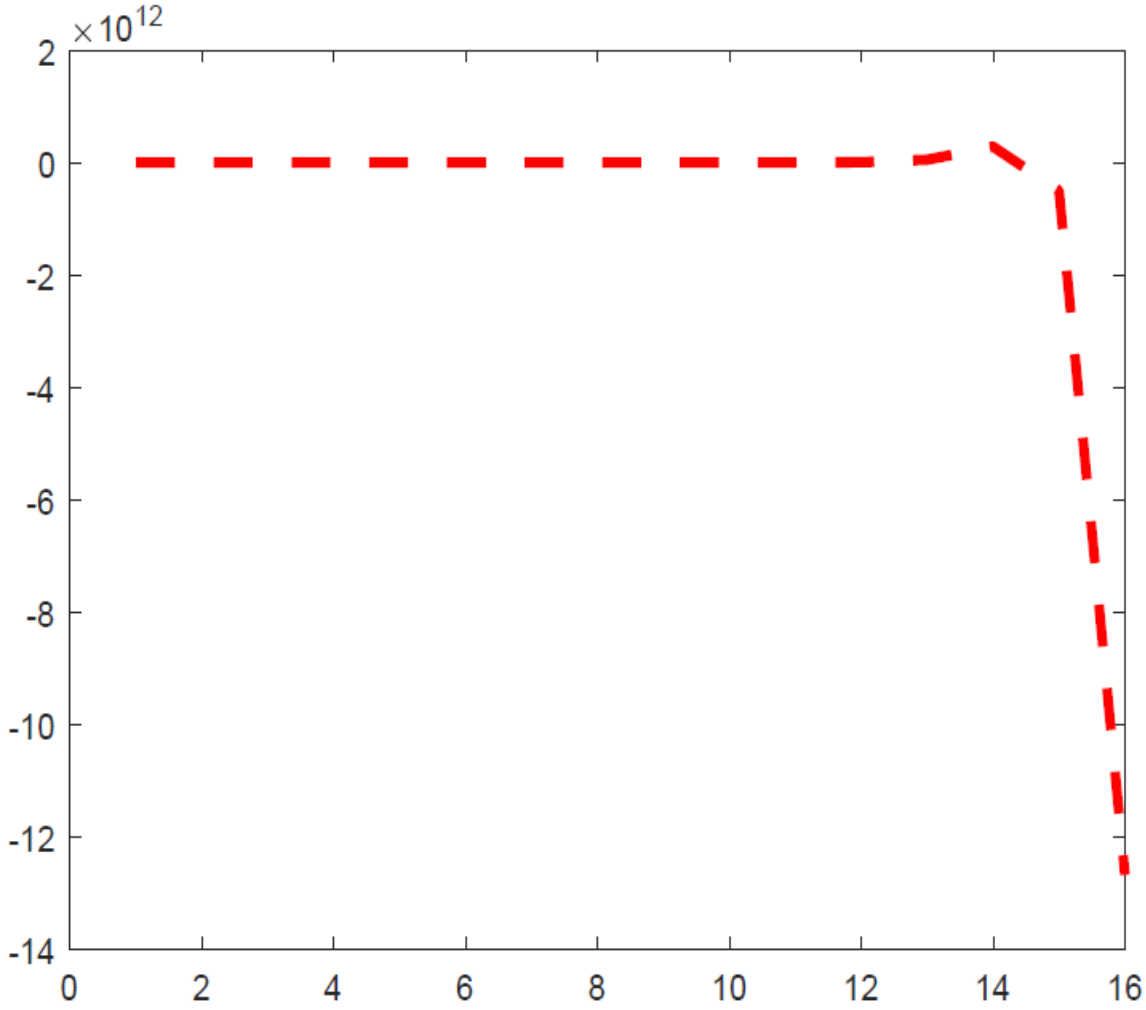


Code(2):

```
%This program creates a plot of the number of yeast cells
every 1 hour
%given a start of a 10000 yeast cells and calculates the
space the yeast
%cells would take up after 3 days using the Forward Euler
method

a = 1:1:16;
b = zeros(1,16);
b(1) = 10000
i(1) = 1
for i = 1:15
    x = (log(2))*i/1.5;
    y = x*b(i);
    y1 = b(i)/(2*10^11);
    z = y*(1-y1);
    b(i+1) = b(i)+z

end
%This loop inputs values for a vector for the number of
yeast cells after a
%given time interval
plot(a,b,'--r','LineWidth',3.0);
%This creates a plot for all values of a vs b with the
requested parameter
fig = figure(1);
%This changes the handle of the plotted figure to fig
print(fig,'LogisticEulerForwardMethod','-dpdf')
%This saves the figure as a pdf
```

Execution:

```
b =

  1.0e+13 *

 Columns 1 through 10

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 11 through 16

   0.0001    0.0008    0.0053    0.0285   -0.0501   -1.2686
```



Discussion:

The two codes are representations of the some equation and application, but with different time intervals. The time interval for the first set of code is .25 hours, whereas the second is 1 hour. The first set of code leads to a saturated population due to the population capacity being reached. This is the expected model, and fits well, albeit not perfectly due to small variations in the population on the saturation curve not being entirely accurate. The second set of code leads to an extinction of the population, which may be expected if the population went drastically over the population capacity. However, that would only be expected using the model and not in real life, as some portion of the populace would likely survive regardless of the rest of the species dying out, with respect to a population capacity being reached. The population capacity is unlikely be surpassed, however, so the first model fits better. The reason the two models are significantly different is that a smaller step interval often leads to a more accurate model. If I were to put the step into milliseconds, its likely to be more accurate than the first set of code. A step using microseconds might be even more accurate than a millisecond code as well. However, at a certain step size, further reductions make a negligible difference to the model.