ENGR3 Final Project

What is Tasmanian Devil Facial Tumor Disease?

Devil facial tumor disease (DFTD) is a transmissible cancer that affects Tasmanian devils. DFTD begins with lumps of soft tissue around the mouth or eyes. These tumors grow aggressively and eventually lead to organ failure, infection, or starvation from the inability to feed. DFTD is rare in juveniles and it affects males and females equally. It is believed that the disease is primarily transmitted through biting, mating, and sharing of food. DFTD was first observed in Tasmania in 1996 and since then it has ravaged Tasmania's wild devils.

(a) Healthy Tasmanian devils

(b) Devil facial tumor disease

You have been hired by the Tasmanian government to model the spread of the disease and see if anything can be done to save the population of Tasmanian devils. You will use data of the population of Tasmanian devils to find a model describing the dynamics of how the disease spreads. The government would like you to use the model to determine if any initiatives can be done to save the Tasmanian devils before it's too late.

Part 1: Exploring the data (10 points)

The first step of tackling a time series problem like this is getting to know the data; therefore, your first task consists of loading the data file `devil_data.mat` provided on GauchoSpace and plotting its contents.

The file consists of a dataset for the total population of Tasmanian devils from 1985 to 2020. It has two columns: the first contains the **Year** and the second contains the **Total Population of Tasmanian Devils at a certain date**.

(a) Create a script named `explore_data.m` that loads the file `devil_data.mat` provided.

(b) Still in your `explore_data.m` script, create a figure for the total number of Tasmanian devils vs time. Make sure that you:

- Show only markers for the individual data points (no line)
- Name the x and y axes as "Year" and "Population" respectively.
- Give your plot an appropriate title
- Turn the grid on using the command `grid on`.
- Increase the font size of legends, axis labels and title using the command: `set(gca,'FontSize',12)`.
- The y-axis will display in scientific notation (x10^4) and we want to display the numbers in thousands instead. Correct this problem with the command: set(gca, 'YTickLabel', get(gca, 'YTick'))

Part 2: Disease-Free ODE Model (20 points)

The population of Tasmanian devils without any disease can be modeled by first writing down mechanisms that occur and then using the mechanisms to develop a set of equations describing the process. Healthy adult devils (S) give birth to juvenile devils (J) at a rate k_{birth}, and juvenile devils mature into adult devils at a rate k_{mature}. Both juvenile and adult devils naturally die at a rate D_s and D_j for healthy adult and juveniles respectively. There is a carrying capacity for the population of devils that causes death at a rate of D_{over} when the species is overpopulated. We can write these mechanistic processes as the following system of coupled, ordinary differential equations (ODEs)

$$\frac{dJ}{dt} = k_{birth}S - k_{mature}J - D_j J - D_{over}JT \tag{1}$$

$$\frac{dS}{dt} = k_{mature}J - D_s S - D_{over}ST, \tag{2}$$

where the total population is given by $T = J + S$. The values of the parameters are $k_{birth} =$ 0.055 devils^{-1}months^{-1}, $k_{mature} =$ 0.04 devils^{-1}months^{-1} , $D_j =$ 0.007 devils^{-1}months^{-1}, $D_s =$ 0.02335 devils^{-1}months^{-1}, and D_{over} =2.3e-7 devils^{-2}months^{-1}.

(a) (10 points) Write a function named `DiseaseFreeODE_model.m` that takes as input `t` (time) and `X` (a vector of J and S in that order), and returns the right-hand side of the ODE system given by Eqs. (1) - (2).

(b) (10 points) Write a script named `DiseaseFreeODE_sim.m` that solves the ODE system.

- Use the `ode45()` function to solve the ODE system. For the initial conditions, start with an initial population of 16165 Juveniles (J) and 18450 healthy adults (S). Integrate the ODE over the same interval of time as the data you plotted in Part 1. Take a look at the units for the ODE's parameters. Since time is in units of months, you will need to integrate your ODE in months and then convert to years after ode45 is called. To get the integration time in months, multiply the number of years for which we have data for by 12. Let ode45() choose the intervals of time that are outputted.

- Create a figure that plots the simulation results for J, S, and T along with the observed data from Part 1, as shown in Fig 2. Give the simulation results a line and keep the observed data as data markers as in Part 1. Make sure to give your figure a title, label the axes, and create a legend.

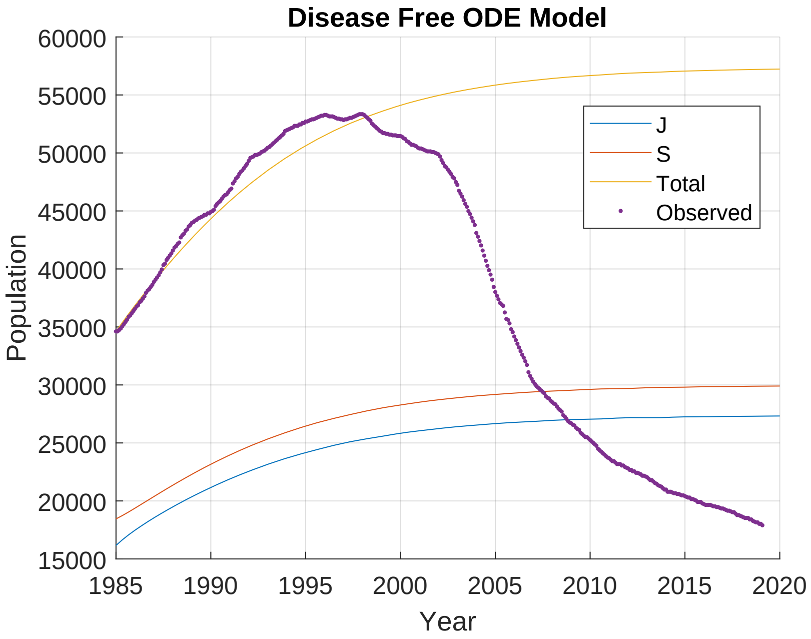

Figure 2: Plot of the solution of the disease free ODE model along with the observed data.

Part 3: Disease ODE Model (35 points)

To model the population of Tasmanian devils with the disease, we can build on the equations we used for the disease-free model. In the disease-free model, S referred to a healthy adult. When modeling with the disease, we sub-divide the adult devils into four stages of the disease: Susceptible (S), Exposed (E), Infected (I), and Diseased (D). Susceptible devils are devils without the disease that are able to pick it up from interactions with an Infected or Diseased devil. Once a susceptible devil has been exposed to the virus from an Infected or Diseased devil, it changes to an Exposed devil at rates $I_{infected}$ and $I_{exposed}$ respectively. An Exposed devil is a devil who has been exposed to the disease but not actively spreading it. Exposed devils move to the Infected class after an incubation period of $T_{incubation}$. Similarly, Infected devils move to the Diseased class after an incubation period of $T_{progression}$. Infected and Diseased devils die from the disease at rates of $D_{infected}$ and $D_{diseased}$. Only Susceptible, Exposed, and Infected devils can give birth to juvenile devils; diseased devils are too sick to reproduce. Additionally, all classes of devils die due to overcrowding effects. We can write these processes as the following system of coupled, ordinary differential equations (ODEs)

$$\frac{dJ}{dt} = k_{birth}(S+E+I) - k_{mature}J - D_jJ - D_{over}JT \tag{3}$$

$$\frac{dS}{dt} = k_{mature}J - I_{infected}SI - I_{diseased}SD - D_sS - D_{over}ST \tag{4}$$

$$\frac{dE}{dt} = I_{infected}SI + I_{diseased}SD - T_{incubation}E - D_eE - D_{over}ET \tag{5}$$

$$\frac{dI}{dt} = T_{incubation}E - T_{progression}I - D_iI - D_{over}IT \tag{6}$$

$$\frac{dD}{dt} = T_{progression}I - D_dD - D_{over}DT, \tag{7}$$

where the total population is given by $T = J+S+E+I+D$. The values of the additional parameters are $I_{infected}$ =1.0e-05 devils^{-1}months^{-1}, $I_{diseased}$ =3.84e-05 devils^{-1}months^{-1}, $T_{incubation}$ = 0.0976 devils^{-1}months^{-1}, $T_{progression}$ = 0.0931 devils^{-1}months^{-1}, D_e = 0.02335 devils^{-1}months^{-1}, D_i = 0.022609 devils^{-1}months^{-1}, and D_d = 0.29017 devils^{-1}months^{-1}

(a) (10 points) Write a function named `DiseaseODE_model.m` that takes as input `t` (time) and `X` (a vector of J, S, E, I, and D in that order), and returns the right-hand side of the ODE system given by Eqs. (3) - (7).

(b) (15 points) Write a script named `DiseaseODE_sim.m` that solves the ODE system.

- Use the `ode45()` function to solve the ODE system. For the initial conditions, start with an initial population of 16165 Juveniles (J), 18450 Susceptibles, 0 Exposed, 0 Infected, and 1 Diseased. Integrate the ODE over the same interval of time as you did in Part 2. Take a look at the units for the ODE's parameters. Since time is in units of months, you will need to integrate your ODE in months and then convert to years after ode45 is called.

- Create a new figure in the same file that plots the simulation results for J, S, E, I, D, and T along with the observed data from Part 1, as shown in Fig 3. Give the simulation results a line and keep the observed data as data markers like in Part 1. Make sure to give your figure a title, label the axes, and create a legend.

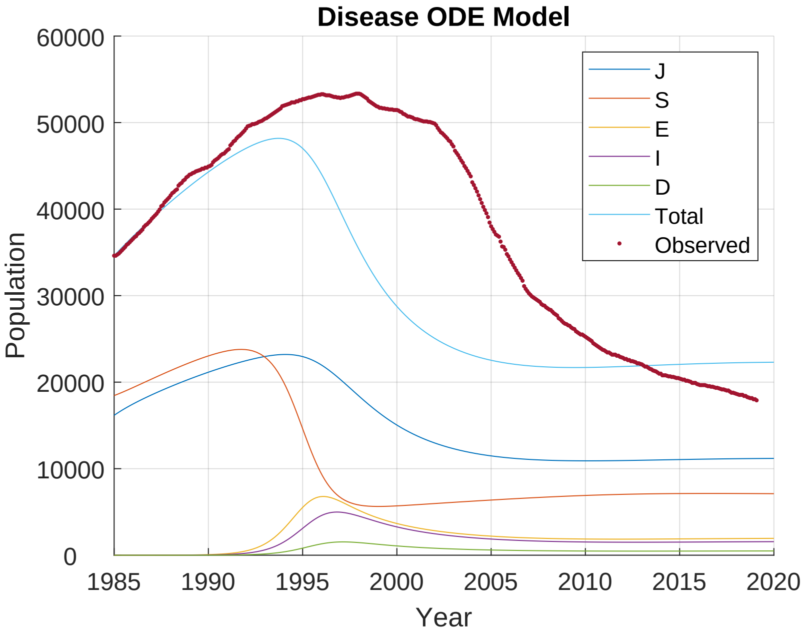

Figure 3: Plot of the solution of the disease ODE model along with the observed data.

Part 4: Estimating the DFTD_start parameter from the data (45 points)

The data for the population of Tasmanian devils includes data for the population both before and after the devil facial tumor disease (DFTD) existed. To create an accurate model that fits the data well, we will need to use both the disease-free model and the disease model. The simulation starts by using the disease-free model up to the time when the first case of the disease appears. Following the first case of DFTD, the disease version of the model needs to be used. Take another look at the ODE equations for both the disease and disease-free model. If you set $E = 0$, $I = 0$, and $D = 0$ in the disease model, you will recover the same equations as in the disease-free model. Since the disease model can be used to simulate the disease-free model with the right initial conditions, you can use either the disease model or disease free model to simulate times prior to the appearance of DFTD.

To fit our model to the data, we will need to first determine, as best we can given the data, the date when the disease first appeared (we don't know this value). We will call this value DFTD_start. We can find an approximate start date for DFTD (DFTD_start) by performing simulations with different values for DFTD_start and determining which simulation matches the data the closest. One common approach for determining the distance between two time series of data are is to minimize the Euclidean norm of the distance (err) between them:

$$err = \sqrt{\frac{\sum_{i=1}^{Ndays}(T(i) - \text{devil\_data}(i,2))^2}{N}}, \tag{8}$$

where N is the number of data values in the data (i.e., the length of the `devil_data` matrix).

Here is a brief summary of what you need to do. For various values of DFTD_start: solve the disease-free model up to DFTD_start, solve the disease model up to year 2020, then calculate the error using the above equation. The DFTD_start value with the lowest error is the value that matches the data the best. The steps to take are enumerated below.

(a) (20 points) Write a function file named `CompleteSim_fun.m` that takes as input a value of DFTD_start, solves the ODE system, and then returns as function output the time, ODE solution, and error (using Eq. (8)). The time and ODE solution are the combined t and y values from calling ode45(). Have the function return the time in units of months. Hint: the function cat(), which concatenates (combines) arrays might be helpful.

- Use the `ode45()` function to solve the disease-free version of the model up to DFTD_start using the same initial conditions as in Part 2. You can use either the Disease-FreeODE_model or DiseaseODE_model with proper initial conditions if you remember that $E = 0$, $I = 0$, and $D = 0$ for the disease-free case. Remember that the function takes time in months as its input. Have ode45 return the time in the range: 0:1:DFTD_start.
- Then immediately use the `ode45()` function to solve the disease version of the model from DFTD_start to 2020 using the last values of J and S from the disease-free version of the model as initial conditions. For the other initial conditions, use $E = 0$, $I = 0$, and $D = 1$. Remember that the function takes time in months as its input. Have ode45 return the time in the range: DFTD_start:1:409.

- Concatenate the time (t) and solution (y) arrays outputted from the two separate calls to ode45 to get one array for t and one array for y. Make sure that the time corresponding to DFTD_start doesn't repeat twice in your final array t (to prevent this don't include the last time from the disease-free simulation in the concatenation step). Also make sure that the combined t and y arrays have the same length.

- Compute the error using Eq. (8) for the value of DFTD_start (function input). The calculated error should be returned by the function as a scalar value. When calculating the error, the size of the output from the integration needs to be of the same size as the observed data and the values for time must match the ones in the observed data. If you receive error messages, look at the values of time outputted from the integration and check that they match the observed data (same values and same length).

- Here is the structure of your final function: [t, y, error]=CompleteSim_fun(DFTD_start)

(b) (25 points) Write a script named `DFTD_inference.m` that solves the ODE system for various values of DFTD_start (use a for loop for that).

- Use the following range of values for DFTD_start: `DFTD_start = (5:1:180)`, where the units are months. For each value, call the `CompleteSim_fun.m` function to solve the ODE system and calculate the error. Store the error outputted from the function in an array. You need this array to generate a figure and choose the value of DFTD_start that fits the data the best.

- Since the `CompleteSim_fun.m` function integrates the equations with ode45, you should not call ode45 in this script. The integration is taken care of inside of that function.

(c) Still in your script `DFTD_inference.m`, plot DFTD_start vs *err*. Part b should have given you the arrays you need to generate the figure. Make sure that:

- Your plots look like Fig. 4.

- Give your plot the following title: "Error Analysis (Complete ODE Model vs Data)".

- Name the x and y axis as "DFTD Start Time (months)" and "Error", respectively.

- Turn the grid on using the command `grid on`.

- Increase the font size of legends, axis labels and title using the command: `set(gca,'FontSize',12)`.

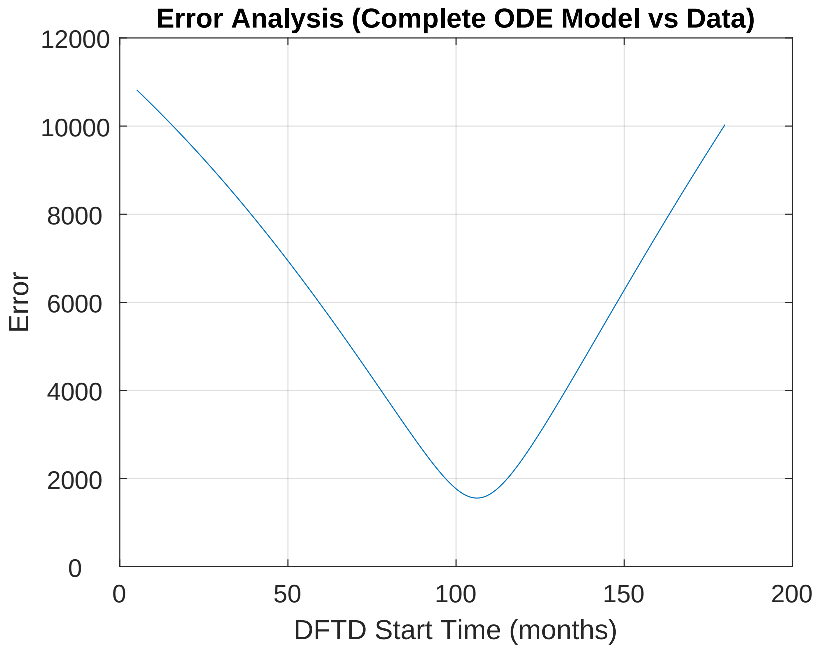

Figure 4: Evoluation of the Euclidean norm of the error as a function of DFTD_start.

(d) Still in your script `DFTD_inference.m`, plot J, S, E, I, D, and T for the best value of DFTD_start found in part (b-c) along with the observed data. Make sure that:

- Your plots look like Fig. 5.
- Change the line thickness to 3 for the simulation results. Keep the observed data as individual marker points like in Part 1.
- Give your plot the following title: "ODE Model for Population of Tasmanian Devils".
- Name the x and y axis as "Year" and "Population", respectively.
- Turn the grid on using the command `grid on`.
- Increase the font size of legends, axis labels and title using the command: `set(gca,'FontSize',12)`.
- Add a dashed vertical line corresponding to the DFTD start time. Label this line in your legend as "DFTD Start Time." Hint: use xline()
- The y-axis will display in scientific notation (x10^4) and we want to display the numbers in thousands instead. Correct this problem with the command: set(gca, 'YTickLabel', get(gca, 'YTick'))
- Note: your legend does not need to have the same colors or the same order. Just make sure it has all of the labels correctly corresponding with the components of the figure.

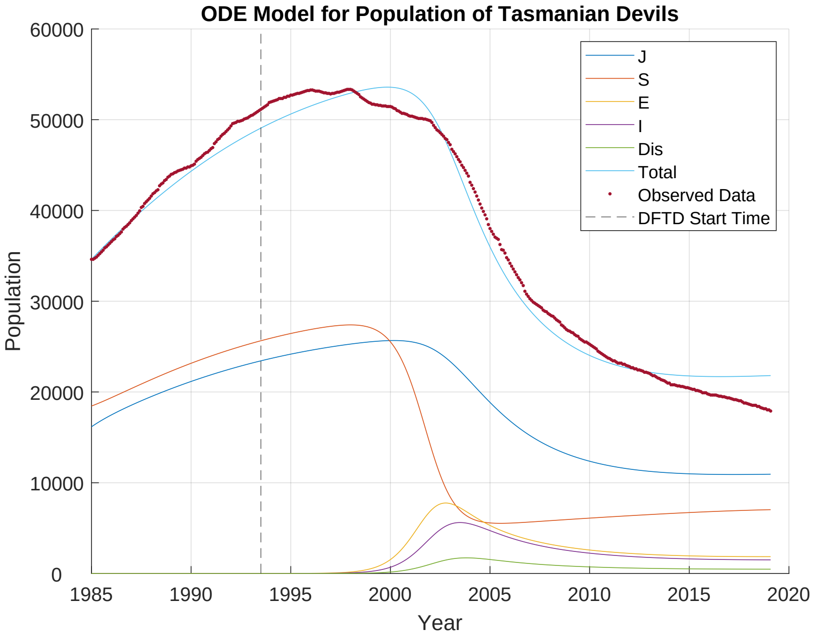

Figure 5: Plot of the solution of the complete ODE model following DFTD start time parameter inference.

(e) Modify the `CompleteSim_fun.m` function to also plot a version of Fig 5 corresponding to the value of DFTD_start passed into the function. Nothing about your function changes besides the fact that it now generates a figure corresponding to the DFTD_start input.

- Copy the part of code used to generate Fig 5 into the `CompleteSim_fun.m` function.
- Use the command clf to clear the figure prior to plotting
- Run your `DFTD_inference.m` file again. It should now generate code resembling an animation. This animation shows how the model changes for different values of DFTD_start. Add a pause(0.005) inside your for loop to control the animation speed. A video of the animation is provided on GauchoSpace to check your work. You don't need to generate this mp4 file.

Submission Checklist

Here's a list of the 7 files that need to be submitted:

- `explore_data.m` (10 points)
- `DiseaseFreeODE_model.m` (10 points)
- `DiseaseFreeODE_sim.m` (10 points)
- `DiseaseODE_model.m` (15 points)
- `DiseaseODE_sim.m` (10 points)
- `CompleteSim_fun.m` (20 points)
- `DFTD_inference.m` (25 points)

As always, make sure that you include comments in your code. Comments are an important part of coding, as they help others understand your work. When making comments, add enough so that you'd be able to understand your code if you looked at it 6 months from now. Don't stress too much about this, as it shouldn't take more than 10 minutes to comment all 7 of your files.

References

[1] Brian Drawert, Sean Matthew, Megan Powell, and Bryan Rumsey. Saving the devils is in the details: Tasmanian devil facial tumor disease can be eliminated with interventions, 2022.