Modeling Cancer Growth Using Differential Equations

Connor Leipelt

Department of Mathematics

California Polytechnic State University

San Luis Obispo, California

April 2023

APPROVAL PAGE

TITLE: Modeling Cancer Growth Using Differential Equations

AUTHORS: Connor Leipelt

DATE SUBMITTED: April 2023

Joyce Lin
Senior Project Advisor

Signature

Ben Richert
Mathematics Department Chair

Signature

Abstract

According to the CDC, cancer currently is among the top leading causes of death in America. Growth models are an indispensable aspect of cancer treatment, not only by helping researchers understand the complexities of cancer growth but also allowing medical professionals to predict treatment efficacy and thereby improve current treatment schedules. The first portion of this project investigates the application of three models to a set of Fibroblast tumor cells: the Logistic, Bertalanffy, and Gompertz models. The effectiveness of these models is assessed through a normalized mean square error (NMSE). The second portion of the project follows the process of mining through a set of unprocessed data and graphing the results. After procession, the final part consists of the creation of a mathematical differential equation that models the growth of one of the graphs from the newly-sifted data. From this model, one can ascertain how different treatments affects tumor growth, and the structure of cancerous growth in general.

Acknowledgements

First and foremost I would like to thank Dr. Lin for working and guiding me through this project and for being my senior project advisor. She has helped me better understand cancer and the way it grows and how to accurately model data using differential equations. I would also like to thank Dr. Wang from Union College as she was the one who created the initial idea and guidelines for the project. Thanks to Dr. Tully-Doyle and my classmates as well for support and ideas.

Contents

List of Figures

1 Part 1 Modeling Fibroblast Cells

1.1 Types of Differential Equations

1.1.1 Understanding a differential equation and how it models growth

A differential equation is an equation that relates a function(s) and its derivative(s) [11]. A first-order differential equation is an equation that establishes a relationship between a function and its first derivative. The growth of cancer can be modeled by a first-order differential equation by relating the change in the volume V of a tumor over change in time t with an initial volume designated $V(0) = V_0$.

Note: We will be using the variable a to denote the growth rate, the variable K to denote the carrying capacity (directly used in our logistic model), and b to denote the relation between a and K where $K = \frac{a}{b}$.

1.1.2 Logistic Model

Pierre Verhulst introduced the logistic model in 1845 [9]. Its originally function was to model the exponential growth of population. The model begins assuming exponential growth, then transitions to a linear decrease of the relative growth rate in relation to the population size. The Logistic Model has an internal maximum size, called the carrying capacity, which we shall label K.

The differential equation for this model is:

$$\frac{dV}{dt} = aV\left(1 - \frac{V}{K}\right) \tag{1}$$

We will see the application of this model in MATLAB in 2.2.2 and 2.2.6.

1.1.3 Bertalanffy Model

The Bertalanffy Model was introduced by Von Bertalanffy in 1957, for the purpose of investigating the relationship between metabolism and growth in living things [10]. This model is concerned with how cells synthesize energy to promote growth. Assuming that growth is proportional to surface area, as nutrients enter through the surface, it follows that death is proportional to tumor size. This model, importantly, has successfully been used to model human tumor growth.

The differential equation for this model is:

$$\frac{dV}{dt} = aV^{2/3} - bV \tag{2}$$

We will see the application of this model in MATLAB in 2.2.3 and 2.2.7.

1.1.4 Gompertz Model

Introduced by Benjamin Gompertz in 1825, this model deals with human mortality curves and the valuation of life contingencies [3]. Having been previously used to successfully model two types of cancer growth (breast and lung cancer), this model reveals an exponential deterioration of growth.

The differential equation for this model is:

$$\frac{dV}{dt} = V(a - b\ln V) \tag{3}$$

We will see the application of this model in MATLAB in 2.2.4 and 2.2.8.

1.1.5 Normal type of model for cancer growth

For multi-cellular tumor spheroids, cancer growth follows an S-shaped curve consisting of three phases: initial exponential growth, linear growth, and a plateau. Our three models above, Logistic, Bertalanffy, and Gompertz, all reflect the sigmoid nature of growth. Each of the models are autonomous differential equations, therefore they only need an initial value point and are then able to create the model [7].

1.1.6 Error analysis of models

The precision for our models will be evaluated by the normalized mean square error (NMSE). This analysis also minimizes the sum of squared residuals (SSR).

The equation for the NMSE is:

$$NMSE = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i y_i^2} \tag{4}$$

Which uses the SSR as the numerator:

$$SSR = \sum_i (y_i - \hat{y}_i)^2 \tag{5}$$

The approximation uses y_i as the given volume size at a designated time, and $\hat{y}_i$ as the value created by our model for the same specified time. We then take the SSR and normalize.

1.2 Applying Differential Equations to our Data

Important Detail: All three models used are programmed in MATLAB using the built in function ODE45 [8].

1.2.1 Data Set 1

Note: This first data set was given in a paper created by Dr. Wang from Union College [12].

Our first set of data is from the Chinese hamster V79 Fibroblast tumor cell line [7, 12].

t (days)	**V** ($10^9 \mu m^3$)	**t**	**V**	**t**	**V**	**t**	**V**	**t**	**V**
3.46	0.0158	12.39	0.4977	24.33	3.2046	35.20	5.9668	48.29	7.0694
4.58	0.0264	13.42	0.6033	25.58	4.5241	36.34	6.6945	49.24	7.4971
5.67	0.0326	15.19	0.8441	26.43	4.3459	37.29	6.6395	50.19	6.9974
6.64	0.0445	16.24	1.2163	27.44	5.1374	38.50	6.8971	51.14	6.7219
7.63	0.0646	17.23	1.4470	28.43	5.5376	39.67	7.2966	52.10	7.0523
8.41	0.0933	18.18	2.3298	30.49	4.8946	41.37	7.2268	54.00	7.1095
9.32	0.1454	19.29	2.5342	31.34	5.0660	42.58	6.8815	56.33	7.0694
10.27	0.2183	21.23	3.0064	32.34	6.1494	45.39	8.0993	57.33	8.0562
11.19	0.2842	21.99	3.4044	33.00	6.8548	46.38	7.2112	59.38	7.2268

Note: This experiment was tracked over 60 days and had 45 total measurements.

The optimal parameter values given by the paper by Dr. Wang for a, b, and K for Data Set 1 are as follows:

Logistic	a = 0.3389,	b = 0.0489,	K = 6.3905
Bertalanffy	a = 0.4340,	b = 0.2158,	K = 2.0111
Gompertz	a = 0.2375,	b = 0.1179,	K = 2.0144

1.2.2 Logistic Application 1

To model the given data, we have created a MATLAB code that takes in the initial condition of the data and outputs a logistic model. The code has been placed in the appendix for cleanliness. Here is the output of the MATLAB code:

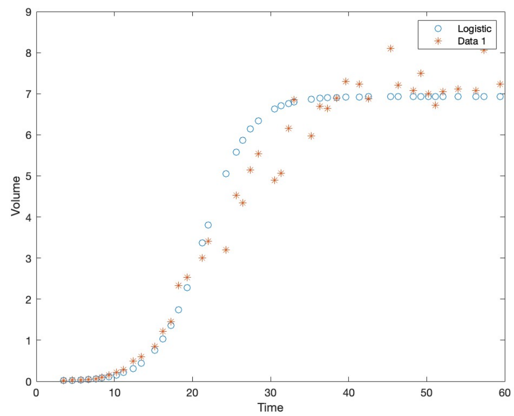

Figure 1: Logistic Model for Data Set 1

1.2.3 Bertalanffy Application 1

Similarly to 1.2.2, the Bertalanffy code takes in an initial condition and outputs a model. The code has been placed in the appendix for cleanliness.

Here is the output of the MATLAB code:

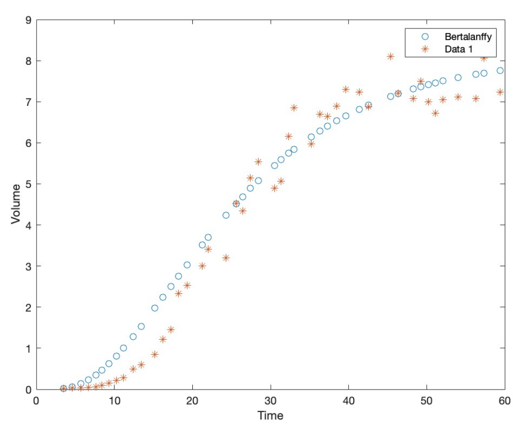

Figure 2: Bertalanffy Model for Data Set 1

1.2.4 Gompertz Application 1

As 1.2.2 and 1.2.3, the Gompertz code takes in an initial condition and outputs a model. The code has been placed in the appendix for cleanliness.

Here is the output of the MATLAB code:

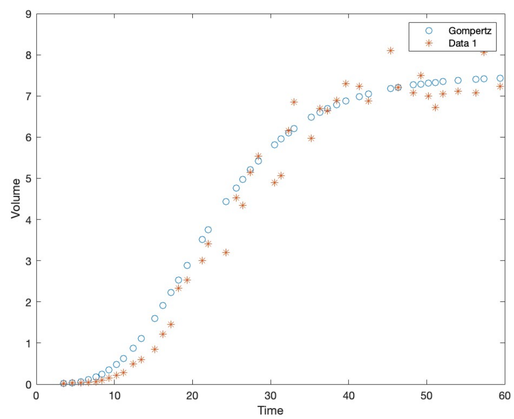

Figure 3: Gompertz Model for Data Set 1

1.2.5 Data Set 2

This is a second set of data that models the growth of mice tumors that arise in situ from normal skin as a result of chemical mutagenesis. This data resembles the complexity and dynamics of tumorigenesis more closely as it develops naturally in the context of a full immune system [6, 12].

t (days)	**V** (mm^3)	**t** (days)	**V** (mm^3)	**t** (days)	**V** (mm^3)	**t** (days)	**V** (mm^3)
0	1.0410	21	35.1929	42	87.7625	62	407.5250
3	4.3156	24	32.3758	45	98.7135	65	303.3899
7	2.8523	28	36.9683	49	131.8011	69	393.7301
10	4.1246	31	62.4162	52	147.4074	72	337.5277
14	5.9634	35	135.3351	56	267.1758	76	376.4903
17	8.2353	38	137.0422	59	421.1538	78	395.8532

Note: This experiment consists of 24 measurements over an 11 week time period.

The optimal parameter values given by the paper by Dr. Wang for a, b, and K for Data Set 2 are as follows:

Logistic	a = 0.1402,	b = 0.000357,	K = 392.7171
Bertalanffy	a = 0.4947,	b = 0.0465,	K = 10.6387
Gompertz	a = 0.2619,	b = 0.0413,	K = 6.3414

1.2.6 Logistic Application 2

Just like in 1.2.2, we have created a MATLAB code that takes in the initial condition of the data and outputs a logistic model. The code has been placed in the appendix for cleanliness.

Here is the output of the MATLAB code:

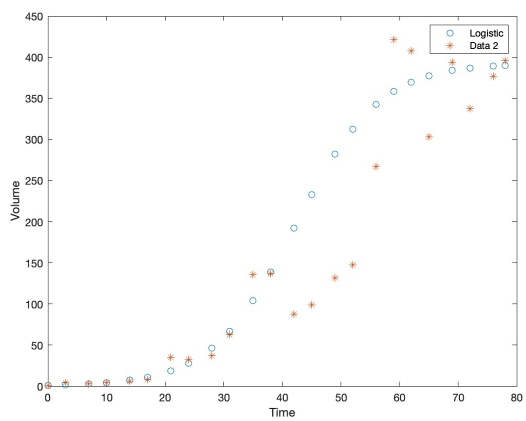

Figure 4: Logistic Model for Data Set 2

1.2.7 Bertalanffy Application 2

Following the same procedure done in 1.2.3, we used the Bertalanffy code, taking in an initial condition, and created a model for the data. The code has been placed in the appendix for cleanliness.

Here is the output of the MATLAB code:

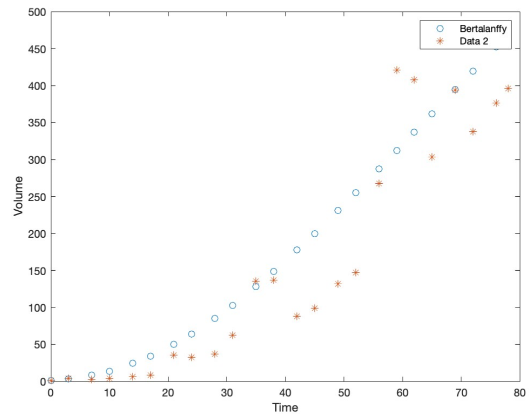

Figure 5: Bertalanffy Model for Data Set 2

1.2.8 Gompertz Application 2

Using the same steps we used in 1.2.4, we used our Gompertz code and the initial data to create a model. The code has been placed in the appendix for cleanliness.

Here is the output of the MATLAB code:

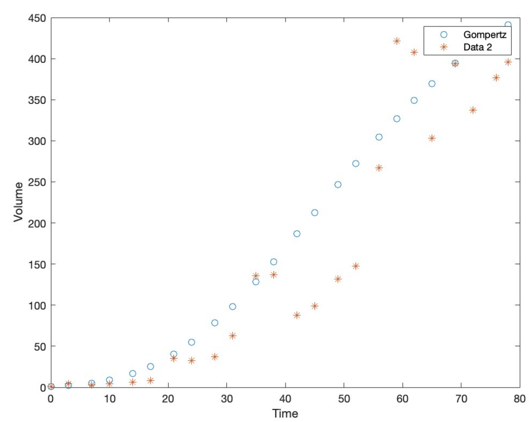

Figure 6: Gompertz Model for Data Set 2

1.3 Error Analysis and Conclusions

1.3.1 Error Analysis for Data Set 1

As talked about above, we will be analysing the accuracy of our models through a normalized mean square error (NMSE). We used equation 4 and equation 5 for the Logistic, Bertalanffy, and Gompertz models. These values are also obtained through the code in A.1 respective to their model and Data Set.

Here are the error values for Data Set 1:

Logistic	SSR = 0.0878	NMSE = 0.0159
Bertalanffy	SSR = 0.2818	NMSE = 0.0117
Gompertz	SSR = 0.0427	NMSE = 0.0077

The Gompertz model performed the best for Data Set 1 as it had a 0.77% error relative to the actual data recorded.

1.3.2 Error Analysis for Data Set 2

Alike 1.3.1, we used equation 4 and equation 5 to analyze the accuracy of our models for Data Set 2. These values are also obtained through the code in A.1 respective to their model and Data Set.

Here are the error values for Data Set 2:

Logistic	SSR = 33.0853	NMSE = 0.0696
Bertalanffy	SSR = 5.25e + 03	NMSE = 0.0590
Gompertz	SSR = 2.0768e + 03	NMSE = 0.0586

The Gompertz model performed the best for Data Set 2 as well with an NMSE of 5.86% performing narrowly better than the Bertalanffy model.

2 Part 2 Data Processing

2.1 Second Set of Data

The second part of this project is focused on processing through a large Excel file of data given by Dr. Wang. The goal is to decipher the raw data into useful information that can then be compared with models.

This set of data is a continuation of the data recorded in the research paper that was used in our Data Set 2 [6]. It is a large Excel file containing recorded data of five distinct groups of mice. The five groups are labeled CM3, CM4, CM5, CM6, and CM7. Each mouse had three tumors which were measured over a given period of time. The Excel file has 9 tabs: a Main experiment summary tab, Tumor sizes CM3, Tumor sizes CM4, Tumor sizes CM5, Tumor sizes CM6, Tumor sizes CM7, Weights (for all mice), Drug injection dates (for all mice), and an Endpoints summary tab.

The main goal of this portion is to find a way to interpret the data and use the drug injection information to see how the cancerous tumors are affected by the injections. The two drugs used in this experiment were Imiquimod and 5-Fluorouracil (5-FU).

2.2 Processing Data

Important Note: Mouse was sacrificed if the size of the cancer tumor reached $\sim 1cm$ in diameter.

2.2.1 Organization of Data

To start the processing we dove directly into each group.

- CM3 had 11 mice in its group: CM37, CM 38, CM39, CM40, CM41, CM42, CM43, CM46, CM47, CM48, CM49.

  9 of the 11 mice died: 7 because of the size of the cancer tumor reaching $\sim 1cm$, 2 because of natural causes unrelated to the cancer.

  9 of the 11 mice had 3 tumors, while the other 2 had 4 tumors. Both mice with 4 tumors died because of the size of their tumors.

- CM4 had 9 mice in its group: CM51, CM52, CM53, CM54, CM55, CM56, CM57, CM58, CM59.

  1 of the 9 mice died: 1 because of the size of the cancer tumor reaching $\sim 1cm$.

  All 9 mice had 3 tumors.

- CM5 had 9 mice in its group: CM60, CM62, CM63, CM66, CM67, CM68, CM69, CM70, CM 71.

  3 of the 9 mice died: 3 because of the size of the cancer tumor reaching $\sim 1cm$. Side note: CM5 had 12 mice but 3 died of natural causes and their data was not recorded in the Excel file.

  All 9 mice had 3 tumors.

- CM6 had 8 mice in its group: CM72, CM73, CM74, CM75, CM76, CM77, CM78, CM79.

  2 of the 8 mice died: 2 because of the size of the cancer tumor reaching $\sim 1cm$.

  All 8 mice had 3 tumors.

- CM7 had 8 mice in its group: CM83, CM84, CM85, CM86, CM90, CM91, CM92, CM93.

  2 of the 8 mice died: 2 because of the size of the cancer tumor reaching $\sim 1cm$.

  All 8 mice had 3 tumors.

2.2.2 Understanding Tumor Data

The given tumor data has a specific structure. The research group took 3 measurements in a horizontal direction and vertical direction of each tumor for each documented measurement day. This means that each mouse essentially had 18 measurements taken for each time stamp, 6 for each of the three tumors. This is good, but not enough for us to compare our volume models to the raw data.

We use the shape of an ellipsoid to describe the shape of the cancerous tumors. To calculate the volume of our ellipsoid shape one needs three measurements: the length, width, and height of said object. As we were given two of the three measurements, we decided to take the average of the 3 horizontal measurements and the average of the 3 vertical measurements and average the two values.

Note: This is equal to taking the average of H1 and V1, and the average of H2 and V2, and the average of H3 and V3, and then averaging those 3 averages.

Now that we have a third value attributed to our tumor we can calculate the volume of the mice tumors from the raw data!

The formula we used to calculate the volume of each tumor at its respective time is the formula for the volume of an ellipsoid:

$$V = \frac{4}{3}\pi r_1 r_2 r_3 \tag{6}$$

r_1, r_2, r_3 are half the length of the horizontal measurement, half the length of the vertical measurement, and half the length of the newly computed average length respectively.

Now we are finally capable of graphing the raw data. It is important to remember that each mouse has three distinct tumors which are all being monitored and recorded.

2.2.3 Graphing CM3

To model this given data for CM3 we have created a lengthy MATLAB code that will be placed in the appendix (A.2) in order to try to create some semblance of cleanliness. The code goes through the process of taking in matrix files of the data and creating a new third length and determining the volume from this and plotting the data found. The following graphs are the processed data of CM3:

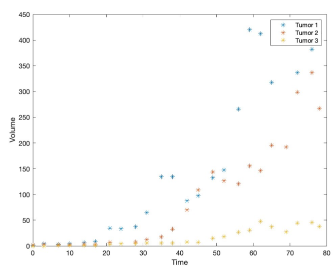

Figure 7: CM37 Graph

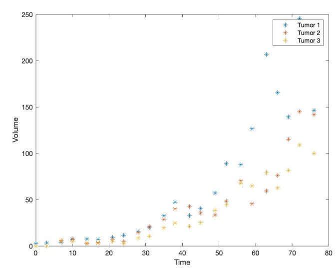

Figure 8: CM38 Graph

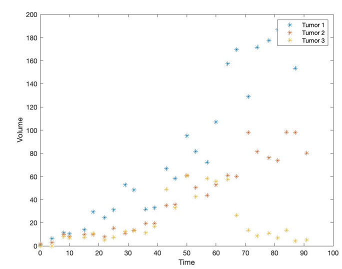

Figure 9: CM39 Graph

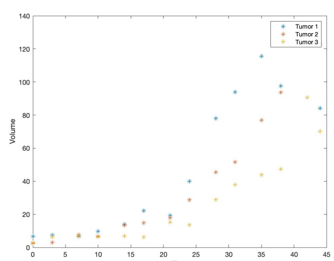

Figure 10: CM40 Graph

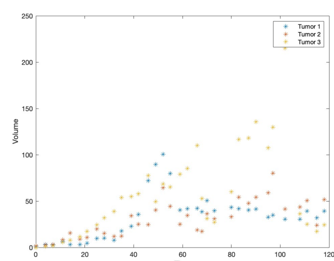

Figure 11: CM41 Graph

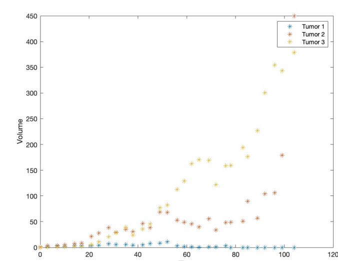

Figure 12: CM42 Graph

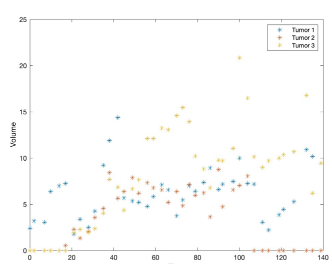

Figure 13: CM43 Graph

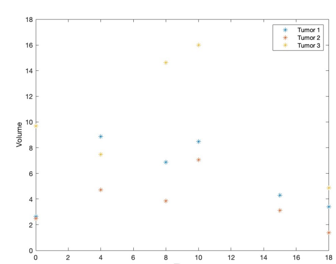

Figure 14: CM46 Graph

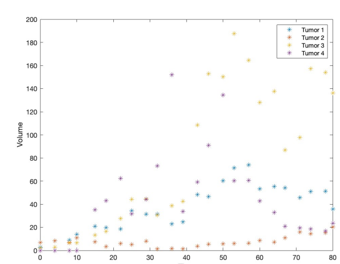

Figure 15: CM47 Graph

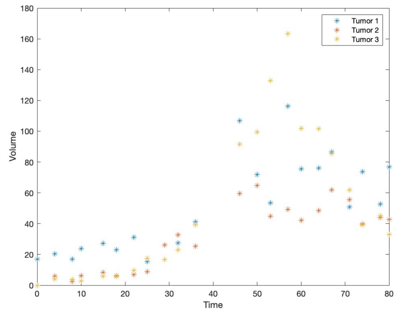

Figure 16: CM48 Graph

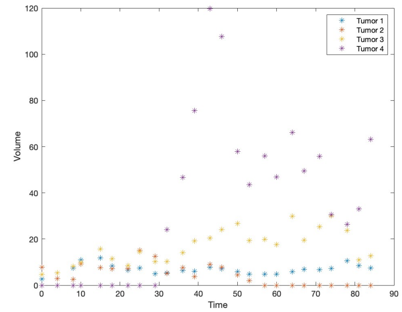

Figure 17: CM49 Graph

2.2.4 Graphing CM4

As stated above, the code for all of the CM4 graphs will be in the appendix (A.2) for cleanliness. The following graphs are the processed data of CM4:

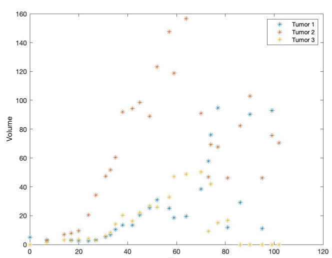

Figure 18: CM51 Graph

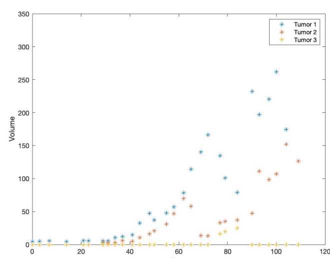

Figure 19: CM52 Graph

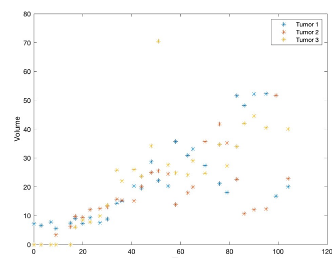

Figure 20: CM53 Graph

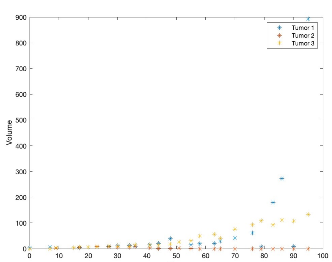

Figure 21: CM54 Graph

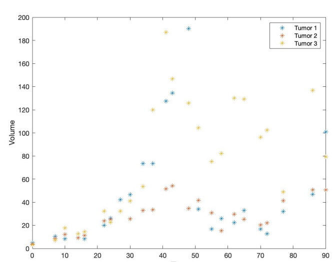

Figure 22: CM55 Graph

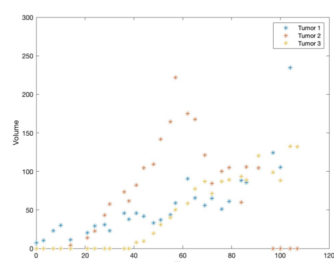

Figure 23: CM56 Graph

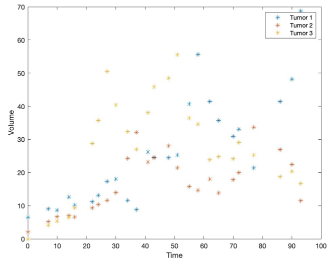

Figure 24: CM57 Graph

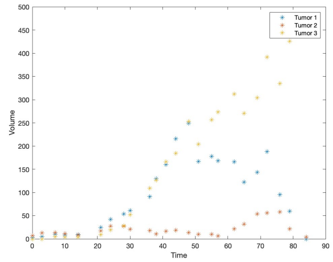

Figure 25: CM58 Graph

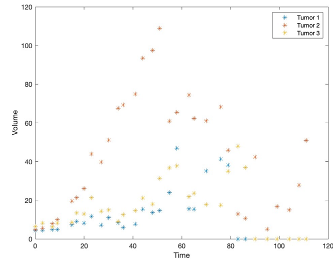

Figure 26: CM59 Graph

2.2.5 Graphing CM5

Similarly to CM3 and CM4, the code for all of CM5's graphs will be down in the appendix (A.2) for cleanliness. The following graphs are the processed data of CM5:

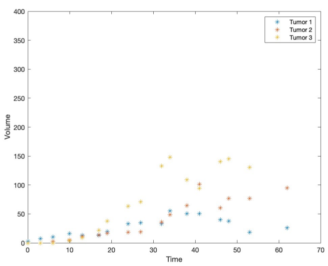

Figure 27: CM60 Graph

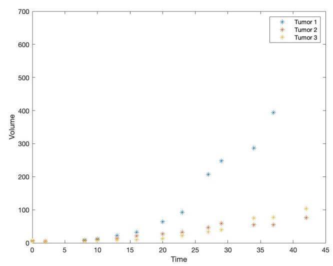

Figure 28: CM62 Graph

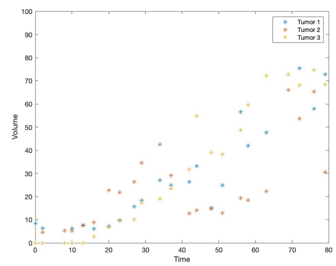

Figure 29: CM63 Graph

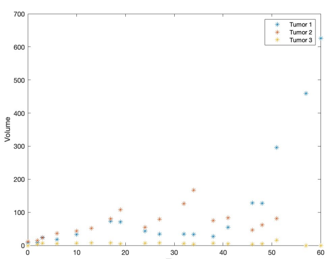

Figure 30: CM66 Graph

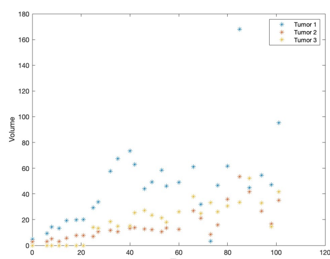

Figure 31: CM67 Graph

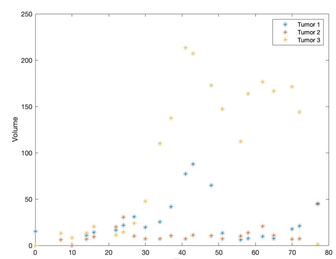

Figure 32: CM68 Graph

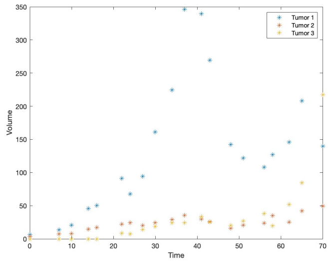

Figure 33: CM69 Graph

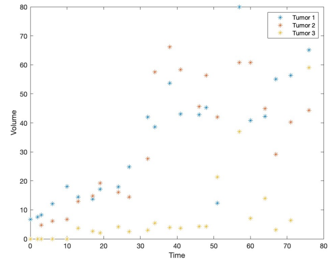

Figure 34: CM70 Graph

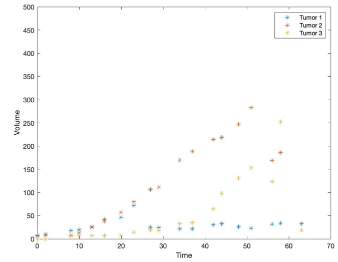

Figure 35: CM71 Graph

2.2.6 Graphing CM6

As previously stated, like the other CM groups, the code for CM6's graphs will be located in the appendix (A.2) for cleanliness. The following graphs are the processed data of CM6:

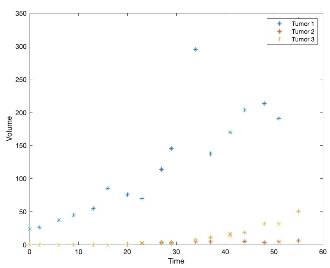

Figure 36: CM72 Graph

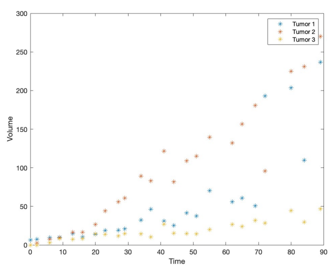

Figure 37: CM73 Graph

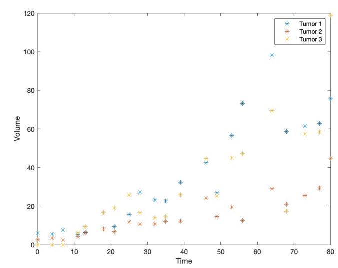

Figure 38: CM74 Graph

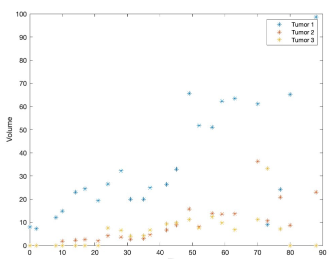

Figure 39: CM75 Graph

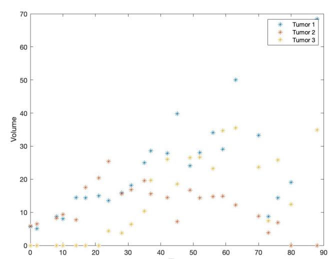

Figure 40: CM76 Graph

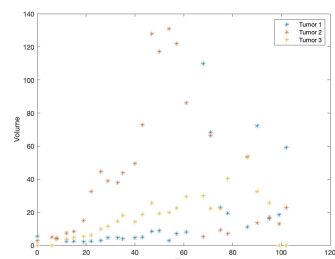

Figure 41: CM77 Graph

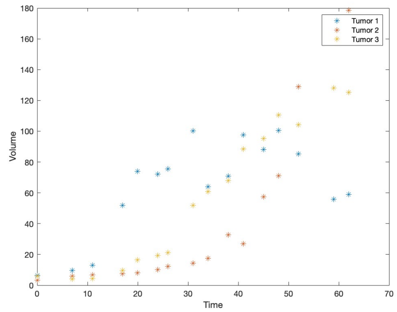

Figure 42: CM78 Graph

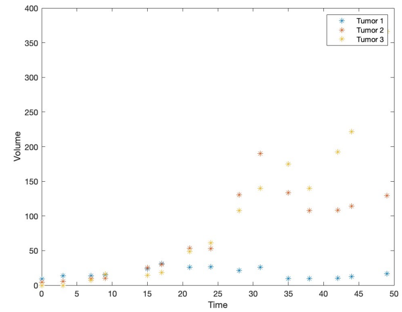

Figure 43: CM79 Graph

2.2.7 Graphing CM7

As for the final installment of the CM's I am happy to say that you can find the code for all of the graphs in CM7 down in the appendix (A.2)! The following graphs are the processed data of CM7:

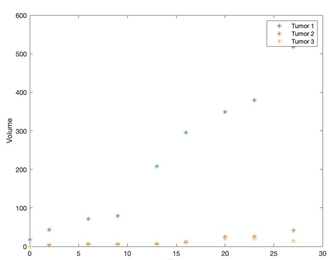

Figure 44: CM83 Graph

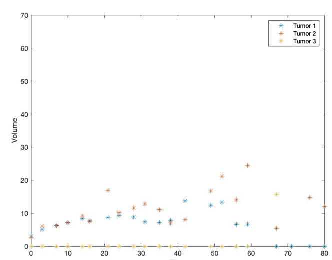

Figure 45: CM84 Graph

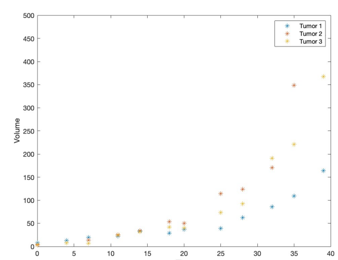

Figure 46: CM85 Graph

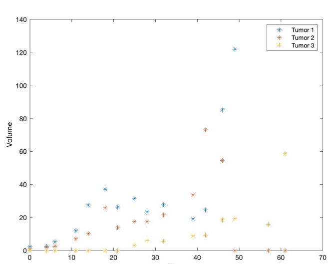

Figure 47: CM86 Graph

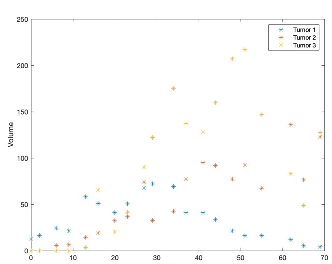

Figure 48: CM90 Graph

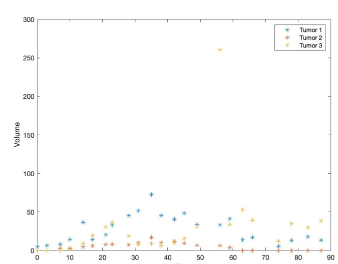

Figure 49: CM91 Graph

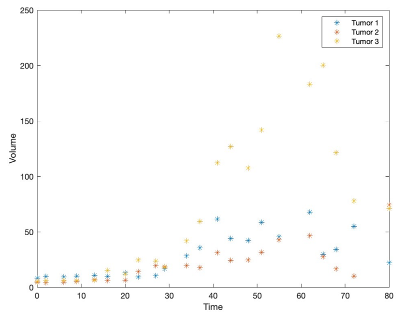

Figure 50: CM92 Graph

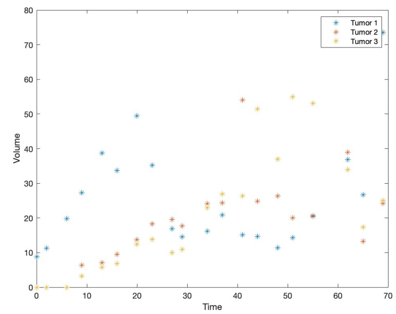

Figure 51: CM93 Graph

3 Part 3 Creating a Differential Equation

3.1 Chosen Data

For the final part of this project we decided to look at 10 of the mice in the full experiment. We chose CM41 and CM42 from group CM3, CM51 and CM54 from group CM4, CM63 and CM67 and CM70 from group CM5, CM74 from group CM6, and CM84 and CM86 from group CM7.

Our criteria for these specific mice was a delayed injection date and a long lifespan. With these requirements we believe it is possible to try to process how the injection is interacting with the tumors over the period of time.

3.2 Graphs

We have assembled the graphs for the 10 mice talked about above to make it easier to understand the correlation that we are looking for:

Note 1: The code for all of these graphs have a vertical line at the day that they first received their treatment.

Note 2: The MATLAB code for these graphs can be found in A3.

3.2.1 CM3 Mice

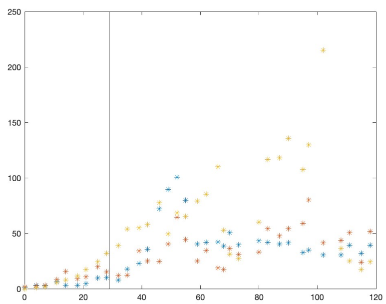

Figure 52: CM41 Injection Graph

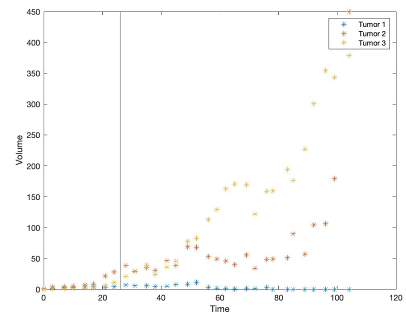

Figure 53: CM42 Injection Graph

3.2.2 CM4 Mice

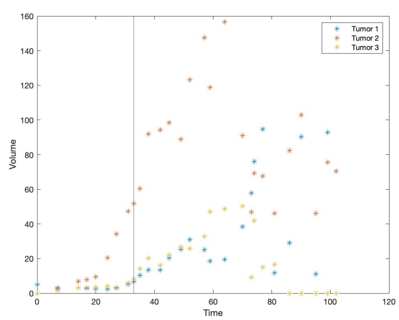

Figure 54: CM51 Injection Graph

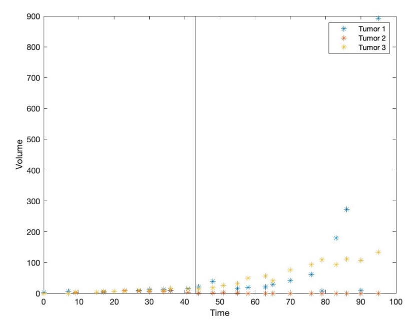

Figure 55: CM54 Injection Graph

3.2.3 CM5 Mice

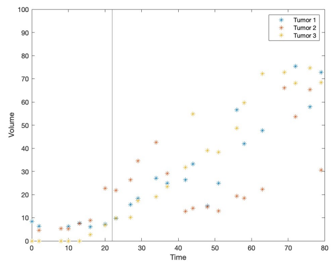

Figure 56: CM63 Inj. Graph

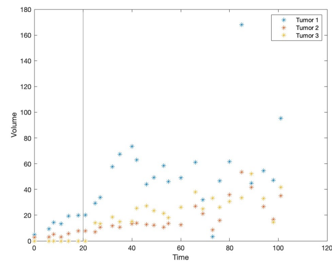

Figure 57: CM67 Inj. Graph

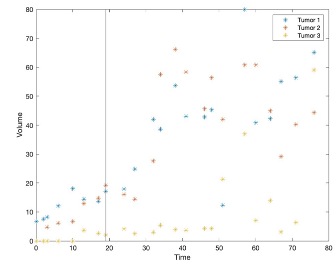

Figure 58: CM70 Inj. Graph

3.2.4 CM6 and CM7 Mice

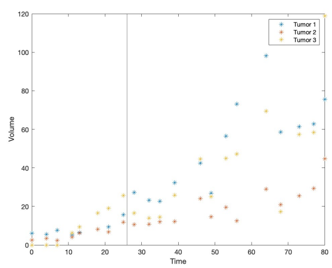

Figure 59: CM74 Inj. Graph

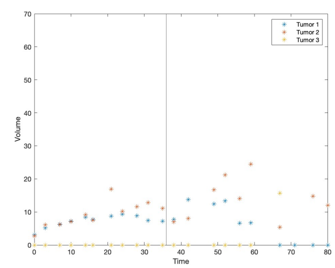

Figure 60: CM84 Inj. Graph

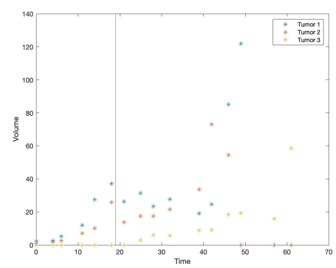

Figure 61: CM86 Inj. Graph

3.3 How to create an accurate model

Our final goal of this project is to produce a differential equation that accurately models the growth of one of the graphs in 3.2. We have chosen to create a model following the data for the mouse CM41. Our first idea to creating an accurate graph of the data was to have our function follow an exponential growth up until the first date of the injections.

3.4 Finding the best Logistic Curve to our Data

Interestingly, the Logistic Model follows this idea. While researching for a way to find out how to obtain the optimal values of a, b, and K for our Logistic model we found a file called "Fit Logistic Curve to a Data Set" [1]. This file uses three .m files which have been put in the appendix (A.3) for cleanliness.

The function takes in data points, for us these are volume and time, and finds the optimal values for a best fitting Logistic function. When running Logistic in the terminal a GUI opens and you must input a .txt Y-File and .txt X-File for the data and then press Plot Initial which plots the inputted data. Next, press Find Fit and it will calculate the best Logistic curve and give you the values for the actual Logistic Equation of the form [5]:

$$f(x) = \frac{K}{1 + e^{-a(x-x_0)}} \tag{7}$$

where K is our carrying capacity and a is our growth rate.

3.4.1 Logistic fitted Graphs for CM41 Tumors

We proceeded to apply this file to each of the three tumors of mouse CM41 up to the injection date and as well as applying it to all of the data of the three tumors simultaneously (Injection date started on day 29).

Here is a photo of the GUI and it's plotted curve for Tumor 1 of mouse CM41:

Note: The .txt files used are put in the appendix for cleanliness.

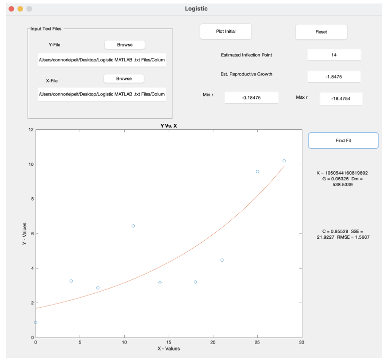

Figure 62: Logistic fit for CM41 Tumor 1

This model produces a growth rate of $a = 0.06326$ and a carrying capacity, $K = 1.050544160819892e+15$.

Here is a photo of the newly fitted graph for tumor 2 on the mouse CM41:

Note: The .txt files used are put in the appendix for cleanliness.

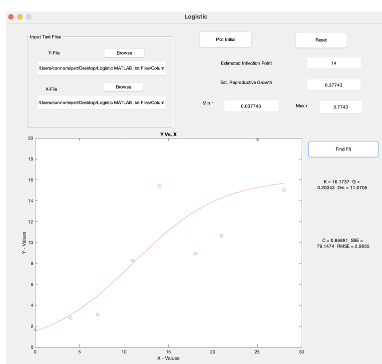

Figure 63: Logistic fit for CM41 Tumor 2

This model produces a growth rate of $a = 0.2034$ and a carrying capacity of $K = 16.173660538886249$.

For tumor 3 the GUI is as follows:

Note: The .txt files used are put in the appendix for cleanliness.

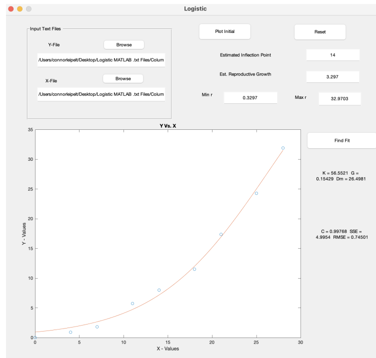

Figure 64: Logistic fit for CM41 Tumor 3

This model produces a growth rate of $a = 0.15429$ and a carrying capacity of $K = 56.552052808282419$.

For the last use of this file we decided to see the best logistic function for all of the data of all three tumors:

Note: The .txt files used are put in the appendix for cleanliness.

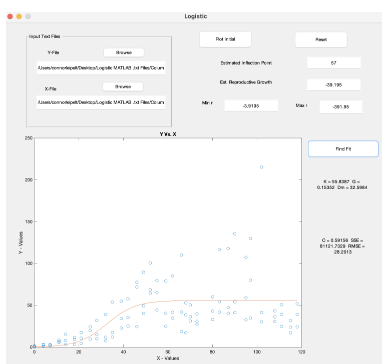

Figure 65: Logistic fit for all CM41 Tumor Data

This model produces a growth rate of $a = 0.15352$ and a carrying capacity, $K = 55.838668735004994$.

3.5 Fitted Differential Equation

3.5.1 Explanation of model

The final piece of this project was to create a differential equation that fits the growth that we are seeing in the data for CM41. We decided to use the parameters found for tumor 3 of the mouse as they fit well with the data. The process of creating a fitting differential equation starts with the exponential growth that we talked about earlier. Very soon after the date where the first injection happens (this point is chosen to be the value at which the data seems to become stable/is a returning point for the data after a period of time), the differential equation adds in another nonzero evaluating term. This is done by using a Heaviside function which activates at $V = 35mm^3$. This new term is a combination of two parts: the first part is a term that lessens the output of the function proportionally to the steepness of the graph at that point, and the second part is another decreasing term that is controlled by another Heaviside function which is activated whenever the volume of the tumor is large enough to counteract a value that is determined by the specific time at which it is evaluated (this second part of the new term becomes turned on indefinitely after $t = 120$ days).
The Heaviside function is a function that was created by Oliver Heaviside [4]. The case of the Heaviside that we are using is the piecewise definition of the function ($H(x)$ is how we denote the function):

$$H(x) := \left\{ \begin{array}{cc} 1, & x > 0 \\ 0, & x <= 0 \end{array} \right\} \tag{8}$$

A very important distinction that needs to be made is how we are able to use a derivative as a value in our differential equation. We used an iterative method that calculates the previous derivative at each point and assigns this new value to a variable that is used in the differential equation. Since this code pertains directly to the differential equation that we used to create our model we have inserted the code for this below:

```
clear vars; close all; format long;
% Looking at CM3: CM41
load('CM41Hvals.mat') % 34 days

CM41_H_M1= [];
CM41_V_M1 = [];
CM41_H_M2 = [];
CM41_V_M2 = [];
CM41_H_M3 = [];
CM41_V_M3 = [];

% CM41 Mean 34 days
for i = 1:34
    CM41_H_M1(i) = mean(CM41(1:3,i));
    CM41_V_M1(i) = mean(CM41(4:6,i));
    CM41_H_M2(i) = mean(CM41(7:9,i));
    CM41_V_M2(i) = mean(CM41(10:12,i));
    if i >= 2
        CM41_H_M3(i) = mean(CM41(13:15,i));
        CM41_V_M3(i) = mean(CM41(16:18,i));
    end
end
CM41_H_meanvals = [CM41_H_M1; CM41_H_M2; CM41_H_M3];
CM41_V_meanvals = [CM41_V_M1; CM41_V_M2; CM41_V_M3];
CM41_3rd_lengths = leng3(CM41_H_meanvals, CM41_V_meanvals);
CM41_vol_tumors = vol_solver(CM41_H_meanvals, CM41_V_meanvals,...
                             CM41_3rd_lengths);
```

```
CM41_time = [0 4 7 11 14 18 ...
             21 25 28 32 35 39 ...
             42 46 49 52 55 59 ...
             62 66 68 73 70 80 ...
             83 87 90 95 97 102 ...
             108 111 115 118];

figure
plot(CM41_time,CM41_vol_tumors(1,:),'*')
hold on;
plot(CM41_time,CM41_vol_tumors(2,:),'*')
hold on;
plot(CM41_time,CM41_vol_tumors(3,:),'*')
xline(29)
hold on;

% For Tumor 3 parameters
a = 0.15429;
K = 56.552052808282419;

% Time elapsed
t1 = 0:1:29;

func1 =
@(t1,x,xp) a.*x + (-1.7*abs(xp)-1.8*abs(xp).*heaviside(x+100/60*(t1-120))).*heaviside(x-35)

h1 = @(t,x) [a.*x(1).*(1-(x(1)/K))]
[t1 za] = ode45(h1,t1,0.90477); % For tumor 3
```

```
plot(t1,za(:),'o')
yline(35)
hold on
old_za = za(end);
deriv_za = h1(0,old_za);
dt = .1;

for i = 1:1500

    new_za = old_za+dt*func1(29+i*dt,old_za,deriv_za);
    deriv_za = (new_za-old_za)/dt;
    old_za = new_za;

    plot(29+i*dt, new_za, 'o')
end

xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

%%
% Function to create third diameter from horizontal and vertical
function third_length = leng3(H, V)
num_of_days = size(H,2);
num_of_tumors = size(H,1);
third_length = [];
for i = 1:num_of_tumors
```

```
    for j = 1:num_of_days
        leng3_val = (H(i,j) + V(i,j))/2;
        third_length(i,j) = leng3_val;
    end
end
end

% Function to compute volume of ellipsoid
function vol_tumors = vol_solver(H, V, T)
tumor_num = size(H,1);
day_num = size(H,2);
vol_tumors = [];
for i = 1:tumor_num
    for j = 1:day_num
        rad_H = H(i,j)/2;
        rad_V = V(i,j)/2;
        rad_T = T(i,j)/2;
        vol = (4/3)*pi*rad_H*rad_V*rad_T;
        vol_tumors(i,j) = vol;
    end
end
end
```

3.5.2 Graph of Model v Data

The output of this codes shows our differential equation model that fits the given data:

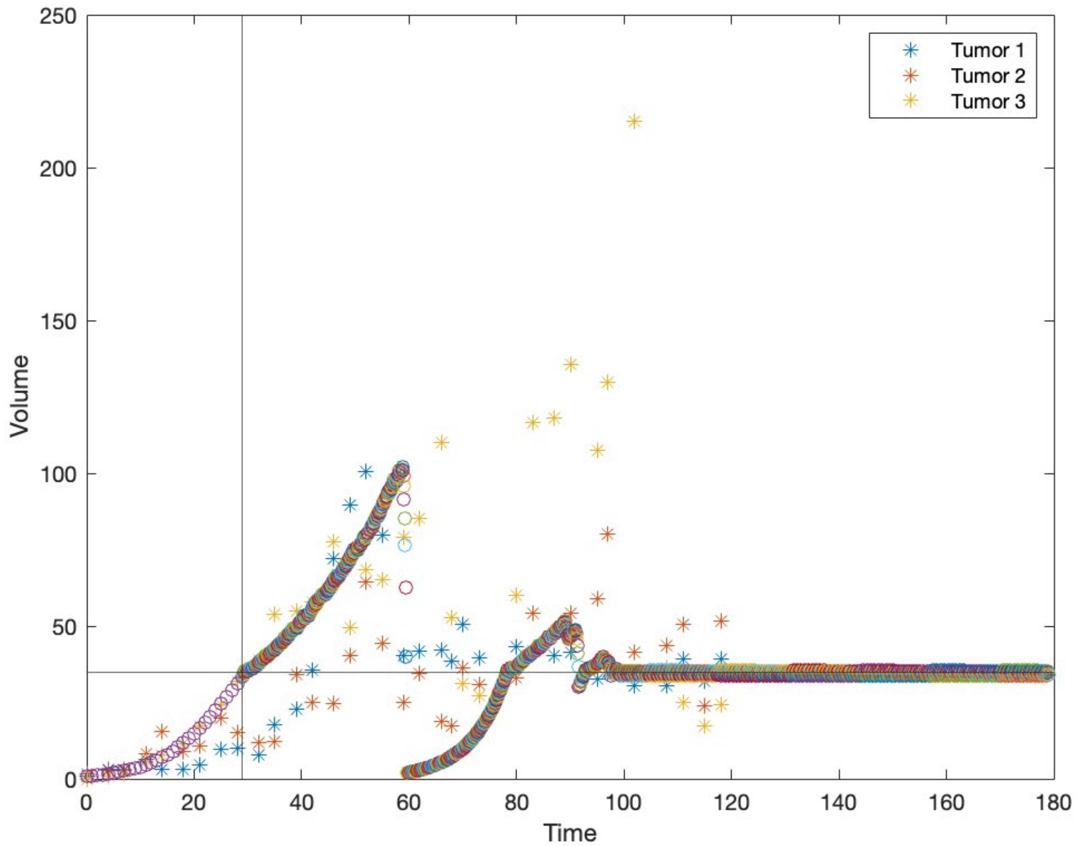

Figure 66: Differential Equation Model for CM41 Data

This differential equation models the beginning exponential growth then starts to show a slight decline in its growth. After reaching a certain height the model then drops down below our equilibrium value and starts to show growth of the tumor. This cycle happens once more, but since the drop from the second max point isn't as drastic, the model is sucked in the stable equilibrium and continues to stay at such values.

3.5.3 Possible Reasons for Behavior of Data

Possible reasons for this is the idea that the injection doesn't kick into effect until a few days after its first injection date. Coupled with this is the idea that the injection is slowing down the growth rate of the tumor until it is actually able to start the shrinking of such tumor. A possible explanation as to why the growth rate starts up again after the drop in volume is the idea that the injection needs an object to enact on and the new small size of the tumor is too small for the drug to affect it. From this, the idea is that once the tumor grows past the equilibrium point the injection interacts again with the tumor which is seen later in the model. The process of this shrinking and regrowing also shortens the range in which the tumor can grow freely. This shortening decreases the time for exponential growth to build up and allows the injections to keep the tumor at the designated equilibrium state after a period of time.

4 Summary/Reflection

This project has been very interesting and enjoyable. We were able to create three existing models for two chosen sets of data and see how well these models fit the given data. This allowed me to explore the uses of MATLAB and helped me to better understand how we can numerically approximate differential equations. I found the second portion of the project to be exciting as I was taking on a new challenge of interpreting raw data. I think this was very important as it has helped me to prepare for a career in research and pushed me to come up with ideas to make such information useful.The final part of this project allowed me to use my knowledge of theory and differential equations on real applications. The time in which we worked on getting a relatively good model let me experiment with ideas and different approaches to such a problem. I am truly thankful for having Dr. Lin as my advisor for this senior project and wanted to give my gratitude to her as she has been an essential part of this project and I could not have done it without her guidance and help. Thank you for this wonderful opportunity and I am excited to explore more of this in the future!

A Appendix

A.1 Code for Part 1

A.1.1 Logistic Code Data Set 1

```
clear vars; close all; format long;
% This file is for modeling growth of cancerous tumors over time using the
% logisitc model

% Data Table 1
Vol1 = [0.0158 0.0264 0.0326 0.0445 0.0646 0.0933 0.1454 0.2183 0.2842...
       0.4977 0.6033 0.8441 1.2163 1.4470 2.3298 2.5342 3.0064 3.4044...
       3.2046 4.5241 4.3459 5.1374 5.5376 4.8946 5.0660 6.1494 6.8548...
       5.9668 6.6945 6.6395 6.8971 7.2966 7.2268 6.8815 8.0993 7.2112...
       7.0694 7.4971 6.9974 6.7219 7.0523 7.1095 7.0694 8.0562 7.2268];
T1_time = [3.46 4.58 5.67 6.64 7.63 8.41 9.32 10.27 11.19...
           12.39 13.42 15.19 16.24 17.23 18.18 19.29 21.23 21.99...
           24.33 25.58 26.43 27.44 28.43 30.49 31.34 32.34 33.00...
           35.20 36.34 37.29 38.50 39.67 41.37 42.58 45.39 46.38...
           48.29 49.24 50.19 51.14 52.10 54.00 56.33 57.33 59.38];

%% Data Set 1

% Parameters
% Growth rate
a = 0.3389;
b = 0.0489;
% Carrying capacity
K = a/b;
% Time elapsed
% t = 0:1:60;

h1 = @(t,x) (a.*x.*(1-(x/K)));
[t1, za] = ode45(h1,T1_time,0.0158);
figure
plot(t1,za(:,1),'o')
hold on
plot(T1_time,Vol1,'*')
xlabel('Time')
ylabel('Volume')
legend('Logistic','Data 1')
% Calculating NMSE1
SSR_list1 = zeros(1,length(Vol1));
tumor_vlist1 = zeros(1,length(Vol1));
```

```
for i=1:45
    SSR1 = ((Vol1(i)) - za(i)).^2;
    SSR_list1(1,i) = SSR1;
    tumor_vsq1 = (za(i)).^2;                % Using estimates
    tumor_vlist1(1,i) = tumor_vsq1;
end
SSR_sum1 = sum(SSR_list1);
tumor_vsum1 = sum(tumor_vlist1);
NMSE1 = SSR_sum1/tumor_vsum1;
```

A.1.2 Logistic code Data Set 2

```
clear vars; close all; format long;
% This file is for modeling growth of cancerous tumors over time using the
% logisitc model
% Data table 2
Vol2 = [1.0410 4.3156 2.8523 4.1246 5.9634 8.2353...
        35.1929 32.3758 36.9683 62.4162 135.3351 137.0422...
        87.7625 98.7135 131.8011 147.4074 267.1758 421.1538...
        407.5250 303.3899 393.7301 337.5277 376.4903 395.8532];
T2_time = [0 3 7 10 14 17 ...
           21 24 28 31 35 38 ...
           42 45 49 52 56 59 ...
           62 65 69 72 76 78];

%% Data Set 2

% New Parameters
a = 0.1402;
b = 0.000357;
% t = 0:1:80;
K = a/b;

h2 = @(t,y) (a.*y.*(1-(y/K)));
[t2, ya] = ode45(h2,T2_time,1.0410);
figure
plot(t2,ya(:,1),'o')
hold on
plot(T2_time,Vol2,'*')
xlabel('Time')
ylabel('Volume')
legend('Logistic','Data 2')
% Calculating NMSE2
SSR_list2 = zeros(1,length(Vol2));
tumor_vlist2 = zeros(1,length(Vol2));
```

```
% true_sol_list2 = [];
for i=1:length(Vol2)
    SSR2 = ((Vol2(i)) - ya(i)).^2;
    SSR_list2(1,i) = SSR2;
    tumor_vsq2 = (ya(i)).^2;                % Using estimates
    tumor_vlist2(1,i) = tumor_vsq2;
end
SSR_sum2 = sum(SSR_list2);
tumor_vsum2 = sum(tumor_vlist2);
NMSE2 = SSR_sum2/tumor_vsum2;
```

A.1.3 Bertalanffy Code Data Set 1

```
clear vars; close all; format long;
% This file is for modeling growth of cancerous tumors over time using the
% Bertalanffy model

% Data Table 1
Vol1 = [0.0158 0.0264 0.0326 0.0445 0.0646 0.0933 0.1454 0.2183 0.2842...
       0.4977 0.6033 0.8441 1.2163 1.4470 2.3298 2.5342 3.0064 3.4044...
       3.2046 4.5241 4.3459 5.1374 5.5376 4.8946 5.0660 6.1494 6.8548...
       5.9668 6.6945 6.6395 6.8971 7.2966 7.2268 6.8815 8.0993 7.2112...
       7.0694 7.4971 6.9974 6.7219 7.0523 7.1095 7.0694 8.0562 7.2268];
T1_time = [3.46 4.58 5.67 6.64 7.63 8.41 9.32 10.27 11.19...
           12.39 13.42 15.19 16.24 17.23 18.18 19.29 21.23 21.99...
           24.33 25.58 26.43 27.44 28.43 30.49 31.34 32.34 33.00...
           35.20 36.34 37.29 38.50 39.67 41.37 42.58 45.39 46.38...
           48.29 49.24 50.19 51.14 52.10 54.00 56.33 57.33 59.38];

%% Data Set 1
h1 = @(t,x) ((a.*(x.^(2/3))-b.*x));
[t1, za] = ode45(h1,T1_time,0.0158);
figure
plot(t1,za(:,1),'o')
hold on
plot(T1_time,Vol1,'*')
xlabel('Time')
ylabel('Volume')
legend('Bertalanffy','Data 1')
% Calculating NMSE1
SSR_list1 = zeros(1,length(Vol1));
tumor_vlist1 = zeros(1,length(Vol1));
for i=1:45
    SSR1 = ((Vol1(i)) - za(i)).^2;
    SSR_list1(1,i) = SSR1;
```

```
    tumor_vsq1 = (za(i)).^2;                % Using estimates
    tumor_vlist1(1,i) = tumor_vsq1;
end
SSR_sum1 = sum(SSR_list1);
tumor_vsum1 = sum(tumor_vlist1);
NMSE1 = SSR_sum1/tumor_vsum1;
% Growth rate
a = 0.4340;
b = 0.2158;

% Carrying capacity
% K = a/b;

% Time elapsed
% t = 0:1:60;
```

A.1.4 Bertalanffy Code Data Set 2

```
clear vars; close all; format long;
% This file is for modeling growth of cancerous tumors over time using the
% Bertalanffy model

% Data table 2
Vol2 = [1.0410 4.3156 2.8523 4.1246 5.9634 8.2353...
        35.1929 32.3758 36.9683 62.4162 135.3351 137.0422...
        87.7625 98.7135 131.8011 147.4074 267.1758 421.1538...
        407.5250 303.3899 393.7301 337.5277 376.4903 395.8532];
T2_time = [0 3 7 10 14 17 ...
           21 24 28 31 35 38 ...
           42 45 49 52 56 59 ...
           62 65 69 72 76 78];

%% Data Set 2

% New Parameters
a = 0.4947;
b = 0.0465;
% t = 0:1:80;
% K = a/b;

h2 = @(t,y) ((a.*(y.^(2/3))-b.*y));
[t2, ya] = ode45(h2,T2_time,1.0410);
figure
plot(t2,ya(:,1),'o')
hold on
```

```
plot(T2_time,Vol2,'*')
xlabel('Time')
ylabel('Volume')
legend('Bertalanffy','Data 2')
% Calculating NMSE2
SSR_list2 = zeros(1,length(Vol2));
tumor_vlist2 = zeros(1,length(Vol2));
for i=1:24
    SSR2 = ((Vol2(i)) - ya(i)).^2;
    SSR_list2(1,i) = SSR2;
    tumor_vsq2 = (ya(i)).^2;                % Using estimates
    tumor_vlist2(1,i) = tumor_vsq2;
end
SSR_sum2 = sum(SSR_list2);
tumor_vsum2 = sum(tumor_vlist2);
NMSE2 = SSR_sum2/tumor_vsum2;
```

A.1.5 Gompertz Code Data Set 1

```
clear vars; close all; format long;
% This file is for modeling growth of cancerous tumors over time using the
% Gompertz model

% Data Table 1
Vol1 = [0.0158 0.0264 0.0326 0.0445 0.0646 0.0933 0.1454 0.2183 0.2842...
       0.4977 0.6033 0.8441 1.2163 1.4470 2.3298 2.5342 3.0064 3.4044...
       3.2046 4.5241 4.3459 5.1374 5.5376 4.8946 5.0660 6.1494 6.8548...
       5.9668 6.6945 6.6395 6.8971 7.2966 7.2268 6.8815 8.0993 7.2112...
       7.0694 7.4971 6.9974 6.7219 7.0523 7.1095 7.0694 8.0562 7.2268];
T1_time = [3.46 4.58 5.67 6.64 7.63 8.41 9.32 10.27 11.19...
           12.39 13.42 15.19 16.24 17.23 18.18 19.29 21.23 21.99...
           24.33 25.58 26.43 27.44 28.43 30.49 31.34 32.34 33.00...
           35.20 36.34 37.29 38.50 39.67 41.37 42.58 45.39 46.38...
           48.29 49.24 50.19 51.14 52.10 54.00 56.33 57.33 59.38];

% Growth rate
a = 0.2375;
b = 0.1179;

% Carrying capacity
% K = a/b;

% Time elapsed
% t = 0:1:60;
```

```
%% Data Set 1
h1 = @(t,x) (x.*(a-b.*log(x)));
[t1, za] = ode45(h1,T1_time,0.0158);
figure
plot(t1,za(:,1),'o')
hold on
plot(T1_time,Vol1,'*')
xlabel('Time')
ylabel('Volume')
legend('Gompertz','Data 1')
% Calculating NMSE1
SSR_list1 = zeros(1,length(Vol1));
tumor_vlist1 = zeros(1,length(Vol2));
for i=1:45
    SSR1 = ((Vol1(i)) - za(i)).^2;
    SSR_list1(1,i) = SSR1;
    tumor_vsq1 = (za(i)).^2;                % Using estimates
    tumor_vlist1(1,i) = tumor_vsq1;
end
SSR_sum1 = sum(SSR_list1);
tumor_vsum1 = sum(tumor_vlist1);
NMSE1 = SSR_sum1/tumor_vsum1;
```

A.1.6 Gompertz Code Data Set 2

```
clear vars; close all; format long;
% This file is for modeling growth of cancerous tumors over time using the
% Gompertz model

% Data table 2
Vol2 = [1.0410 4.3156 2.8523 4.1246 5.9634 8.2353...
        35.1929 32.3758 36.9683 62.4162 135.3351 137.0422...
        87.7625 98.7135 131.8011 147.4074 267.1758 421.1538...
        407.5250 303.3899 393.7301 337.5277 376.4903 395.8532];
T2_time = [0 3 7 10 14 17 ...
           21 24 28 31 35 38 ...
           42 45 49 52 56 59 ...
           62 65 69 72 76 78];

%% Data Set 2

% New Parameters
a = 0.2619;
b = 0.0413;
% t = 0:1:80;
```

```
% K = a/b;

h2 = @(t,y) (y.*(a-b.*log(y)));
[t2, ya] = ode45(h2,T2_time,1.0410);
figure
plot(t2,ya(:,1),'o')
hold on
plot(T2_time,Vol2,'*')
xlabel('Time')
ylabel('Volume')
legend('Gompertz','Data 2')
% Calculating NMSE2
SSR_list2 = zeros(1,length(Vol2));
tumor_vlist2 = zeros(1,length(Vol2));
for i=1:24
    SSR2 = ((Vol2(i)) - ya(i)).^2;
    SSR_list2(1,i) = SSR2;
    tumor_vsq2 = (ya(i)).^2;                % Using estimates
    tumor_vlist2(1,i) = tumor_vsq2;
end
SSR_sum2 = sum(SSR_list2);
tumor_vsum2 = sum(tumor_vlist2);
NMSE2 = SSR_sum2/tumor_vsum2;
```

A.2 Code for Part 2

A.2.1 Code for CM3

```
clear vars; close all; format long;
% Script to import excel data
% Note: No data for CM.22, CM.33, CM.45
% CM.22 and CM.33 were in control group
% CM.45 died of natural causes (no data given)
load('CM37Hvals.mat') % 24 days * Yellow Death
load('CM38Hvals.mat') % 23 days * Yellow Death
load('CM39Hvals.mat') % 27 days * Yellow Death
load('CM40Hvals.mat') % 14 days * Yellow Death
load('CM41Hvals.mat') % 34 days
load('CM42Hvals.mat') % 31 days * Yellow Death
load('CM43Hvals.mat') % 40 days
load('CM46Hvals.mat') % 06 days * Red Death
load('CM47Hvals.mat') % 24 days *4 tumors * Yellow Death
load('CM48Hvals.mat') % 22 days * Yellow Death
load('CM49Hvals.mat') % 25 days *4 tumors * Yellow Death

CM37_H_M1= [];
```

```
CM37_V_M1 = [];
CM37_H_M2 = [];
CM37_V_M2 = [];
CM37_H_M3 = [];
CM37_V_M3 = [];

CM38_H_M1= [];
CM38_V_M1 = [];
CM38_H_M2 = [];
CM38_V_M2 = [];
CM38_H_M3 = [];
CM38_V_M3 = [];

CM39_H_M1= [];
CM39_V_M1 = [];
CM39_H_M2 = [];
CM39_V_M2 = [];
CM39_H_M3 = [];
CM39_V_M3 = [];

CM40_H_M1= [];
CM40_V_M1 = [];
CM40_H_M2 = [];
CM40_V_M2 = [];
CM40_H_M3 = [];
CM40_V_M3 = [];

CM41_H_M1= [];
CM41_V_M1 = [];
CM41_H_M2 = [];
CM41_V_M2 = [];
CM41_H_M3 = [];
CM41_V_M3 = [];

CM42_H_M1= [];
CM42_V_M1 = [];
CM42_H_M2 = [];
CM42_V_M2 = [];
CM42_H_M3 = [];
CM42_V_M3 = [];

CM43_H_M1= [];
CM43_V_M1 = [];
CM43_H_M2 = [];
CM43_V_M2 = [];
```

```
CM43_H_M3 = [];
CM43_V_M3 = [];

CM46_H_M1= [];
CM46_V_M1 = [];
CM46_H_M2 = [];
CM46_V_M2 = [];
CM46_H_M3 = [];
CM46_V_M3 = [];

CM47_H_M1= [];
CM47_V_M1 = [];
CM47_H_M2 = [];
CM47_V_M2 = [];
CM47_H_M3 = [];
CM47_V_M3 = [];
CM47_H_M4 = [];
CM47_V_M4 = [];

CM48_H_M1= [];
CM48_V_M1 = [];
CM48_H_M2 = [];
CM48_V_M2 = [];
CM48_H_M3 = [];
CM48_V_M3 = [];

CM49_H_M1= [];
CM49_V_M1 = [];
CM49_H_M2 = [];
CM49_V_M2 = [];
CM49_H_M3 = [];
CM49_V_M3 = [];
CM49_H_M4 = [];
CM49_V_M4 = [];

% CM37 Mean 24 days * Yellow Death
for i = 1:24
    CM37_H_M1(i) = mean(CM37(1:3,i));
    CM37_V_M1(i) = mean(CM37(4:6,i));
    CM37_H_M2(i) = mean(CM37(7:9,i));
    CM37_V_M2(i) = mean(CM37(10:12,i));
    if i >= 3
        CM37_H_M3(i) = mean(CM37(13:15,i));
        CM37_V_M3(i) = mean(CM37(16:18,i));
    end
```

```
end
CM37_H_meanvals = [CM37_H_M1; CM37_H_M2; CM37_H_M3];
CM37_V_meanvals = [CM37_V_M1; CM37_V_M2; CM37_V_M3];
CM37_3rd_lengths = leng3(CM37_H_meanvals, CM37_V_meanvals);
CM37_vol_tumors = vol_solver(CM37_H_meanvals, CM37_V_meanvals,...
                             CM37_3rd_lengths);
CM37_time = [0 3 7 10 14 17 ...
             21 24 28 31 35 38 ...
             42 45 49 52 56 59 ...
             62 65 69 72 76 78];
figure
plot(CM37_time,CM37_vol_tumors(1,:),'*')
hold on;
plot(CM37_time,CM37_vol_tumors(2,:),'*')
hold on;
plot(CM37_time,CM37_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM38 Mean 23 days * Yellow Death
for i = 1:23
    CM38_H_M1(i) = mean(CM38(1:3,i));
    CM38_V_M1(i) = mean(CM38(4:6,i));
    if i >= 3
        CM38_H_M2(i) = mean(CM38(7:9,i));
        CM38_V_M2(i) = mean(CM38(10:12,i));
        CM38_H_M3(i) = mean(CM38(13:15,i));
        CM38_V_M3(i) = mean(CM38(16:18,i));
    end
end
CM38_H_meanvals = [CM38_H_M1; CM38_H_M2; CM38_H_M3];
CM38_V_meanvals = [CM38_V_M1; CM38_V_M2; CM38_V_M3];
CM38_3rd_lengths = leng3(CM38_H_meanvals, CM38_V_meanvals);
CM38_vol_tumors = vol_solver(CM38_H_meanvals, CM38_V_meanvals,...
                             CM38_3rd_lengths);
CM38_time = [0 3 7 10 14 17 ...
             21 24 28 31 35 38 ...
             42 45 49 52 56 59 ...
             63 66 69 72 76];
figure
plot(CM38_time,CM38_vol_tumors(1,:),'*')
hold on;
plot(CM38_time,CM38_vol_tumors(2,:),'*')
hold on;
```

```
plot(CM38_time,CM38_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM39 Mean 27 days * Yellow Death
for i = 1:27
    CM39_H_M1(i) = mean(CM39(1:3,i));
    CM39_V_M1(i) = mean(CM39(4:6,i));
    CM39_H_M2(i) = mean(CM39(7:9,i));
    CM39_V_M2(i) = mean(CM39(10:12,i));
    if i >= 3
        CM39_H_M3(i) = mean(CM39(13:15,i));
        CM39_V_M3(i) = mean(CM39(16:18,i));
    end
end
CM39_H_meanvals = [CM39_H_M1; CM39_H_M2; CM39_H_M3];
CM39_V_meanvals = [CM39_V_M1; CM39_V_M2; CM39_V_M3];
CM39_3rd_lengths = leng3(CM39_H_meanvals, CM39_V_meanvals);
CM39_vol_tumors = vol_solver(CM39_H_meanvals, CM39_V_meanvals,...
                             CM39_3rd_lengths);
CM39_time = [0 4 8 10 15 18 22 25 29 ...
        32 36 39 43 46 50 53 57 60 ...
        64 67 71 74 78 81 84 87 91];
figure
plot(CM39_time,CM39_vol_tumors(1,:),'*')
hold on;
plot(CM39_time,CM39_vol_tumors(2,:),'*')
hold on;
plot(CM39_time,CM39_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM40 Mean 14 days * Yellow Death
for i = 1:14
    CM40_H_M1(i) = mean(CM40(1:3,i));
    CM40_V_M1(i) = mean(CM40(4:6,i));
    CM40_H_M2(i) = mean(CM40(7:9,i));
    CM40_V_M2(i) = mean(CM40(10:12,i));
    CM40_H_M3(i) = mean(CM40(13:15,i));
    CM40_V_M3(i) = mean(CM40(16:18,i));
end
CM40_H_meanvals = [CM40_H_M1; CM40_H_M2; CM40_H_M3];
CM40_V_meanvals = [CM40_V_M1; CM40_V_M2; CM40_V_M3];
```

```
CM40_3rd_lengths = leng3(CM40_H_meanvals, CM40_V_meanvals);
CM40_vol_tumors = vol_solver(CM40_H_meanvals, CM40_V_meanvals,...
                             CM40_3rd_lengths);
CM40_time = [0 3 7 10 14 17 21 ...
             24 28 31 35 38 42 44];
figure
plot(CM40_time,CM40_vol_tumors(1,:),'*')
hold on;
plot(CM40_time,CM40_vol_tumors(2,:),'*')
hold on;
plot(CM40_time,CM40_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM41 Mean 34 days
for i = 1:34
    CM41_H_M1(i) = mean(CM41(1:3,i));
    CM41_V_M1(i) = mean(CM41(4:6,i));
    CM41_H_M2(i) = mean(CM41(7:9,i));
    CM41_V_M2(i) = mean(CM41(10:12,i));
    if i >= 2
        CM41_H_M3(i) = mean(CM41(13:15,i));
        CM41_V_M3(i) = mean(CM41(16:18,i));
    end
end
CM41_H_meanvals = [CM41_H_M1; CM41_H_M2; CM41_H_M3];
CM41_V_meanvals = [CM41_V_M1; CM41_V_M2; CM41_V_M3];
CM41_3rd_lengths = leng3(CM41_H_meanvals, CM41_V_meanvals);
CM41_vol_tumors = vol_solver(CM41_H_meanvals, CM41_V_meanvals,...
                             CM41_3rd_lengths);
CM41_time = [0 4 7 11 14 18 ...
             21 25 28 32 35 39 ...
             42 46 49 52 55 59 ...
             62 66 68 73 70 80 ...
             83 87 90 95 97 102 ...
             108 111 115 118];
figure
plot(CM41_time,CM41_vol_tumors(1,:),'*')
hold on;
plot(CM41_time,CM41_vol_tumors(2,:),'*')
hold on;
plot(CM41_time,CM41_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
```

```
legend('Tumor 1','Tumor 2','Tumor 3')

% CM42 Mean 31 days * Yellow Death
for i = 1:31
    if i<=23
        CM42_H_M1(i) = mean(CM42(1:3,i));
        CM42_V_M1(i) = mean(CM42(4:6,i));
    else
        CM42_H_M1(i) = 0;
        CM42_V_M1(i) = 0;
    end
    CM42_H_M2(i) = mean(CM42(7:9,i));
    CM42_V_M2(i) = mean(CM42(10:12,i));
    if i >= 2
        CM42_H_M3(i) = mean(CM42(13:15,i));
        CM42_V_M3(i) = mean(CM42(16:18,i));
    end
end
CM42_H_meanvals = [CM42_H_M1; CM42_H_M2; CM42_H_M3];
CM42_V_meanvals = [CM42_V_M1; CM42_V_M2; CM42_V_M3];
CM42_3rd_lengths = leng3(CM42_H_meanvals, CM42_V_meanvals);
CM42_vol_tumors = vol_solver(CM42_H_meanvals, CM42_V_meanvals,...
                             CM42_3rd_lengths);
CM42_time = [0 3 7 10 14 17 ...
             21 24 28 31 35 38 ...
             42 45 49 52 56 59 ...
             62 65 69 72 76 78 ...
             83 85 89 92 96 99 104];
figure
plot(CM42_time,CM42_vol_tumors(1,:),'*')
hold on;
plot(CM42_time,CM42_vol_tumors(2,:),'*')
hold on;
plot(CM42_time,CM42_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM43 Mean 40 days
for i = 1:40
    if i<=39
        CM43_H_M1(i) = mean(CM43(1:3,i));
        CM43_V_M1(i) = mean(CM43(4:6,i));
    else
        CM43_H_M1(i) = 0;
```

```
        CM43_V_M1(i) = 0;
    end
    if i >= 6 && i <= 31
        CM43_H_M2(i) = mean(CM43(7:9,i));
        CM43_V_M2(i) = mean(CM43(10:12,i));
    elseif i >=32
        CM43_H_M2(i) = 0;
        CM43_V_M2(i) = 0;
    end
    if i >= 7
        CM43_H_M3(i) = mean(CM43(13:15,i));
        CM43_V_M3(i) = mean(CM43(16:18,i));
    end
end
CM43_H_meanvals = [CM43_H_M1; CM43_H_M2; CM43_H_M3];
CM43_V_meanvals = [CM43_V_M1; CM43_V_M2; CM43_V_M3];
CM43_3rd_lengths = leng3(CM43_H_meanvals, CM43_V_meanvals);
CM43_vol_tumors = vol_solver(CM43_H_meanvals, CM43_V_meanvals,...
                             CM43_3rd_lengths);
CM43_time = [0 2 7 10 14 ...
             17 21 24 28 31 ...
             35 38 42 45 49 ...
             52 56 59 63 66 ...
             70 73 76 79 83 ...
             86 90 92 97 100 ...
             104 107 111 114 119 ...
             121 126 132 135 139];
figure
plot(CM43_time,CM43_vol_tumors(1,:),'*')
hold on;
plot(CM43_time,CM43_vol_tumors(2,:),'*')
hold on;
plot(CM43_time,CM43_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM46 Mean 06 days *rat died naturally (Red Death)
for i = 1:6
    CM46_H_M1(i) = mean(CM46(1:3,i));
    CM46_V_M1(i) = mean(CM46(4:6,i));
    CM46_H_M2(i) = mean(CM46(7:9,i));
    CM46_V_M2(i) = mean(CM46(10:12,i));
    CM46_H_M3(i) = mean(CM46(13:15,i));
    CM46_V_M3(i) = mean(CM46(16:18,i));
```

```
end
CM46_H_meanvals = [CM46_H_M1; CM46_H_M2; CM46_H_M3];
CM46_V_meanvals = [CM46_V_M1; CM46_V_M2; CM46_V_M3];
CM46_3rd_lengths = leng3(CM46_H_meanvals, CM46_V_meanvals);
CM46_vol_tumors = vol_solver(CM46_H_meanvals, CM46_V_meanvals,...
                             CM46_3rd_lengths);
CM46_time = [0 4 8 10 15 18];
figure
plot(CM46_time,CM46_vol_tumors(1,:),'*')
hold on;
plot(CM46_time,CM46_vol_tumors(2,:),'*')
hold on;
plot(CM46_time,CM46_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM47 Mean 24 days *4 tumors * Yellow Death
for i = 1:24
    CM47_H_M1(i) = mean(CM47(1:3,i));
    CM47_V_M1(i) = mean(CM47(4:6,i));
    CM47_H_M2(i) = mean(CM47(7:9,i));
    CM47_V_M2(i) = mean(CM47(10:12,i));
    CM47_H_M3(i) = mean(CM47(13:15,i));
    CM47_V_M3(i) = mean(CM47(16:18,i));
    if i >= 5
        CM47_H_M4(i) = mean(CM47(19:21,i));
        CM47_V_M4(i) = mean(CM47(22:24,i));
    end
end
CM47_H_meanvals = [CM47_H_M1; CM47_H_M2; CM47_H_M3; CM47_H_M4];
CM47_V_meanvals = [CM47_V_M1; CM47_V_M2; CM47_V_M3; CM47_V_M4];
CM47_3rd_lengths = leng3(CM47_H_meanvals, CM47_V_meanvals);
CM47_vol_tumors = vol_solver(CM47_H_meanvals, CM47_V_meanvals,...
                             CM47_3rd_lengths);
CM47_time = [0 4 8 10 15 18 ...
             22 25 29 32 36 39 ...
             43 46 50 53 57 60 ...
             64 67 71 74 78 80];
figure
plot(CM47_time,CM47_vol_tumors(1,:),'*')
hold on;
plot(CM47_time,CM47_vol_tumors(2,:),'*')
hold on;
plot(CM47_time,CM47_vol_tumors(3,:),'*')
```

```
hold on;
plot(CM47_time,CM47_vol_tumors(4,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3','Tumor 4')

% CM48 Mean 22 days * Yellow Death
for i = 1:22
    CM48_H_M1(i) = mean(CM48(1:3,i));
    CM48_V_M1(i) = mean(CM48(4:6,i));
    if i>= 2
        CM48_H_M2(i) = mean(CM48(7:9,i));
        CM48_V_M2(i) = mean(CM48(10:12,i));
        CM48_H_M3(i) = mean(CM48(13:15,i));
        CM48_V_M3(i) = mean(CM48(16:18,i));
    end
end
CM48_H_meanvals = [CM48_H_M1; CM48_H_M2; CM48_H_M3];
CM48_V_meanvals = [CM48_V_M1; CM48_V_M2; CM48_V_M3];
CM48_3rd_lengths = leng3(CM48_H_meanvals, CM48_V_meanvals);
CM48_vol_tumors = vol_solver(CM48_H_meanvals, CM48_V_meanvals,...
                             CM48_3rd_lengths);
CM48_time = [0 4 8 10 15 18 ...
             22 25 29 32 36 ...
             46 50 53 57 60 ...
             64 67 71 74 78 80];
figure
plot(CM48_time,CM48_vol_tumors(1,:),'*')
hold on;
plot(CM48_time,CM48_vol_tumors(2,:),'*')
hold on;
plot(CM48_time,CM48_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM49 Mean 25 days *4 tumors * Yellow Death
for i = 1:25
    CM49_H_M1(i) = mean(CM49(1:3,i));
    CM49_V_M1(i) = mean(CM49(4:6,i));
    if i <= 16
        CM49_H_M2(i) = mean(CM49(7:9,i));
        CM49_V_M2(i) = mean(CM49(10:12,i));
    else
        CM49_H_M2(i) = 0;
```

```
        CM49_V_M2(i) = 0;
    end
    CM49_H_M3(i) = mean(CM49(13:15,i));
    CM49_V_M3(i) = mean(CM49(16:18,i));
    if i >= 10
        CM49_H_M4(i) = mean(CM49(19:21,i));
        CM49_V_M4(i) = mean(CM49(22:24,i));
    end
end
CM49_H_meanvals = [CM49_H_M1; CM49_H_M2; CM49_H_M3; CM49_H_M4];
CM49_V_meanvals = [CM49_V_M1; CM49_V_M2; CM49_V_M3; CM49_V_M4];
CM49_3rd_lengths = leng3(CM49_H_meanvals, CM49_V_meanvals);
CM49_vol_tumors = vol_solver(CM49_H_meanvals, CM49_V_meanvals,...
                             CM49_3rd_lengths);
CM49_time = [0 4 8 10 15 18 ...
             22 25 29 32 36 39 ...
             43 46 50 53 57 60 ...
             64 67 71 74 78 81 84];
figure
plot(CM49_time,CM49_vol_tumors(1,:),'*')
hold on;
plot(CM49_time,CM49_vol_tumors(2,:),'*')
hold on;
plot(CM49_time,CM49_vol_tumors(3,:),'*')
hold on;
plot(CM49_time,CM49_vol_tumors(4,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3','Tumor 4')


%%
% Function to create third diameter from horizontal and vertical
function third_length = leng3(H, V)
num_of_days = size(H,2);
num_of_tumors = size(H,1);
third_length = [];
for i = 1:num_of_tumors
    for j = 1:num_of_days
        leng3_val = (H(i,j) + V(i,j))/2;
        third_length(i,j) = leng3_val;
    end
end
end
```

```
% Function to compute volume of ellipsoid
function vol_tumors = vol_solver(H, V, T)
tumor_num = size(H,1);
day_num = size(H,2);
vol_tumors = [];
for i = 1:tumor_num
    for j = 1:day_num
        rad_H = H(i,j)/2;
        rad_V = V(i,j)/2;
        rad_T = T(i,j)/2;
        vol = (4/3)*pi*rad_H*rad_V*rad_T;
        vol_tumors(i,j) = vol;
    end
end
end
```

A.2.2 Code for CM4

```
clear vars; close all; format long;
% Script to import excel data
% Note: No data for CM.50, nothing wrong with CM.50 that was noted?
load('CM51Hvals.mat') % 28 days
load('CM52Hvals.mat') % 29 days
load('CM53Hvals.mat') % 29 days
load('CM54Hvals.mat') % 26 days
load('CM55Hvals.mat') % 24 days
load('CM56Hvals.mat') % 30 days
load('CM57Hvals.mat') % 25 days
load('CM58Hvals.mat') % 24 days * Yellow Death
load('CM59Hvals.mat') % 31 days

CM51_H_M1 = [];
CM51_V_M1 = [];
CM51_H_M2 = [];
CM51_V_M2 = [];
CM51_H_M3 = [];
CM51_V_M3 = [];

CM52_H_M1 = [];
CM52_V_M1 = [];
CM52_H_M2 = [];
CM52_V_M2 = [];
CM52_H_M3 = [];
CM52_V_M3 = [];
```

```
CM53_H_M1 = [];
CM53_V_M1 = [];
CM53_H_M2 = [];
CM53_V_M2 = [];
CM53_H_M3 = [];
CM53_V_M3 = [];

CM54_H_M1 = [];
CM54_V_M1 = [];
CM54_H_M2 = [];
CM54_V_M2 = [];
CM54_H_M3 = [];
CM54_V_M3 = [];

CM55_H_M1 = [];
CM55_V_M1 = [];
CM55_H_M2 = [];
CM55_V_M2 = [];
CM55_H_M3 = [];
CM55_V_M3 = [];

CM56_H_M1 = [];
CM56_V_M1 = [];
CM56_H_M2 = [];
CM56_V_M2 = [];
CM56_H_M3 = [];
CM56_V_M3 = [];

CM57_H_M1 = [];
CM57_V_M1 = [];
CM57_H_M2 = [];
CM57_V_M2 = [];
CM57_H_M3 = [];
CM57_V_M3 = [];

CM58_H_M1 = [];
CM58_V_M1 = [];
CM58_H_M2 = [];
CM58_V_M2 = [];
CM58_H_M3 = [];
CM58_V_M3 = [];

CM59_H_M1 = [];
CM59_V_M1 = [];
CM59_H_M2 = [];
```

```
CM59_V_M2 = [];
CM59_H_M3 = [];
CM59_V_M3 = [];

% CM51 Mean 28 days
% NOTE: 1st day for CM51 is counted twice!!! 21/9 and then 21/9 for 2nd day
% Decided to make the 2nd day middle ground between 1st and 3rd day
% 1st day = 21/9, 3rd day = 5/10, therefore 2nd day = 28/9
for i = 1:28
    if i <= 27
        CM51_H_M1(i) = mean(CM51(1:3,i));
        CM51_V_M1(i) = mean(CM51(4:6,i));
    else
        CM51_H_M1(i) = 0;
        CM51_V_M1(i) = 0;
    end
    if i >= 2
        CM51_H_M2(i) = mean(CM51(7:9,i));
        CM51_V_M2(i) = mean(CM51(10:12,i));
    end
    if i >= 2 && i <= 23
        CM51_H_M3(i) = mean(CM51(13:15,i));
        CM51_V_M3(i) = mean(CM51(16:18,i));
    elseif i >= 24
        CM51_H_M3(i) = 0;
        CM51_V_M3(i) = 0;
    end
end
CM51_H_meanvals = [CM51_H_M1; CM51_H_M2; CM51_H_M3];
CM51_V_meanvals = [CM51_V_M1; CM51_V_M2; CM51_V_M3];
CM51_3rd_lengths = leng3(CM51_H_meanvals, CM51_V_meanvals);
CM51_vol_tumors = vol_solver(CM51_H_meanvals, CM51_V_meanvals,...
                             CM51_3rd_lengths);
CM51_time = [0 7 14 17 20 24 27 ...
             31 33 35 38 42 45 49 ...
             52 57 59 64 70 73 74 ...
             77 81 86 90 95 99 102];
figure
plot(CM51_time,CM51_vol_tumors(1,:),'*')
hold on;
plot(CM51_time,CM51_vol_tumors(2,:),'*')
hold on;
plot(CM51_time,CM51_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
```

```
legend('Tumor 1','Tumor 2','Tumor 3')

% CM52 Mean 29 days
for i = 1:29
    CM52_H_M1(i) = mean(CM52(1:3,i));
    CM52_V_M1(i) = mean(CM52(4:6,i));
    if i >= 7
        CM52_H_M2(i) = mean(CM52(7:9,i));
        CM52_V_M2(i) = mean(CM52(10:12,i));
    end
    if i >= 21 && i <= 23
        CM52_H_M3(i) = mean(CM52(13:15,i));
        CM52_V_M3(i) = mean(CM52(16:18,i));
    elseif i >= 24
        CM52_H_M3(i) = 0;
        CM52_V_M3(i) = 0;
    end
end
CM52_H_meanvals = [CM52_H_M1; CM52_H_M2; CM52_H_M3];
CM52_V_meanvals = [CM52_V_M1; CM52_V_M2; CM52_V_M3];
CM52_3rd_lengths = leng3(CM52_H_meanvals, CM52_V_meanvals);
CM52_vol_tumors = vol_solver(CM52_H_meanvals, CM52_V_meanvals,...
                             CM52_3rd_lengths);
CM52_time = [0 3 7 14 21 23 29 ...
             31 34 37 41 44 48 50 ...
             55 58 62 65 69 72 77 ...
             79 84 90 93 97 100 104 109];
figure
plot(CM52_time,CM52_vol_tumors(1,:),'*')
hold on;
plot(CM52_time,CM52_vol_tumors(2,:),'*')
hold on;
plot(CM52_time,CM52_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM53 Mean 29 days
for i = 1:29
    CM53_H_M1(i) = mean(CM53(1:3,i));
    CM53_V_M1(i) = mean(CM53(4:6,i));
    if i >= 4
        CM53_H_M2(i) = mean(CM53(7:9,i));
        CM53_V_M2(i) = mean(CM53(10:12,i));
    end
```

```
    if i >= 6
        CM53_H_M3(i) = mean(CM53(13:15,i));
        CM53_V_M3(i) = mean(CM53(16:18,i));
    end
end
CM53_H_meanvals = [CM53_H_M1; CM53_H_M2; CM53_H_M3];
CM53_V_meanvals = [CM53_V_M1; CM53_V_M2; CM53_V_M3];
CM53_3rd_lengths = leng3(CM53_H_meanvals, CM53_V_meanvals);
CM53_vol_tumors = vol_solver(CM53_H_meanvals, CM53_V_meanvals,...
                             CM53_3rd_lengths);
CM53_time = [0 3 7 9 15 17 20 ...
             23 27 30 34 36 41 44 ...
             48 51 55 58 63 65 70 ...
             76 79 83 86 90 95 99 104];
figure
plot(CM53_time,CM53_vol_tumors(1,:),'*')
hold on;
plot(CM53_time,CM53_vol_tumors(2,:),'*')
hold on;
plot(CM53_time,CM53_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM54 Mean 26 days
for i = 1:26
    CM54_H_M1(i) = mean(CM54(1:3,i));
    CM54_V_M1(i) = mean(CM54(4:6,i));
    if i >= 3 && i <= 16
        CM54_H_M2(i) = mean(CM54(7:9,i));
        CM54_V_M2(i) = mean(CM54(10:12,i));
    elseif i >= 17
        CM54_H_M2(i) = 0;
        CM54_V_M2(i) = 0;
    end
    if i >= 3
        CM54_H_M3(i) = mean(CM54(13:15,i));
        CM54_V_M3(i) = mean(CM54(16:18,i));
    end
end
CM54_H_meanvals = [CM54_H_M1; CM54_H_M2; CM54_H_M3];
CM54_V_meanvals = [CM54_V_M1; CM54_V_M2; CM54_V_M3];
CM54_3rd_lengths = leng3(CM54_H_meanvals, CM54_V_meanvals);
CM54_vol_tumors = vol_solver(CM54_H_meanvals, CM54_V_meanvals,...
                             CM54_3rd_lengths);
```

```
CM54_time = [0 7 9 15 17 20 ...
             23 27 30 34 36 41 44 ...
             48 51 55 58 63 65 70 ...
             76 79 83 86 90 95];
figure
plot(CM54_time,CM54_vol_tumors(1,:),'*')
hold on;
plot(CM54_time,CM54_vol_tumors(2,:),'*')
hold on;
plot(CM54_time,CM54_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM55 Mean 24 days
for i = 1:24
    CM55_H_M1(i) = mean(CM55(1:3,i));
    CM55_V_M1(i) = mean(CM55(4:6,i));
    CM55_H_M2(i) = mean(CM55(7:9,i));
    CM55_V_M2(i) = mean(CM55(10:12,i));
    CM55_H_M3(i) = mean(CM55(13:15,i));
    CM55_V_M3(i) = mean(CM55(16:18,i));
end
CM55_H_meanvals = [CM55_H_M1; CM55_H_M2; CM55_H_M3];
CM55_V_meanvals = [CM55_V_M1; CM55_V_M2; CM55_V_M3];
CM55_3rd_lengths = leng3(CM55_H_meanvals, CM55_V_meanvals);
CM55_vol_tumors = vol_solver(CM55_H_meanvals, CM55_V_meanvals,...
                             CM55_3rd_lengths);
CM55_time = [0 7 10 14 16 22 24 27 ...
             30 34 37 41 43 48 51 55 ...
             58 62 65 70 72 77 86 90];
figure
plot(CM55_time,CM55_vol_tumors(1,:),'*')
hold on;
plot(CM55_time,CM55_vol_tumors(2,:),'*')
hold on;
plot(CM55_time,CM55_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM56 Mean 30 days
for i = 1:30
    CM56_H_M1(i) = mean(CM56(1:3,i));
    CM56_V_M1(i) = mean(CM56(4:6,i));
```

```
    if i >= 5 && i <= 26
        CM56_H_M2(i) = mean(CM56(7:9,i));
        CM56_V_M2(i) = mean(CM56(10:12,i));
    elseif i >= 27
        CM56_H_M2(i) = 0;
        CM56_V_M2(i) = 0;
    end
    if i >= 12
        CM56_H_M3(i) = mean(CM56(13:15,i));
        CM56_V_M3(i) = mean(CM56(16:18,i));
    end
end
CM56_H_meanvals = [CM56_H_M1; CM56_H_M2; CM56_H_M3];
CM56_V_meanvals = [CM56_V_M1; CM56_V_M2; CM56_V_M3];
CM56_3rd_lengths = leng3(CM56_H_meanvals, CM56_V_meanvals);
CM56_vol_tumors = vol_solver(CM56_H_meanvals, CM56_V_meanvals,...
                             CM56_3rd_lengths);
CM56_time = [0 3 7 10 14 21 ...
             24 28 30 36 38 41 ...
             44 48 51 55 57 62 ...
             65 69 72 76 79 84 ...
             86 91 97 100 104 107];
figure
plot(CM56_time,CM56_vol_tumors(1,:),'*')
hold on;
plot(CM56_time,CM56_vol_tumors(2,:),'*')
hold on;
plot(CM56_time,CM56_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM57 Mean 25 days
for i = 1:25
    CM57_H_M1(i) = mean(CM57(1:3,i));
    CM57_V_M1(i) = mean(CM57(4:6,i));
    CM57_H_M2(i) = mean(CM57(7:9,i));
    CM57_V_M2(i) = mean(CM57(10:12,i));
    if i >= 2
        CM57_H_M3(i) = mean(CM57(13:15,i));
        CM57_V_M3(i) = mean(CM57(16:18,i));
    end
end
CM57_H_meanvals = [CM57_H_M1; CM57_H_M2; CM57_H_M3];
CM57_V_meanvals = [CM57_V_M1; CM57_V_M2; CM57_V_M3];
```

```
CM59_3rd_lengths = leng3(CM57_H_meanvals, CM57_V_meanvals);
CM57_vol_tumors = vol_solver(CM57_H_meanvals, CM57_V_meanvals,...
                             CM59_3rd_lengths);
CM57_time = [0 7 10 14 16 22 24 27 ...
             30 34 37 41 43 48 51 55 ...
             58 62 65 70 72 77 86 90 93];
figure
plot(CM57_time,CM57_vol_tumors(1,:),'*')
hold on;
plot(CM57_time,CM57_vol_tumors(2,:),'*')
hold on;
plot(CM57_time,CM57_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM58 Mean 24 days * Yellow Death
for i = 1:24
    if i <= 23
    CM58_H_M1(i) = mean(CM58(1:3,i));
    CM58_V_M1(i) = mean(CM58(4:6,i));
    elseif i == 24
        CM58_H_M1(i) = 0;
        CM58_V_M1(i) = 0;
    end
    CM58_H_M2(i) = mean(CM58(7:9,i));
    CM58_V_M2(i) = mean(CM58(10:12,i));
    if i >= 3
        CM58_H_M3(i) = mean(CM58(13:15,i));
        CM58_V_M3(i) = mean(CM58(16:18,i));
    end
end
CM58_H_meanvals = [CM58_H_M1; CM58_H_M2; CM58_H_M3];
CM58_V_meanvals = [CM58_V_M1; CM58_V_M2; CM58_V_M3];
CM58_3rd_lengths = leng3(CM58_H_meanvals, CM58_V_meanvals);
CM58_vol_tumors = vol_solver(CM58_H_meanvals, CM58_V_meanvals,...
                             CM58_3rd_lengths);
CM58_time = [0 3 7 10 14 21 ...
             24 28 30 36 38 41 ...
             44 48 51 55 57 62 ...
             65 69 72 76 79 84];
figure
plot(CM58_time,CM58_vol_tumors(1,:),'*')
hold on;
plot(CM58_time,CM58_vol_tumors(2,:),'*')
```

```
hold on;
plot(CM58_time,CM58_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM59 Mean 31 days
for i = 1:31
    if i <= 23
        CM59_H_M1(i) = mean(CM59(1:3,i));
        CM59_V_M1(i) = mean(CM59(4:6,i));
    elseif i >= 24
        CM59_H_M1(i) = 0;
        CM59_V_M1(i) = 0;
    end
    CM59_H_M2(i) = mean(CM59(7:9,i));
    CM59_V_M2(i) = mean(CM59(10:12,i));
    if i <= 25
        CM59_H_M3(i) = mean(CM59(13:15,i));
        CM59_V_M3(i) = mean(CM59(16:18,i));
    elseif i >= 26
        CM59_H_M3(i) = 0;
        CM59_V_M3(i) = 0;
    end
end
CM59_H_meanvals = [CM59_H_M1; CM59_H_M2; CM59_H_M3];
CM59_V_meanvals = [CM59_V_M1; CM59_V_M2; CM59_V_M3];
CM59_3rd_lengths = leng3(CM59_H_meanvals, CM59_V_meanvals);
CM59_vol_tumors = vol_solver(CM59_H_meanvals, CM59_V_meanvals,...
                             CM59_3rd_lengths);
CM59_time = [0 3 7 9 15 17 20 ...
             23 27 30 34 36 41 44 ...
             48 51 55 58 63 65 70 ...
             76 79 83 86 90 95 99 104 108 111];
figure
plot(CM59_time,CM59_vol_tumors(1,:),'*')
hold on;
plot(CM59_time,CM59_vol_tumors(2,:),'*')
hold on;
plot(CM59_time,CM59_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')
```

```
%%
% Function to create third diameter from horizontal and vertical
function third_length = leng3(H, V)
num_of_days = size(H,2);
num_of_tumors = size(H,1);
third_length = [];
for i = 1:num_of_tumors
    for j = 1:num_of_days
        leng3_val = (H(i,j) + V(i,j))/2;
        third_length(i,j) = leng3_val;
    end
end
end

% Function to compute volume of ellipsoid
function vol_tumors = vol_solver(H, V, T)
tumor_num = size(H,1);
day_num = size(H,2);
vol_tumors = [];
for i = 1:tumor_num
    for j = 1:day_num
        rad_H = H(i,j)/2;
        rad_V = V(i,j)/2;
        rad_T = T(i,j)/2;
        vol = (4/3)*pi*rad_H*rad_V*rad_T;
        vol_tumors(i,j) = vol;
    end
end
end
```

A.2.3 Code for CM5

```
clear vars; close all; format long;
% Script to import excel data
% Note: CM61, CM64, CM65 all died of natural causes, no data recored
load('CM60Hvals.mat') % 17 days
load('CM62Hvals.mat') % 13 days * Yellow Death
load('CM63Hvals.mat') % 23 days
load('CM66Hvals.mat') % 19 days
load('CM67Hvals.mat') % 28 days
load('CM68Hvals.mat') % 22 days
load('CM69Hvals.mat') % 20 days * Yellow Death
load('CM70Hvals.mat') % 23 days
load('CM71Hvals.mat') % 19 days * Yellow Death
```

```
CM60_H_M1 = [];
CM60_V_M1 = [];
CM60_H_M2 = [];
CM60_V_M2 = [];
CM60_H_M3 = [];
CM60_V_M3 = [];

CM62_H_M1 = [];
CM62_V_M1 = [];
CM62_H_M2 = [];
CM62_V_M2 = [];
CM62_H_M3 = [];
CM62_V_M3 = [];

CM63_H_M1 = [];
CM63_V_M1 = [];
CM63_H_M2 = [];
CM63_V_M2 = [];
CM63_H_M3 = [];
CM63_V_M3 = [];

CM66_H_M1 = [];
CM66_V_M1 = [];
CM66_H_M2 = [];
CM66_V_M2 = [];
CM66_H_M3 = [];
CM66_V_M3 = [];

CM67_H_M1 = [];
CM67_V_M1 = [];
CM67_H_M2 = [];
CM67_V_M2 = [];
CM67_H_M3 = [];
CM67_V_M3 = [];

CM68_H_M1 = [];
CM68_V_M1 = [];
CM68_H_M2 = [];
CM68_V_M2 = [];
CM68_H_M3 = [];
CM68_V_M3 = [];

CM69_H_M1 = [];
CM69_V_M1 = [];
CM69_H_M2 = [];
```

```
CM69_V_M2 = [];
CM69_H_M3 = [];
CM69_V_M3 = [];

CM70_H_M1 = [];
CM70_V_M1 = [];
CM70_H_M2 = [];
CM70_V_M2 = [];
CM70_H_M3 = [];
CM70_V_M3 = [];

CM71_H_M1 = [];
CM71_V_M1 = [];
CM71_H_M2 = [];
CM71_V_M2 = [];
CM71_H_M3 = [];
CM71_V_M3 = [];

% CM60 Mean 17 days
% NOTE: 1st day for CM51 is counted twice!!! 21/9 and then 21/9 for 2nd day
% Decided to make the 2nd day middle ground between 1st and 3rd day
% 1st day = 21/9, 3rd day = 5/10, therefore 2nd day = 28/9
for i = 1:17
    CM60_H_M1(i) = mean(CM60(1:3,i));
    CM60_V_M1(i) = mean(CM60(4:6,i));
    if i >= 3
        CM60_H_M2(i) = mean(CM60(7:9,i));
        CM60_V_M2(i) = mean(CM60(10:12,i));
    end
    if i >= 4
        CM60_H_M3(i) = mean(CM60(13:15,i));
        CM60_V_M3(i) = mean(CM60(16:18,i));
    end
end
CM60_H_meanvals = [CM60_H_M1; CM60_H_M2; CM60_H_M3];
CM60_V_meanvals = [CM60_V_M1; CM60_V_M2; CM60_V_M3];
CM60_3rd_lengths = leng3(CM60_H_meanvals, CM60_V_meanvals);
CM60_vol_tumors = vol_solver(CM60_H_meanvals, CM60_V_meanvals,...
                             CM60_3rd_lengths);
CM60_time = [0 3 6 10 13 17 19 24 27 ...
             32 34 38 41 46 48 53 62];
figure
plot(CM60_time,CM60_vol_tumors(1,:),'*')
hold on;
plot(CM60_time,CM60_vol_tumors(2,:),'*')
```

```
hold on;
plot(CM60_time,CM60_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM62 Mean 13 days * Yellow Death
for i = 1:13
    CM62_H_M1(i) = mean(CM62(1:3,i));
    CM62_V_M1(i) = mean(CM62(4:6,i));
    CM62_H_M2(i) = mean(CM62(7:9,i));
    CM62_V_M2(i) = mean(CM62(10:12,i));
    CM62_H_M3(i) = mean(CM62(13:15,i));
    CM62_V_M3(i) = mean(CM62(16:18,i));
end
CM62_H_meanvals = [CM62_H_M1; CM62_H_M2; CM62_H_M3];
CM62_V_meanvals = [CM62_V_M1; CM62_V_M2; CM62_V_M3];
CM62_3rd_lengths = leng3(CM62_H_meanvals, CM62_V_meanvals);
CM62_vol_tumors = vol_solver(CM62_H_meanvals, CM62_V_meanvals,...
                             CM62_3rd_lengths);
CM62_time = [0 2 8 10 13 16 20 ...
             23 27 29 34 37 42];
figure
plot(CM62_time,CM62_vol_tumors(1,:),'*')
hold on;
plot(CM62_time,CM62_vol_tumors(2,:),'*')
hold on;
plot(CM62_time,CM62_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM63 Mean 23 days
for i = 1:23
    CM63_H_M1(i) = mean(CM63(1:3,i));
    CM63_V_M1(i) = mean(CM63(4:6,i));
    if i >= 2
        CM63_H_M2(i) = mean(CM63(7:9,i));
        CM63_V_M2(i) = mean(CM63(10:12,i));
    end
    if i >= 6
        CM63_H_M3(i) = mean(CM63(13:15,i));
        CM63_V_M3(i) = mean(CM63(16:18,i));
    end
end
```

```
CM63_H_meanvals = [CM63_H_M1; CM63_H_M2; CM63_H_M3];
CM63_V_meanvals = [CM63_V_M1; CM63_V_M2; CM63_V_M3];
CM63_3rd_lengths = leng3(CM63_H_meanvals, CM63_V_meanvals);
CM63_vol_tumors = vol_solver(CM63_H_meanvals, CM63_V_meanvals,...
                             CM63_3rd_lengths);
CM63_time = [0 2 8 10 13 16 20 ...
             23 27 29 34 37 42 ...
             44 48 51 56 58 63 ...
             69 72 76 79];
figure
plot(CM63_time,CM63_vol_tumors(1,:),'*')
hold on;
plot(CM63_time,CM63_vol_tumors(2,:),'*')
hold on;
plot(CM63_time,CM63_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM66 Mean 19 days
for i = 1:19
    CM66_H_M1(i) = mean(CM66(1:3,i));
    CM66_V_M1(i) = mean(CM66(4:6,i));
    if i <= 17
        CM66_H_M2(i) = mean(CM66(7:9,i));
        CM66_V_M2(i) = mean(CM66(10:12,i));
    elseif i >= 18
        CM66_H_M2(i) = 0;
        CM66_V_M2(i) = 0;
    end
    if i >= 2 && i<= 17
        CM66_H_M3(i) = mean(CM66(13:15,i));
        CM66_V_M3(i) = mean(CM66(16:18,i));
    elseif i >= 18
        CM66_H_M3(i) = 0;
        CM66_V_M3(i) = 0;
    end
end
CM66_H_meanvals = [CM66_H_M1; CM66_H_M2; CM66_H_M3];
CM66_V_meanvals = [CM66_V_M1; CM66_V_M2; CM66_V_M3];
CM66_3rd_lengths = leng3(CM66_H_meanvals, CM66_V_meanvals);
CM66_vol_tumors = vol_solver(CM66_H_meanvals, CM66_V_meanvals,...
                             CM66_3rd_lengths);
CM66_time = [0 2 3 6 10 ...
             13 17 19 24 27 ...
```

```
             32 34 38 41 46 ...
             48 51 57 60];
figure
plot(CM66_time,CM66_vol_tumors(1,:),'*')
hold on;
plot(CM66_time,CM66_vol_tumors(2,:),'*')
hold on;
plot(CM66_time,CM66_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM67 Mean 28 days
for i = 1:28
    CM67_H_M1(i) = mean(CM67(1:3,i));
    CM67_V_M1(i) = mean(CM67(4:6,i));
    CM57_H_M2(i) = mean(CM67(7:9,i));
    CM67_V_M2(i) = mean(CM67(10:12,i));
    if i >= 8
        CM67_H_M3(i) = mean(CM67(13:15,i));
        CM67_V_M3(i) = mean(CM67(16:18,i));
    end
end
CM67_H_meanvals = [CM67_H_M1; CM57_H_M2; CM67_H_M3];
CM67_V_meanvals = [CM67_V_M1; CM67_V_M2; CM67_V_M3];
CM67_3rd_lengths = leng3(CM67_H_meanvals, CM67_V_meanvals);
CM67_vol_tumors = vol_solver(CM67_H_meanvals, CM67_V_meanvals,...
                             CM67_3rd_lengths);
CM67_time = [0 6 8 11 14 18 21 25 27 32 ...
             35 40 42 46 49 53 55 60 66 69 ...
             73 76 80 85 89 94 98 101];
figure
plot(CM67_time,CM67_vol_tumors(1,:),'*')
hold on;
plot(CM67_time,CM67_vol_tumors(2,:),'*')
hold on;
plot(CM67_time,CM67_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM68 Mean 22 days
for i = 1:22
    if i == 3
        CM68_H_M1(i) = 0;
```

```
        CM68_V_M1(i) = 0;
    else
        CM68_H_M1(i) = mean(CM68(1:3,i));
        CM68_V_M1(i) = mean(CM68(4:6,i));
    end
    if i == 1
        CM68_H_M2(i) = 0;
        CM68_V_M2(i) = 0;
    elseif i == 3
        CM68_H_M2(i) = 0;
        CM68_V_M2(i) = 0;
    else
        CM68_H_M2(i) = mean(CM68(7:9,i));
        CM68_V_M2(i) = mean(CM68(10:12,i));
    end
    if i >= 2
        CM68_H_M3(i) = mean(CM68(13:15,i));
        CM68_V_M3(i) = mean(CM68(16:18,i));
    end
end
CM68_H_meanvals = [CM68_H_M1; CM68_H_M2; CM68_H_M3];
CM68_V_meanvals = [CM68_V_M1; CM68_V_M2; CM68_V_M3];
CM68_3rd_lengths = leng3(CM68_H_meanvals, CM68_V_meanvals);
CM68_vol_tumors = vol_solver(CM68_H_meanvals, CM68_V_meanvals,...
                             CM68_3rd_lengths);
CM68_time = [0 7 10 14 16 22 24 27 30 34 ...
             37 41 43 48 51 56 58 62 65 70 ...
             72 77];
figure
plot(CM68_time,CM68_vol_tumors(1,:),'*')
hold on;
plot(CM68_time,CM68_vol_tumors(2,:),'*')
hold on;
plot(CM68_time,CM68_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM69 Mean 20 days * Yellow Death
for i = 1:20
    CM69_H_M1(i) = mean(CM69(1:3,i));
    CM69_V_M1(i) = mean(CM69(4:6,i));
    CM69_H_M2(i) = mean(CM69(7:9,i));
    CM69_V_M2(i) = mean(CM69(10:12,i));
    if i >= 6
```

```
        CM69_H_M3(i) = mean(CM69(13:15,i));
        CM69_V_M3(i) = mean(CM69(16:18,i));
    end
end
CM69_H_meanvals = [CM69_H_M1; CM69_H_M2; CM69_H_M3];
CM69_V_meanvals = [CM69_V_M1; CM69_V_M2; CM69_V_M3];
CM69_3rd_lengths = leng3(CM69_H_meanvals, CM69_V_meanvals);
CM69_vol_tumors = vol_solver(CM69_H_meanvals, CM69_V_meanvals,...
                             CM69_3rd_lengths);
CM69_time = [0 7 10 14 16 22 24 27 30 34 ...
             37 41 43 48 51 56 58 62 65 70];
figure
plot(CM69_time,CM69_vol_tumors(1,:),'*')
hold on;
plot(CM69_time,CM69_vol_tumors(2,:),'*')
hold on;
plot(CM69_time,CM69_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM70 Mean 23 days
for i = 1:23
    CM70_H_M1(i) = mean(CM70(1:3,i));
    CM70_V_M1(i) = mean(CM70(4:6,i));
    if i >= 3
        CM70_H_M2(i) = mean(CM70(7:9,i));
        CM70_V_M2(i) = mean(CM70(10:12,i));
    end
    if i >= 6
        CM70_H_M3(i) = mean(CM70(13:15,i));
        CM70_V_M3(i) = mean(CM70(16:18,i));
    end
end
CM70_H_meanvals = [CM70_H_M1; CM70_H_M2; CM70_H_M3];
CM70_V_meanvals = [CM70_V_M1; CM70_V_M2; CM70_V_M3];
CM70_3rd_lengths = leng3(CM70_H_meanvals, CM70_V_meanvals);
CM70_vol_tumors = vol_solver(CM70_H_meanvals, CM70_V_meanvals,...
                             CM70_3rd_lengths);
CM70_time = [0 2 3 6 10 ...
             13 17 19 24 27 ...
             32 34 38 41 46 ...
             48 51 57 60 64 ...
             67 71 76];
figure
```

```
plot(CM70_time,CM70_vol_tumors(1,:),'*')
hold on;
plot(CM70_time,CM70_vol_tumors(2,:),'*')
hold on;
plot(CM70_time,CM70_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM71 Mean 19 days * Yellow Death
for i = 1:19
    CM71_H_M1(i) = mean(CM71(1:3,i));
    CM71_V_M1(i) = mean(CM71(4:6,i));
    CM71_H_M2(i) = mean(CM71(7:9,i));
    CM71_V_M2(i) = mean(CM71(10:12,i));
    if i >= 3
        CM71_H_M3(i) = mean(CM71(13:15,i));
        CM71_V_M3(i) = mean(CM71(16:18,i));
    end
end
CM71_H_meanvals = [CM71_H_M1; CM71_H_M2; CM71_H_M3];
CM71_V_meanvals = [CM71_V_M1; CM71_V_M2; CM71_V_M3];
CM71_3rd_lengths = leng3(CM71_H_meanvals, CM71_V_meanvals);
CM71_vol_tumors = vol_solver(CM71_H_meanvals, CM71_V_meanvals,...
                             CM71_3rd_lengths);
CM71_time = [0 2 8 10 13 16 20 ...
             23 27 29 34 37 42 ...
             44 48 51 56 58 63];
figure
plot(CM71_time,CM71_vol_tumors(1,:),'*')
hold on;
plot(CM71_time,CM71_vol_tumors(2,:),'*')
hold on;
plot(CM71_time,CM71_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')


%%
% Function to create third diameter from horizontal and vertical
function third_length = leng3(H, V)
num_of_days = size(H,2);
num_of_tumors = size(H,1);
third_length = [];
```

```
for i = 1:num_of_tumors
    for j = 1:num_of_days
        leng3_val = (H(i,j) + V(i,j))/2;
        third_length(i,j) = leng3_val;
    end
end
end

% Function to compute volume of ellipsoid
function vol_tumors = vol_solver(H, V, T)
tumor_num = size(H,1);
day_num = size(H,2);
vol_tumors = [];
for i = 1:tumor_num
    for j = 1:day_num
        rad_H = H(i,j)/2;
        rad_V = V(i,j)/2;
        rad_T = T(i,j)/2;
        vol = (4/3)*pi*rad_H*rad_V*rad_T;
        vol_tumors(i,j) = vol;
    end
end
end
```

A.2.4 Code for CM6

```
clear vars; close all; format long;
% Script to import excel data

load('CM72Hvals.mat') % 17 days * Yellow Death
load('CM73Hvals.mat') % 24 days
load('CM74Hvals.mat') % 21 days
load('CM75Hvals.mat') % 24 days
load('CM76Hvals.mat') % 24 days
load('CM77Hvals.mat') % 27 days
load('CM78Hvals.mat') % 16 days
load('CM79Hvals.mat') % 15 days * Yellow Death

CM72_H_M1 = [];
CM72_V_M1 = [];
CM72_H_M2 = [];
CM72_V_M2 = [];
CM72_H_M3 = [];
CM72_V_M3 = [];
```

```
CM73_H_M1 = [];
CM73_V_M1 = [];
CM73_H_M2 = [];
CM73_V_M2 = [];
CM73_H_M3 = [];
CM73_V_M3 = [];

CM74_H_M1 = [];
CM74_V_M1 = [];
CM74_H_M2 = [];
CM74_V_M2 = [];
CM74_H_M3 = [];
CM74_V_M3 = [];

CM75_H_M1 = [];
CM75_V_M1 = [];
CM75_H_M2 = [];
CM75_V_M2 = [];
CM75_H_M3 = [];
CM75_V_M3 = [];

CM76_H_M1 = [];
CM76_V_M1 = [];
CM76_H_M2 = [];
CM76_V_M2 = [];
CM76_H_M3 = [];
CM76_V_M3 = [];

CM77_H_M1 = [];
CM77_V_M1 = [];
CM77_H_M2 = [];
CM77_V_M2 = [];
CM77_H_M3 = [];
CM77_V_M3 = [];

CM78_H_M1 = [];
CM78_V_M1 = [];
CM78_H_M2 = [];
CM78_V_M2 = [];
CM78_H_M3 = [];
CM78_V_M3 = [];

CM79_H_M1 = [];
CM79_V_M1 = [];
CM79_H_M2 = [];
```

```
CM79_V_M2 = [];
CM79_H_M3 = [];
CM79_V_M3 = [];

% CM72 Mean 17 days * Yellow Death
for i = 1:17
    CM72_H_M1(i) = mean(CM72(1:3,i));
    CM72_V_M1(i) = mean(CM72(4:6,i));
    if i >= 8
        CM72_H_M2(i) = mean(CM72(7:9,i));
        CM72_V_M2(i) = mean(CM72(10:12,i));
        CM72_H_M3(i) = mean(CM72(13:15,i));
        CM72_V_M3(i) = mean(CM72(16:18,i));
    end
end
CM72_H_meanvals = [CM72_H_M1; CM72_H_M2; CM72_H_M3];
CM72_V_meanvals = [CM72_V_M1; CM72_V_M2; CM72_V_M3];
CM72_3rd_lengths = leng3(CM72_H_meanvals, CM72_V_meanvals);
CM72_vol_tumors = vol_solver(CM72_H_meanvals, CM72_V_meanvals,...
                             CM72_3rd_lengths);
CM72_time = [0 2 6 9 13 ...
             16 20 23 27 29 ...
             34 37 41 44 48 51 55];
figure
plot(CM72_time,CM72_vol_tumors(1,:),'*')
hold on;
plot(CM72_time,CM72_vol_tumors(2,:),'*')
hold on;
plot(CM72_time,CM72_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM73 Mean 24 days
for i = 1:24
    CM73_H_M1(i) = mean(CM73(1:3,i));
    CM73_V_M1(i) = mean(CM73(4:6,i));
    if i >= 2
        CM73_H_M2(i) = mean(CM73(7:9,i));
        CM73_V_M2(i) = mean(CM73(10:12,i));
    end
    if i >= 3
        CM73_H_M3(i) = mean(CM73(13:15,i));
        CM73_V_M3(i) = mean(CM73(16:18,i));
    end
```

```
end
CM73_H_meanvals = [CM73_H_M1; CM73_H_M2; CM73_H_M3];
CM73_V_meanvals = [CM73_V_M1; CM73_V_M2; CM73_V_M3];
CM73_3rd_lengths = leng3(CM73_H_meanvals, CM73_V_meanvals);
CM73_vol_tumors = vol_solver(CM73_H_meanvals, CM73_V_meanvals,...
                             CM73_3rd_lengths);
CM73_time = [0 2 6 9 13 ...
             16 20 23 27 29 ...
             34 37 41 44 48 ...
             51 55 62 65 69 ...
             72 80 84 89];
figure
plot(CM73_time,CM73_vol_tumors(1,:),'*')
hold on;
plot(CM73_time,CM73_vol_tumors(2,:),'*')
hold on;
plot(CM73_time,CM73_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM74 Mean 21 days
for i = 1:21
    CM74_H_M1(i) = mean(CM74(1:3,i));
    CM74_V_M1(i) = mean(CM74(4:6,i));
    CM74_H_M2(i) = mean(CM74(7:9,i));
    CM74_V_M2(i) = mean(CM74(10:12,i));
    if i >= 4
        CM74_H_M3(i) = mean(CM74(13:15,i));
        CM74_V_M3(i) = mean(CM74(16:18,i));
    end
end
CM74_H_meanvals = [CM74_H_M1; CM74_H_M2; CM74_H_M3];
CM74_V_meanvals = [CM74_V_M1; CM74_V_M2; CM74_V_M3];
CM74_3rd_lengths = leng3(CM74_H_meanvals, CM74_V_meanvals);
CM74_vol_tumors = vol_solver(CM74_H_meanvals, CM74_V_meanvals,...
                             CM74_3rd_lengths);
CM74_time = [0 4 7 11 13 ...
             18 21 25 28 32 ...
             35 39 46 49 53 ...
             56 64 68 73 77 80];
figure
plot(CM74_time,CM74_vol_tumors(1,:),'*')
hold on;
plot(CM74_time,CM74_vol_tumors(2,:),'*')
```

```
hold on;
plot(CM74_time,CM74_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM75 Mean 24 days
for i = 1:24
    CM75_H_M1(i) = mean(CM75(1:3,i));
    CM75_V_M1(i) = mean(CM75(4:6,i));
    if i >= 4
        CM75_H_M2(i) = mean(CM75(7:9,i));
        CM75_V_M2(i) = mean(CM75(10:12,i));
    end
    if i >= 8 && i <= 22
        CM75_H_M3(i) = mean(CM75(13:15,i));
        CM75_V_M3(i) = mean(CM75(16:18,i));
    elseif i >= 23
        CM75_H_M3(i) = 0;
        CM75_V_M3(i) = 0;
    end
end
CM75_H_meanvals = [CM75_H_M1; CM75_H_M2; CM75_H_M3];
CM75_V_meanvals = [CM75_V_M1; CM75_V_M2; CM75_V_M3];
CM75_3rd_lengths = leng3(CM75_H_meanvals, CM75_V_meanvals);
CM75_vol_tumors = vol_solver(CM75_H_meanvals, CM75_V_meanvals,...
                             CM75_3rd_lengths);
CM75_time = [0 2 8 10 14 ...
             17 21 24 28 31 ...
             35 37 42 45 49 ...
             52 56 59 63 70 ...
             73 77 80 88];
figure
plot(CM75_time,CM75_vol_tumors(1,:),'*')
hold on;
plot(CM75_time,CM75_vol_tumors(2,:),'*')
hold on;
plot(CM75_time,CM75_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM76 Mean 24 days
for i = 1:24
    CM76_H_M1(i) = mean(CM76(1:3,i));
```

```
    CM76_V_M1(i) = mean(CM76(4:6,i));
    if i <= 22
        CM76_H_M2(i) = mean(CM76(7:9,i));
        CM76_V_M2(i) = mean(CM76(10:12,i));
    elseif i >= 23
        CM76_H_M2(i) = 0;
        CM76_V_M2(i) = 0;
    end
    if i >= 8
        CM76_H_M3(i) = mean(CM76(13:15,i));
        CM76_V_M3(i) = mean(CM76(16:18,i));
    end
end
CM76_H_meanvals = [CM76_H_M1; CM76_H_M2; CM76_H_M3];
CM76_V_meanvals = [CM76_V_M1; CM76_V_M2; CM76_V_M3];
CM76_3rd_lengths = leng3(CM76_H_meanvals, CM76_V_meanvals);
CM76_vol_tumors = vol_solver(CM76_H_meanvals, CM76_V_meanvals,...
                             CM76_3rd_lengths);
CM76_time = [0 2 8 10 14 ...
             17 21 24 28 31 ...
             35 37 42 45 49 ...
             52 56 59 63 70 ...
             73 76 80 88];
figure
plot(CM76_time,CM76_vol_tumors(1,:),'*')
hold on;
plot(CM76_time,CM76_vol_tumors(2,:),'*')
hold on;
plot(CM76_time,CM76_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM77 Mean 27 days
for i = 1:27
    CM77_H_M1(i) = mean(CM77(1:3,i));
    CM77_V_M1(i) = mean(CM77(4:6,i));
    CM77_H_M2(i) = mean(CM77(7:9,i));
    CM77_V_M2(i) = mean(CM77(10:12,i));
    if i >= 3 && i <= 25
        CM77_H_M3(i) = mean(CM77(13:15,i));
        CM77_V_M3(i) = mean(CM77(16:18,i));
    elseif i >= 26
        CM77_H_M3(i) = 0;
        CM77_V_M3(i) = 0;
```

```
    end
end
CM77_H_meanvals = [CM77_H_M1; CM77_H_M2; CM77_H_M3];
CM77_V_meanvals = [CM77_V_M1; CM77_V_M2; CM77_V_M3];
CM77_3rd_lengths = leng3(CM77_H_meanvals, CM77_V_meanvals);
CM77_vol_tumors = vol_solver(CM77_H_meanvals, CM77_V_meanvals,...
                             CM77_3rd_lengths);
CM77_time = [0 6 8 12 15 19 22 26 29 ...
             33 35 40 43 47 50 54 57 61 ...
             68 71 75 78 86 90 95 99 102];
figure
plot(CM77_time,CM77_vol_tumors(1,:),'*')
hold on;
plot(CM77_time,CM77_vol_tumors(2,:),'*')
hold on;
plot(CM77_time,CM77_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM78 Mean 16 days
for i = 1:16
    CM78_H_M1(i) = mean(CM78(1:3,i));
    CM78_V_M1(i) = mean(CM78(4:6,i));
    CM78_H_M2(i) = mean(CM78(7:9,i));
    CM78_V_M2(i) = mean(CM78(10:12,i));
    CM78_H_M3(i) = mean(CM78(13:15,i));
    CM78_V_M3(i) = mean(CM78(16:18,i));
end
CM78_H_meanvals = [CM78_H_M1; CM78_H_M2; CM78_H_M3];
CM78_V_meanvals = [CM78_V_M1; CM78_V_M2; CM78_V_M3];
CM78_3rd_lengths = leng3(CM78_H_meanvals, CM78_V_meanvals);
CM78_vol_tumors = vol_solver(CM78_H_meanvals, CM78_V_meanvals,...
                             CM78_3rd_lengths);
CM78_time = [0 7 11 17 ...
             20 24 26 31 ...
             34 38 41 45 ...
             48 52 59 62];
figure
plot(CM78_time,CM78_vol_tumors(1,:),'*')
hold on;
plot(CM78_time,CM78_vol_tumors(2,:),'*')
hold on;
plot(CM78_time,CM78_vol_tumors(3,:),'*')
xlabel('Time')
```

```
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM79 Mean 15 days * Yellow Death
for i = 1:15
    CM79_H_M1(i) = mean(CM79(1:3,i));
    CM79_V_M1(i) = mean(CM79(4:6,i));
    CM79_H_M2(i) = mean(CM79(7:9,i));
    CM79_V_M2(i) = mean(CM79(10:12,i));
    if i >= 3
        CM79_H_M3(i) = mean(CM79(13:15,i));
        CM79_V_M3(i) = mean(CM79(16:18,i));
    end
end
CM79_H_meanvals = [CM79_H_M1; CM79_H_M2; CM79_H_M3];
CM79_V_meanvals = [CM79_V_M1; CM79_V_M2; CM79_V_M3];
CM79_3rd_lengths = leng3(CM79_H_meanvals, CM79_V_meanvals);
CM79_vol_tumors = vol_solver(CM79_H_meanvals, CM79_V_meanvals,...
                             CM79_3rd_lengths);
CM79_time = [0 3 7 9 15 ...
             17 21 24 28 31 ...
             35 38 42 44 49];
figure
plot(CM79_time,CM79_vol_tumors(1,:),'*')
hold on;
plot(CM79_time,CM79_vol_tumors(2,:),'*')
hold on;
plot(CM79_time,CM79_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')


%%
% Function to create third diameter from horizontal and vertical
function third_length = leng3(H, V)
num_of_days = size(H,2);
num_of_tumors = size(H,1);
third_length = [];
for i = 1:num_of_tumors
    for j = 1:num_of_days
        leng3_val = (H(i,j) + V(i,j))/2;
        third_length(i,j) = leng3_val;
    end
end
```

```
end

% Function to compute volume of ellipsoid
function vol_tumors = vol_solver(H, V, T)
tumor_num = size(H,1);
day_num = size(H,2);
vol_tumors = [];
for i = 1:tumor_num
    for j = 1:day_num
        rad_H = H(i,j)/2;
        rad_V = V(i,j)/2;
        rad_T = T(i,j)/2;
        vol = (4/3)*pi*rad_H*rad_V*rad_T;
        vol_tumors(i,j) = vol;
    end
end
end
```

A.2.5 Code for CM7

```
clear vars; close all; format long;
% Script to import excel data
% Note: CM80, CM81, CM 82, CM87, CM88, CM89 all apart of the control group
load('CM83Hvals.mat') % 9 days  * Yellow Death
load('CM84Hvals.mat') % 21 days
load('CM85Hvals.mat') % 12 days * Yellow Death
load('CM86Hvals.mat') % 16 days
load('CM90Hvals.mat') % 20 days
load('CM91Hvals.mat') % 23 days
load('CM92Hvals.mat') % 22 days
load('CM93Hvals.mat') % 20 days

CM83_H_M1 = [];
CM83_V_M1 = [];
CM83_H_M2 = [];
CM83_V_M2 = [];
CM83_H_M3 = [];
CM83_V_M3 = [];

CM84_H_M1 = [];
CM84_V_M1 = [];
CM84_H_M2 = [];
CM84_V_M2 = [];
CM84_H_M3 = [];
CM84_V_M3 = [];
```

```
CM85_H_M1 = [];
CM85_V_M1 = [];
CM85_H_M2 = [];
CM85_V_M2 = [];
CM85_H_M3 = [];
CM85_V_M3 = [];

CM86_H_M1 = [];
CM86_V_M1 = [];
CM86_H_M2 = [];
CM86_V_M2 = [];
CM86_H_M3 = [];
CM86_V_M3 = [];

CM90_H_M1 = [];
CM90_V_M1 = [];
CM90_H_M2 = [];
CM90_V_M2 = [];
CM90_H_M3 = [];
CM90_V_M3 = [];

CM91_H_M1 = [];
CM91_V_M1 = [];
CM91_H_M2 = [];
CM91_V_M2 = [];
CM91_H_M3 = [];
CM91_V_M3 = [];

CM92_H_M1 = [];
CM92_V_M1 = [];
CM92_H_M2 = [];
CM92_V_M2 = [];
CM92_H_M3 = [];
CM92_V_M3 = [];

CM93_H_M1 = [];
CM93_V_M1 = [];
CM93_H_M2 = [];
CM93_V_M2 = [];
CM93_H_M3 = [];
CM93_V_M3 = [];

% CM83 Mean 9 days * Yellow Death
% NOTE: Only rat out of whole experiment where for one day there is a
```

```
% measurement for horizontal but NOT vertical: Happened Day 1, Tumor 1
% I just put in vertical measurements equal to horizontal of day 1
for i = 1:9
    CM83_H_M1(i) = mean(CM83(1:3,i));
    if i >= 2
        CM83_V_M1(i) = mean(CM83(4:6,i));
        CM83_H_M2(i) = mean(CM83(7:9,i));
        CM83_V_M2(i) = mean(CM83(10:12,i));
        CM83_H_M3(i) = mean(CM83(13:15,i));
        CM83_V_M3(i) = mean(CM83(16:18,i));
    end
end
CM83_V_M1(1,1) = CM83_H_M1(1,1);
CM83_H_meanvals = [CM83_H_M1; CM83_H_M2; CM83_H_M3];
CM83_V_meanvals = [CM83_V_M1; CM83_V_M2; CM83_V_M3];
CM83_3rd_lengths = leng3(CM83_H_meanvals, CM83_V_meanvals);
CM83_vol_tumors = vol_solver(CM83_H_meanvals, CM83_V_meanvals,...
                             CM83_3rd_lengths);
CM83_time = [0 2 6 9 13 ...
             16 20 23 27];
figure
plot(CM83_time,CM83_vol_tumors(1,:),'*')
hold on;
plot(CM83_time,CM83_vol_tumors(2,:),'*')
hold on;
plot(CM83_time,CM83_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM84 Mean 21 days
for i = 1:21
    if i <= 17
        CM84_H_M1(i) = mean(CM84(1:3,i));
        CM84_V_M1(i) = mean(CM84(4:6,i));
    elseif i >= 18
        CM84_H_M1(i) = 0;
        CM84_V_M1(i) = 0;
    end
    CM84_H_M2(i) = mean(CM84(7:9,i));
    CM84_V_M2(i) = mean(CM84(10:12,i));
    if i >= 18
        CM84_H_M3(i) = mean(CM84(13:15,i));
        CM84_V_M3(i) = mean(CM84(16:18,i));
    elseif i >= 19
```

```
        CM84_H_M3(i) = 0;
        CM84_V_M3(i) = 0;
    end
end
CM84_H_meanvals = [CM84_H_M1; CM84_H_M2; CM84_H_M3];
CM84_V_meanvals = [CM84_V_M1; CM84_V_M2; CM84_V_M3];
CM84_3rd_lengths = leng3(CM84_H_meanvals, CM84_V_meanvals);
CM84_vol_tumors = vol_solver(CM84_H_meanvals, CM84_V_meanvals,...
                             CM84_3rd_lengths);
CM84_time = [0 3 7 10 14 16 21 24 ...
             28 31 35 38 42 49 52 56 ...
             59 67 71 76 80];
figure
plot(CM84_time,CM84_vol_tumors(1,:),'*')
hold on;
plot(CM84_time,CM84_vol_tumors(2,:),'*')
hold on;
plot(CM84_time,CM84_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM85 Mean 12 days * Yellow Death
for i = 1:12
    CM85_H_M1(i) = mean(CM85(1:3,i));
    CM85_V_M1(i) = mean(CM85(4:6,i));
    CM85_H_M2(i) = mean(CM85(7:9,i));
    CM85_V_M2(i) = mean(CM85(10:12,i));
    CM85_H_M3(i) = mean(CM85(13:15,i));
    CM85_V_M3(i) = mean(CM85(16:18,i));
end
CM85_H_meanvals = [CM85_H_M1; CM85_H_M2; CM85_H_M3];
CM85_V_meanvals = [CM85_V_M1; CM85_V_M2; CM85_V_M3];
CM85_3rd_lengths = leng3(CM85_H_meanvals, CM85_V_meanvals);
CM85_vol_tumors = vol_solver(CM85_H_meanvals, CM85_V_meanvals,...
                             CM85_3rd_lengths);
CM85_time = [0 4 7 11 ...
             14 18 20 25 ...
             28 32 35 39];
figure
plot(CM85_time,CM85_vol_tumors(1,:),'*')
hold on;
plot(CM85_time,CM85_vol_tumors(2,:),'*')
hold on;
plot(CM85_time,CM85_vol_tumors(3,:),'*')
```

```
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM86 Mean 16 days
for i = 1:16
    if i <= 14
        CM86_H_M1(i) = mean(CM86(1:3,i));
        CM86_V_M1(i) = mean(CM86(4:6,i));
    elseif i >= 15
        CM86_H_M1(i) = mean(CM86(1:3,i));
        CM86_V_M1(i) = mean(CM86(4:6,i));
    end
    if i <= 13
        CM86_H_M2(i) = mean(CM86(7:9,i));
        CM86_V_M2(i) = mean(CM86(10:12,i));
    elseif i >= 14
        CM86_H_M2(i) = 0;
        CM86_V_M2(i) = 0;
    end
    if i >= 8
        CM86_H_M3(i) = mean(CM86(13:15,i));
        CM86_V_M3(i) = mean(CM86(16:18,i));
    end
end
CM86_H_meanvals = [CM86_H_M1; CM86_H_M2; CM86_H_M3];
CM86_V_meanvals = [CM86_V_M1; CM86_V_M2; CM86_V_M3];
CM86_3rd_lengths = leng3(CM86_H_meanvals, CM86_V_meanvals);
CM86_vol_tumors = vol_solver(CM86_H_meanvals, CM86_V_meanvals,...
                             CM86_3rd_lengths);
CM86_time = [0 4 6 11 ...
             14 18 21 25 ...
             28 32 39 42 ...
             46 49 57 61];
figure
plot(CM86_time,CM86_vol_tumors(1,:),'*')
hold on;
plot(CM86_time,CM86_vol_tumors(2,:),'*')
hold on;
plot(CM86_time,CM86_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM90 Mean 20 days
```

```
for i = 1:20
    CM90_H_M1(i) = mean(CM90(1:3,i));
    CM90_V_M1(i) = mean(CM90(4:6,i));
    if i >= 3
        CM90_H_M2(i) = mean(CM90(7:9,i));
        CM90_V_M2(i) = mean(CM90(10:12,i));
    end
    if i >= 5
        CM90_H_M3(i) = mean(CM90(13:15,i));
        CM90_V_M3(i) = mean(CM90(16:18,i));
    end
end
CM90_H_meanvals = [CM90_H_M1; CM90_H_M2; CM90_H_M3];
CM90_V_meanvals = [CM90_V_M1; CM90_V_M2; CM90_V_M3];
CM90_3rd_lengths = leng3(CM90_H_meanvals, CM90_V_meanvals);
CM90_vol_tumors = vol_solver(CM90_H_meanvals, CM90_V_meanvals,...
                             CM90_3rd_lengths);
CM90_time = [0 2 6 9 13 ...
             16 20 23 27 29 ...
             34 37 41 44 48 ...
             51 55 62 65 69];
figure
plot(CM90_time,CM90_vol_tumors(1,:),'*')
hold on;
plot(CM90_time,CM90_vol_tumors(2,:),'*')
hold on;
plot(CM90_time,CM90_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM91 Mean 23 days
for i = 1:23
    CM91_H_M1(i) = mean(CM91(1:3,i));
    CM91_V_M1(i) = mean(CM91(4:6,i));
    if i >= 3 && i <= 17
        CM91_H_M2(i) = mean(CM91(7:9,i));
        CM91_V_M2(i) = mean(CM91(10:12,i));
    elseif i >= 18
        CM91_H_M2(i) = 0;
        CM91_V_M2(i) = 0;
    end
    if i >= 4
        CM91_H_M3(i) = mean(CM91(13:15,i));
        CM91_V_M3(i) = mean(CM91(16:18,i));
```

```
    end
end
CM91_H_meanvals = [CM91_H_M1; CM91_H_M2; CM91_H_M3];
CM91_V_meanvals = [CM91_V_M1; CM91_V_M2; CM91_V_M3];
CM91_3rd_lengths = leng3(CM91_H_meanvals, CM91_V_meanvals);
CM91_vol_tumors = vol_solver(CM91_H_meanvals, CM91_V_meanvals,...
                             CM91_3rd_lengths);
CM91_time = [0 3 7 10 14 17 ...
             21 23 28 31 35 38 ...
             42 45 49 56 59 63 ...
             66 74 78 83 87];
figure
plot(CM91_time,CM91_vol_tumors(1,:),'*')
hold on;
plot(CM91_time,CM91_vol_tumors(2,:),'*')
hold on;
plot(CM91_time,CM91_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM92 Mean 22 days
for i = 1:22
    CM92_H_M1(i) = mean(CM92(1:3,i));
    CM92_V_M1(i) = mean(CM92(4:6,i));
    CM92_H_M2(i) = mean(CM92(7:9,i));
    CM92_V_M2(i) = mean(CM92(10:12,i));
    CM92_H_M3(i) = mean(CM92(13:15,i));
    CM92_V_M3(i) = mean(CM92(16:18,i));
end
CM92_H_meanvals = [CM92_H_M1; CM92_H_M2; CM92_H_M3];
CM92_V_meanvals = [CM92_V_M1; CM92_V_M2; CM92_V_M3];
CM92_3rd_lengths = leng3(CM92_H_meanvals, CM92_V_meanvals);
CM92_vol_tumors = vol_solver(CM92_H_meanvals, CM92_V_meanvals,...
                             CM92_3rd_lengths);
CM92_time = [0 2 6 9 13 ...
             16 20 23 27 29 ...
             34 37 41 44 48 ...
             51 55 62 65 68 ...
             72 80];
figure
plot(CM92_time,CM92_vol_tumors(1,:),'*')
hold on;
plot(CM92_time,CM92_vol_tumors(2,:),'*')
hold on;
```

```
plot(CM92_time,CM92_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM93 Mean 20 days
for i = 1:20
    CM93_H_M1(i) = mean(CM93(1:3,i));
    CM93_V_M1(i) = mean(CM93(4:6,i));
    if i >= 4
        CM93_H_M2(i) = mean(CM93(7:9,i));
        CM93_V_M2(i) = mean(CM93(10:12,i));
        CM93_H_M3(i) = mean(CM93(13:15,i));
        CM93_V_M3(i) = mean(CM93(16:18,i));
    end
end
CM93_H_meanvals = [CM93_H_M1; CM93_H_M2; CM93_H_M3];
CM93_V_meanvals = [CM93_V_M1; CM93_V_M2; CM93_V_M3];
CM93_3rd_lengths = leng3(CM93_H_meanvals, CM93_V_meanvals);
CM93_vol_tumors = vol_solver(CM93_H_meanvals, CM93_V_meanvals,...
                             CM93_3rd_lengths);
CM93_time = [0 2 6 9 13 ...
             16 20 23 27 29 ...
             34 37 41 44 48 ...
             51 55 62 65 69];
figure
plot(CM93_time,CM93_vol_tumors(1,:),'*')
hold on;
plot(CM93_time,CM93_vol_tumors(2,:),'*')
hold on;
plot(CM93_time,CM93_vol_tumors(3,:),'*')
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')


%%
% Function to create third diameter from horizontal and vertical
function third_length = leng3(H, V)
num_of_days = size(H,2);
num_of_tumors = size(H,1);
third_length = [];
for i = 1:num_of_tumors
    for j = 1:num_of_days
        leng3_val = (H(i,j) + V(i,j))/2;
```

```
        third_length(i,j) = leng3_val;
    end
end
end

% Function to compute volume of ellipsoid
function vol_tumors = vol_solver(H, V, T)
tumor_num = size(H,1);
day_num = size(H,2);
vol_tumors = [];
for i = 1:tumor_num
    for j = 1:day_num
        rad_H = H(i,j)/2;
        rad_V = V(i,j)/2;
        rad_T = T(i,j)/2;
        vol = (4/3)*pi*rad_H*rad_V*rad_T;
        vol_tumors(i,j) = vol;
    end
end
end
```

A.3 Code for Part 3

A.3.1 Code Modeling Chosen Injection CM's

```
clear vars; close all; format long;
% This script will be for the chosen injected mice
% Looking at
% CM3: CM41, CM42
% CM4: CM51, CM54
% CM5: CM63, CM67, CM70
% CM6: CM74
% CM7: CM84, CM86
load('CM41Hvals.mat') % 34 days
load('CM42Hvals.mat') % 31 days * Yellow Death
load('CM51Hvals.mat') % 28 days
load('CM54Hvals.mat') % 26 days
load('CM63Hvals.mat') % 23 days
load('CM67Hvals.mat') % 28 days
load('CM70Hvals.mat') % 23 days
load('CM74Hvals.mat') % 21 days
load('CM84Hvals.mat') % 21 days
load('CM86Hvals.mat') % 16 days

CM41_H_M1= [];
CM41_V_M1 = [];
```

```
CM41_H_M2 = [];
CM41_V_M2 = [];
CM41_H_M3 = [];
CM41_V_M3 = [];

CM42_H_M1= [];
CM42_V_M1 = [];
CM42_H_M2 = [];
CM42_V_M2 = [];
CM42_H_M3 = [];
CM42_V_M3 = [];

CM51_H_M1 = [];
CM51_V_M1 = [];
CM51_H_M2 = [];
CM51_V_M2 = [];
CM51_H_M3 = [];
CM51_V_M3 = [];

CM54_H_M1 = [];
CM54_V_M1 = [];
CM54_H_M2 = [];
CM54_V_M2 = [];
CM54_H_M3 = [];
CM54_V_M3 = [];

CM63_H_M1 = [];
CM63_V_M1 = [];
CM63_H_M2 = [];
CM63_V_M2 = [];
CM63_H_M3 = [];
CM63_V_M3 = [];

CM67_H_M1 = [];
CM67_V_M1 = [];
CM67_H_M2 = [];
CM67_V_M2 = [];
CM67_H_M3 = [];
CM67_V_M3 = [];

CM70_H_M1 = [];
CM70_V_M1 = [];
CM70_H_M2 = [];
CM70_V_M2 = [];
CM70_H_M3 = [];
```

```
CM70_V_M3 = [];

CM74_H_M1 = [];
CM74_V_M1 = [];
CM74_H_M2 = [];
CM74_V_M2 = [];
CM74_H_M3 = [];
CM74_V_M3 = [];

CM84_H_M1 = [];
CM84_V_M1 = [];
CM84_H_M2 = [];
CM84_V_M2 = [];
CM84_H_M3 = [];
CM84_V_M3 = [];

CM86_H_M1 = [];
CM86_V_M1 = [];
CM86_H_M2 = [];
CM86_V_M2 = [];
CM86_H_M3 = [];
CM86_V_M3 = [];

% CM41 Mean 34 days
for i = 1:34
    CM41_H_M1(i) = mean(CM41(1:3,i));
    CM41_V_M1(i) = mean(CM41(4:6,i));
    CM41_H_M2(i) = mean(CM41(7:9,i));
    CM41_V_M2(i) = mean(CM41(10:12,i));
    if i >= 2
        CM41_H_M3(i) = mean(CM41(13:15,i));
        CM41_V_M3(i) = mean(CM41(16:18,i));
    end
end
CM41_H_meanvals = [CM41_H_M1; CM41_H_M2; CM41_H_M3];
CM41_V_meanvals = [CM41_V_M1; CM41_V_M2; CM41_V_M3];
CM41_3rd_lengths = leng3(CM41_H_meanvals, CM41_V_meanvals);
CM41_vol_tumors = vol_solver(CM41_H_meanvals, CM41_V_meanvals,...
                             CM41_3rd_lengths);
CM41_time = [0 4 7 11 14 18 ...
             21 25 28 32 35 39 ...
             42 46 49 52 55 59 ...
             62 66 68 73 70 80 ...
             83 87 90 95 97 102 ...
             108 111 115 118];
```

```
figure
plot(CM41_time,CM41_vol_tumors(1,:),'*')
hold on;
plot(CM41_time,CM41_vol_tumors(2,:),'*')
hold on;
plot(CM41_time,CM41_vol_tumors(3,:),'*')
xline(29)
hold on;


% CM42 Mean 31 days * Yellow Death
for i = 1:31
    if i<=23
        CM42_H_M1(i) = mean(CM42(1:3,i));
        CM42_V_M1(i) = mean(CM42(4:6,i));
    else
        CM42_H_M1(i) = 0;
        CM42_V_M1(i) = 0;
    end
    CM42_H_M2(i) = mean(CM42(7:9,i));
    CM42_V_M2(i) = mean(CM42(10:12,i));
    if i >= 2
        CM42_H_M3(i) = mean(CM42(13:15,i));
        CM42_V_M3(i) = mean(CM42(16:18,i));
    end
end
CM42_H_meanvals = [CM42_H_M1; CM42_H_M2; CM42_H_M3];
CM42_V_meanvals = [CM42_V_M1; CM42_V_M2; CM42_V_M3];
CM42_3rd_lengths = leng3(CM42_H_meanvals, CM42_V_meanvals);
CM42_vol_tumors = vol_solver(CM42_H_meanvals, CM42_V_meanvals,...
                             CM42_3rd_lengths);
CM42_time = [0 3 7 10 14 17 ...
             21 24 28 31 35 38 ...
             42 45 49 52 56 59 ...
             62 65 69 72 76 78 ...
             83 85 89 92 96 99 104];
figure
plot(CM42_time,CM42_vol_tumors(1,:),'*')
hold on;
plot(CM42_time,CM42_vol_tumors(2,:),'*')
hold on;
plot(CM42_time,CM42_vol_tumors(3,:),'*')
xline(26)
xlabel('Time')
ylabel('Volume')
```

```
legend('Tumor 1','Tumor 2','Tumor 3')

% CM51 Mean 28 days
% NOTE: 1st day for CM51 is counted twice!!! 21/9 and then 21/9 for 2nd day
% Decided to make the 2nd day middle ground between 1st and 3rd day
% 1st day = 21/9, 3rd day = 5/10, therefore 2nd day = 28/9
for i = 1:28
    if i <= 27
        CM51_H_M1(i) = mean(CM51(1:3,i));
        CM51_V_M1(i) = mean(CM51(4:6,i));
    else
        CM51_H_M1(i) = 0;
        CM51_V_M1(i) = 0;
    end
    if i >= 2
        CM51_H_M2(i) = mean(CM51(7:9,i));
        CM51_V_M2(i) = mean(CM51(10:12,i));
    end
    if i >= 2 && i <= 23
        CM51_H_M3(i) = mean(CM51(13:15,i));
        CM51_V_M3(i) = mean(CM51(16:18,i));
    elseif i >= 24
        CM51_H_M3(i) = 0;
        CM51_V_M3(i) = 0;
    end
end
CM51_H_meanvals = [CM51_H_M1; CM51_H_M2; CM51_H_M3];
CM51_V_meanvals = [CM51_V_M1; CM51_V_M2; CM51_V_M3];
CM51_3rd_lengths = leng3(CM51_H_meanvals, CM51_V_meanvals);
CM51_vol_tumors = vol_solver(CM51_H_meanvals, CM51_V_meanvals,...
                             CM51_3rd_lengths);
CM51_time = [0 7 14 17 20 24 27 ...
             31 33 35 38 42 45 49 ...
             52 57 59 64 70 73 74 ...
             77 81 86 90 95 99 102];
figure
plot(CM51_time,CM51_vol_tumors(1,:),'*')
hold on;
plot(CM51_time,CM51_vol_tumors(2,:),'*')
hold on;
plot(CM51_time,CM51_vol_tumors(3,:),'*')
xline(33)
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')
```

```
% CM54 Mean 26 days
for i = 1:26
    CM54_H_M1(i) = mean(CM54(1:3,i));
    CM54_V_M1(i) = mean(CM54(4:6,i));
    if i >= 3 && i <= 16
        CM54_H_M2(i) = mean(CM54(7:9,i));
        CM54_V_M2(i) = mean(CM54(10:12,i));
    elseif i >= 17
        CM54_H_M2(i) = 0;
        CM54_V_M2(i) = 0;
    end
    if i >= 3
        CM54_H_M3(i) = mean(CM54(13:15,i));
        CM54_V_M3(i) = mean(CM54(16:18,i));
    end
end
CM54_H_meanvals = [CM54_H_M1; CM54_H_M2; CM54_H_M3];
CM54_V_meanvals = [CM54_V_M1; CM54_V_M2; CM54_V_M3];
CM54_3rd_lengths = leng3(CM54_H_meanvals, CM54_V_meanvals);
CM54_vol_tumors = vol_solver(CM54_H_meanvals, CM54_V_meanvals,...
                             CM54_3rd_lengths);
CM54_time = [0 7 9 15 17 20 ...
             23 27 30 34 36 41 44 ...
             48 51 55 58 63 65 70 ...
             76 79 83 86 90 95];
figure
plot(CM54_time,CM54_vol_tumors(1,:),'*')
hold on;
plot(CM54_time,CM54_vol_tumors(2,:),'*')
hold on;
plot(CM54_time,CM54_vol_tumors(3,:),'*')
xline(43)
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM63 Mean 23 days
for i = 1:23
    CM63_H_M1(i) = mean(CM63(1:3,i));
    CM63_V_M1(i) = mean(CM63(4:6,i));
    if i >= 2
        CM63_H_M2(i) = mean(CM63(7:9,i));
        CM63_V_M2(i) = mean(CM63(10:12,i));
    end
```

```
    if i >= 6
        CM63_H_M3(i) = mean(CM63(13:15,i));
        CM63_V_M3(i) = mean(CM63(16:18,i));
    end
end
CM63_H_meanvals = [CM63_H_M1; CM63_H_M2; CM63_H_M3];
CM63_V_meanvals = [CM63_V_M1; CM63_V_M2; CM63_V_M3];
CM63_3rd_lengths = leng3(CM63_H_meanvals, CM63_V_meanvals);
CM63_vol_tumors = vol_solver(CM63_H_meanvals, CM63_V_meanvals,...
                             CM63_3rd_lengths);
CM63_time = [0 2 8 10 13 16 20 ...
             23 27 29 34 37 42 ...
             44 48 51 56 58 63 ...
             69 72 76 79];
figure
plot(CM63_time,CM63_vol_tumors(1,:),'*')
hold on;
plot(CM63_time,CM63_vol_tumors(2,:),'*')
hold on;
plot(CM63_time,CM63_vol_tumors(3,:),'*')
xline(22)
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM67 Mean 28 days
for i = 1:28
    CM67_H_M1(i) = mean(CM67(1:3,i));
    CM67_V_M1(i) = mean(CM67(4:6,i));
    CM57_H_M2(i) = mean(CM67(7:9,i));
    CM67_V_M2(i) = mean(CM67(10:12,i));
    if i >= 8
        CM67_H_M3(i) = mean(CM67(13:15,i));
        CM67_V_M3(i) = mean(CM67(16:18,i));
    end
end
CM67_H_meanvals = [CM67_H_M1; CM57_H_M2; CM67_H_M3];
CM67_V_meanvals = [CM67_V_M1; CM67_V_M2; CM67_V_M3];
CM67_3rd_lengths = leng3(CM67_H_meanvals, CM67_V_meanvals);
CM67_vol_tumors = vol_solver(CM67_H_meanvals, CM67_V_meanvals,...
                             CM67_3rd_lengths);
CM67_time = [0 6 8 11 14 18 21 25 27 32 ...
             35 40 42 46 49 53 55 60 66 69 ...
             73 76 80 85 89 94 98 101];
figure
```

```
plot(CM67_time,CM67_vol_tumors(1,:),'*')
hold on;
plot(CM67_time,CM67_vol_tumors(2,:),'*')
hold on;
plot(CM67_time,CM67_vol_tumors(3,:),'*')
xline(20)
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM70 Mean 23 days
for i = 1:23
    CM70_H_M1(i) = mean(CM70(1:3,i));
    CM70_V_M1(i) = mean(CM70(4:6,i));
    if i >= 3
        CM70_H_M2(i) = mean(CM70(7:9,i));
        CM70_V_M2(i) = mean(CM70(10:12,i));
    end
    if i >= 6
        CM70_H_M3(i) = mean(CM70(13:15,i));
        CM70_V_M3(i) = mean(CM70(16:18,i));
    end
end
CM70_H_meanvals = [CM70_H_M1; CM70_H_M2; CM70_H_M3];
CM70_V_meanvals = [CM70_V_M1; CM70_V_M2; CM70_V_M3];
CM70_3rd_lengths = leng3(CM70_H_meanvals, CM70_V_meanvals);
CM70_vol_tumors = vol_solver(CM70_H_meanvals, CM70_V_meanvals,...
                             CM70_3rd_lengths);
CM70_time = [0 2 3 6 10 ...
             13 17 19 24 27 ...
             32 34 38 41 46 ...
             48 51 57 60 64 ...
             67 71 76];
figure
plot(CM70_time,CM70_vol_tumors(1,:),'*')
hold on;
plot(CM70_time,CM70_vol_tumors(2,:),'*')
hold on;
plot(CM70_time,CM70_vol_tumors(3,:),'*')
xline(19)
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM74 Mean 21 days
```

```
for i = 1:21
    CM74_H_M1(i) = mean(CM74(1:3,i));
    CM74_V_M1(i) = mean(CM74(4:6,i));
    CM74_H_M2(i) = mean(CM74(7:9,i));
    CM74_V_M2(i) = mean(CM74(10:12,i));
    if i >= 4
        CM74_H_M3(i) = mean(CM74(13:15,i));
        CM74_V_M3(i) = mean(CM74(16:18,i));
    end
end
CM74_H_meanvals = [CM74_H_M1; CM74_H_M2; CM74_H_M3];
CM74_V_meanvals = [CM74_V_M1; CM74_V_M2; CM74_V_M3];
CM74_3rd_lengths = leng3(CM74_H_meanvals, CM74_V_meanvals);
CM74_vol_tumors = vol_solver(CM74_H_meanvals, CM74_V_meanvals,...
                             CM74_3rd_lengths);
CM74_time = [0 4 7 11 13 ...
             18 21 25 28 32 ...
             35 39 46 49 53 ...
             56 64 68 73 77 80];
figure
plot(CM74_time,CM74_vol_tumors(1,:),'*')
hold on;
plot(CM74_time,CM74_vol_tumors(2,:),'*')
hold on;
plot(CM74_time,CM74_vol_tumors(3,:),'*')
xline(26)
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM84 Mean 21 days
for i = 1:21
    if i <= 17
        CM84_H_M1(i) = mean(CM84(1:3,i));
        CM84_V_M1(i) = mean(CM84(4:6,i));
    elseif i >= 18
        CM84_H_M1(i) = 0;
        CM84_V_M1(i) = 0;
    end
    CM84_H_M2(i) = mean(CM84(7:9,i));
    CM84_V_M2(i) = mean(CM84(10:12,i));
    if i >= 18
        CM84_H_M3(i) = mean(CM84(13:15,i));
        CM84_V_M3(i) = mean(CM84(16:18,i));
    elseif i >= 19
```

```
        CM84_H_M3(i) = 0;
        CM84_V_M3(i) = 0;
    end
end
CM84_H_meanvals = [CM84_H_M1; CM84_H_M2; CM84_H_M3];
CM84_V_meanvals = [CM84_V_M1; CM84_V_M2; CM84_V_M3];
CM84_3rd_lengths = leng3(CM84_H_meanvals, CM84_V_meanvals);
CM84_vol_tumors = vol_solver(CM84_H_meanvals, CM84_V_meanvals,...
                             CM84_3rd_lengths);
CM84_time = [0 3 7 10 14 16 21 24 ...
             28 31 35 38 42 49 52 56 ...
             59 67 71 76 80];
figure
plot(CM84_time,CM84_vol_tumors(1,:),'*')
hold on;
plot(CM84_time,CM84_vol_tumors(2,:),'*')
hold on;
plot(CM84_time,CM84_vol_tumors(3,:),'*')
xline(36)
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')

% CM86 Mean 16 days
for i = 1:16
    if i <= 14
        CM86_H_M1(i) = mean(CM86(1:3,i));
        CM86_V_M1(i) = mean(CM86(4:6,i));
    elseif i >= 15
        CM86_H_M1(i) = mean(CM86(1:3,i));
        CM86_V_M1(i) = mean(CM86(4:6,i));
    end
    if i <= 13
        CM86_H_M2(i) = mean(CM86(7:9,i));
        CM86_V_M2(i) = mean(CM86(10:12,i));
    elseif i >= 14
        CM86_H_M2(i) = 0;
        CM86_V_M2(i) = 0;
    end
    if i >= 8
        CM86_H_M3(i) = mean(CM86(13:15,i));
        CM86_V_M3(i) = mean(CM86(16:18,i));
    end
end
CM86_H_meanvals = [CM86_H_M1; CM86_H_M2; CM86_H_M3];
```

```
CM86_V_meanvals = [CM86_V_M1; CM86_V_M2; CM86_V_M3];
CM86_3rd_lengths = leng3(CM86_H_meanvals, CM86_V_meanvals);
CM86_vol_tumors = vol_solver(CM86_H_meanvals, CM86_V_meanvals,...
                             CM86_3rd_lengths);
CM86_time = [0 4 6 11 ...
             14 18 21 25 ...
             28 32 39 42 ...
             46 49 57 61];
figure
plot(CM86_time,CM86_vol_tumors(1,:),'*')
hold on;
plot(CM86_time,CM86_vol_tumors(2,:),'*')
hold on;
plot(CM86_time,CM86_vol_tumors(3,:),'*')
xline(19)
xlabel('Time')
ylabel('Volume')
legend('Tumor 1','Tumor 2','Tumor 3')



%%
% Function to create third diameter from horizontal and vertical
function third_length = leng3(H, V)
num_of_days = size(H,2);
num_of_tumors = size(H,1);
third_length = [];
for i = 1:num_of_tumors
    for j = 1:num_of_days
        leng3_val = (H(i,j) + V(i,j))/2;
        third_length(i,j) = leng3_val;
    end
end
end

% Function to compute volume of ellipsoid
function vol_tumors = vol_solver(H, V, T)
tumor_num = size(H,1);
day_num = size(H,2);
vol_tumors = [];
for i = 1:tumor_num
    for j = 1:day_num
        rad_H = H(i,j)/2;
        rad_V = V(i,j)/2;
        rad_T = T(i,j)/2;
```

```
        vol = (4/3)*pi*rad_H*rad_V*rad_T;
        vol_tumors(i,j) = vol;
    end
end
end
```

A.3.2 Logistic MATLAB Code

Main Logistic Code:

```
function varargout = Logistic(varargin)
% Type Logistic into the Command Window!!!!!
% LOGISTIC M-file for Logistic.fig
%
% Copyright 2011 Varuna De Silva
% I-Lab, CVSSP, University of Surrey
% Guildford
% GU2-7XH
% UK
%
% Email: varunax@gmail.com
%
%
% This work is based on the excellent tutorial by David Arnold
% http://online.redwoods.cc.ca.us/instruct/darnold/diffeq/logistic/logistic
% .pdf
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Logistic_OpeningFcn, ...
                   'gui_OutputFcn',  @Logistic_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Logistic is made visible.
```

```
function Logistic_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Logistic (see VARARGIN)

% Choose default command line output for Logistic
handles.output = hObject;
handles.numFrames = 14;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Logistic wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Logistic_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function C_INP1_Path_Callback(hObject, eventdata, handles)
% hObject    handle to C_INP1_Path (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of C_INP1_Path as text
%        str2double(get(hObject,'String')) returns contents of C_INP1_Path as a double


% --- Executes during object creation, after setting all properties.
function C_INP1_Path_CreateFcn(hObject, eventdata, handles)
% hObject    handle to C_INP1_Path (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function C_INP2_Path_Callback(hObject, eventdata, handles)
% hObject    handle to C_INP2_Path (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of C_INP2_Path as text
%        str2double(get(hObject,'String')) returns contents of C_INP2_Path as a double


% --- Executes during object creation, after setting all properties.
function C_INP2_Path_CreateFcn(hObject, eventdata, handles)
% hObject    handle to C_INP2_Path (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in C_INP1_Browse.
function C_INP1_Browse_Callback(hObject, eventdata, handles)
% hObject    handle to C_INP1_Browse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[FileName,PathName] = uigetfile('*.txt','Select the MOS File');
handles.C_INP1 = fullfile(PathName,FileName);
set(handles.C_INP1_Path,'String',handles.C_INP1);
handles.MOS = dlmread(handles.C_INP1)
guidata(hObject, handles);
```

```
% --- Executes on button press in C_INP2_Browse.
function C_INP2_Browse_Callback(hObject, eventdata, handles)
% hObject    handle to C_INP2_Browse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[FileName,PathName] = uigetfile('*.txt','Select the METRIC File');
handles.C_INP2 = fullfile(PathName,FileName);
set(handles.C_INP2_Path,'String',handles.C_INP2);
handles.METRIC = dlmread(handles.C_INP2)
guidata(hObject, handles);



% --- Executes on button press in resetBut.
function resetBut_Callback(hObject, eventdata, handles)
% hObject    handle to resetBut (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes1,'reset');
set(handles.final_params,'String','');
set(handles.final_params2,'String','');
set(handles.min_r,'String','0');
set(handles.max_r,'String','0');
set(handles.est_r,'String','0');
set(handles.est_t0,'String','0');
guidata(hObject, handles); %updates the handles


% --- Executes on button press in plotBut.
function plotBut_Callback(hObject, eventdata, handles)
% hObject    handle to plotBut (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1)

plot(handles.METRIC,handles.MOS,'o')
%adds a title, x-axis description, and y-axis description


est_t0 = median(handles.METRIC);
xx = handles.METRIC-est_t0;
title('MOS Vs. METRIC')
ylabel('MOS')
xlabel('METRIC')
```

```
[max_xx down_idx]= min(xx);
for idx = 1:size(xx)
    if(xx(idx)<0)
        if(xx(idx)>max_xx)
            max_xx=xx(idx); down_idx = idx;
        end
    end
end
[min_xx up_idx]= max(xx); for idx = 1:size(xx)
    if(xx(idx)>0)
        if(xx(idx)<min_xx)
            min_xx=xx(idx); up_idx = idx;
        end
    end
end

mt0 = (handles.MOS(up_idx)-handles.MOS(down_idx))/(handles.METRIC(up_idx)-handles.METRIC(down_

est_r = 4*mt0;

min_r = est_r/10;
max_r = est_r*10;

set(handles.min_r,'String',num2str(min_r));
set(handles.max_r,'String',num2str(max_r));
set(handles.est_r,'String',num2str(est_r));
set(handles.est_t0,'String',num2str(est_t0));

%Calculate the estimates
guidata(hObject, handles); %updates the handles




function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function est_t0_Callback(hObject, eventdata, handles)
% hObject    handle to est_t0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of est_t0 as text
%        str2double(get(hObject,'String')) returns contents of est_t0 as a double


% --- Executes during object creation, after setting all properties.
function est_t0_CreateFcn(hObject, eventdata, handles)
% hObject    handle to est_t0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function est_r_Callback(hObject, eventdata, handles)
% hObject    handle to est_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of est_r as text
```

```
%        str2double(get(hObject,'String')) returns contents of est_r as a double


% --- Executes during object creation, after setting all properties.
function est_r_CreateFcn(hObject, eventdata, handles)
% hObject    handle to est_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function min_r_Callback(hObject, eventdata, handles)
% hObject    handle to min_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of min_r as text
%        str2double(get(hObject,'String')) returns contents of min_r as a double


% --- Executes during object creation, after setting all properties.
function min_r_CreateFcn(hObject, eventdata, handles)
% hObject    handle to min_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function max_r_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to max_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of max_r as text
%        str2double(get(hObject,'String')) returns contents of max_r as a double


% --- Executes during object creation, after setting all properties.
function max_r_CreateFcn(hObject, eventdata, handles)
% hObject    handle to max_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in FindFit.
function FindFit_Callback(hObject, eventdata, handles)
% hObject    handle to FindFit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

est_r = str2num(get(handles.est_r,'String'));
est_t0 = str2num(get(handles.est_t0,'String'));

min_x = min(handles.METRIC);
max_x = max(handles.METRIC);

min_r = est_r/10;
max_r = est_r*10;

r=linspace(min_r,max_r,40);
t0=linspace(min_x,max_x,40);
[r,t0]=meshgrid(r,t0);

[m,n]=size(r);
e=zeros(size(r));
```

```
for i=1:m
for j=1:n
e(i,j)=myerror([r(i,j);t0(i,j)],handles.METRIC,handles.MOS);
end
end

[row col] = find(min(min(e)));


min2=fminsearch(@myerror,[r(col);t0(row)],[],handles.METRIC,handles.MOS);

r=min2(1);
t0=min2(2);

H=1./(1+exp(-r*(handles.METRIC-t0)));

K=(H'*handles.MOS)/(H'*H)

t=linspace(min(handles.METRIC),max(handles.METRIC));
y=K./(1+exp(-r*(t-t0)));
plot(handles.METRIC,handles.MOS,'o',t,y)

s_sq_e = sum_sq_e([r; t0; K;],handles.METRIC,handles.MOS)
y1 = K./(1+exp(-r*(handles.METRIC-t0)));
R = corrcoef(handles.MOS,y1);
C_cof = R(1,2);
RMSE = sqrt(s_sq_e/size(handles.METRIC,1));
finalparams = ['K = ',num2str(K), '  G = ', num2str(r), '  Dm = ', num2str(t0)];
set(handles.final_params,'String',finalparams);
finalparams = ['C = ',num2str(C_cof), '  SSE = ', num2str(s_sq_e), '  RMSE = ', num2str(RMSE)]
set(handles.final_params2,'String',finalparams);

title('Y Vs. X')
ylabel('Y - Values')
xlabel('X - Values')


guidata(hObject, handles); %updates the handles
```

Error Code:

```
function e=myerror(x,t,m)
r=x(1);
t0=x(2);
h=1./(1+exp(-r*(t-t0)));
e=m'*m-(h'*m)^2/(h'*h);
```

Sum Square Error Code:

```
function e=sum_sq_e(x,t,m)
r=x(1);
t0=x(2);
K = x(3);
h=K./(1+exp(-r*(t-t0)));
e=m'*m-(h'*m)^2/(h'*h);
```

A.3.3 .txt Files for Logistic File

Note: All tumors share the same time values, so for the single tumor pieces, they all use the same X file.

For Tumor 1:

```
X File Values:
0
4
7
11
14
18
21
25
28

Y File Values:
0.879
3.2681
2.8688
6.4412
3.1652
3.208
4.4724
9.5781
10.19
```

For Tumor 2:

```
X File Values:
0
4
```

```
7
11
14
18
21
25
28

Y File Values:
1.6965
2.7822
3.104
8.2796
15.445
8.9401
10.692
19.828
15.021
```

For Tumor 3:

```
X File Values:
0
4
7
11
14
18
21
25
28

Y File Values:
0
0.90477
1.8111
5.7362
7.9946
11.506
17.367
24.278
31.904
```

For All values of CM41 mouse:

```
X File Values:
0
4
7
11
14
18
21
25
28
32
35
39
42
46
49
52
55
59
62
66
68
73
70
80
83
87
90
95
97
102
108
111
115
118
0
4
7
11
14
18
```

21
25
28
32
35
39
42
46
49
52
55
59
62
66
68
73
70
80
83
87
90
95
97
102
108
111
115
118
0
4
7
11
14
18
21
25
28
32
35
39
42
46
49
52
55

```
59
62
66
68
73
70
80
83
87
90
95
97
102
108
111
115
118

Y File Values:
0.879
3.2681
2.8688
6.4412
3.1652
3.208
4.4724
9.5781
10.19
7.7476
17.562
22.852
35.493
71.932
89.505
100.58
79.692
40.497
41.674
41.978
38.571
39.747
50.467
43.148
41.877
40.424
```

41.32
32.772
34.768
30.339
30.597
39.203
31.775
39.063
1.6965
2.7822
3.104
8.2796
15.445
8.9401
10.692
19.828
15.021
11.89
12.171
34.241
25.11
24.519
40.303
64.406
44.315
24.94
34.641
18.768
17.279
30.923
36.359
33.024
54.305
47.523
54.069
58.996
79.993
41.38
43.479
50.676
23.97
51.742
0
0.90477
1.8111

```
5.7362
7.9946
11.506
17.367
24.278
31.904
38.947
53.845
54.758
57.678
77.577
49.352
68.484
65.085
79.096
85.136
110.06
52.909
27.226
31.211
60.086
116.69
117.96
135.52
107.37
129.89
215.13
36.178
24.845
17.256
24.239
```

References

[1] De Silva, Varuna. “Fit Logistic Curve to a Data Set.” MathWorks, https://www.mathworks.com/matlabcentral/fileexchange/31399-fit-logistic-curve-to-a-data-set?s_tid=FX_rc1_behav.

[2] “FASTSTATS - Leading Causes of Death.” Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 18 Jan. 2023, https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm.

[3] Gompertz, Benjamin. “On the Nature of the Function Expressive of the Law of Human Mortality, and on a New Mode of Determining the Value of Life Contingencies.” Philosophical Transactions of the Royal Society of London, vol. 115, 1825, pp. 513–83. JSTOR, http://www.jstor.org/stable/107756.

[4] “Heaviside Step Function.” Wikipedia, Wikimedia Foundation, 7 Mar. 2023, https://en.wikipedia.org/wiki/Heaviside_step_function#: :text=The%20Heaviside%20step%20 function%2C%20or,and%20one%20for%20positive%20arguments.

[5] “Logistic Function.” Wikipedia, Wikimedia Foundation, 7 Apr. 2023, https://en.wikipedia.org/wiki/Logistic_function.

[6] Loizides, Charalambos, et al. “Model-Based Tumor Growth Dynamics and Therapy Response in a Mouse Model of De Novo Carcinogenesis.” PLOS ONE, Public Library of Science, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0143840

[7] Mathematical Models of Tumor Growth - Srce. https://hrcak.srce.hr/file/2874.

[8] “Odefun.” Solve Nonstiff Differential Equations - Medium Order Method - MATLAB, https://www.mathworks.com/help/matlab/ref/ode45.html.

[9] Verhulst, P.F. 1845. Recherches math´ematiques sur la loi d’accroissement de la population. Nouv. m´em. de l’Academie Royale des Sci. et Belles-Lettres de Bruxelles. 18: 1–41.

[10] von Bertalanffy, Ludwig. “Quantitative Laws in Metabolism and Growth.” The Quarterly Review of Biology, vol. 32, no. 3, 1957, pp. 217–31. JSTOR, http://www.jstor.org/stable/2815257.

[11] Wikipedia contributors. ”Differential equation.” Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 8 Apr. 2023. Web. 10 Apr. 2023.

[12] Wong, Jue. “Student Version Modeling Cancer Growth with Differential Equations.” SIMIODE A Systemic Initiative for Modeling Investigations and Opportunities with Differential Equations, https://cauchy.simiode.org/resources/4847/download/1-102-S-CancerGrowth–StudentVersion.pdf.