

Overall Idea:

- A lot of times, the models that predict tumor volume during treatments are made using continuous ODE type equations
- But the treatments themselves are really fixed points in time that have sharp effects
- There are impulsive differential equations (IDEs) that are ODEs when $t \neq \tau$, for τ the time of treatment, and at time $t = \tau$ they directly change the volume (i.e. maybe $V = V(1 - r)$ where r is something determined by the amount of immediate change caused by the treatment at that time period)
- I think that it would stand to reason that on a large scale, these IDEs might have better predictive capabilities than continuous equations, but I want to try it out
- IDEs and the like have been used before, but not in a large scale comparison like this, based on real patient data (which we will snatch up from the hordes that NYU has)
- My thought is to essentially perform two experiments based on 6 different models in ODE and IDE form (so really 12 models if you want to look at it like that because each of the 6 will be tested in both forms)
 - I'll describe the two experiments in detail later, but they really follow the two main experiments laid out in this paper:
<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009822>
 - Using the same 6 basic models that paper (the Laleh paper) uses too (I think it's a really good paper generally!)
- The data will (hopefully) have tumor volume at different points in time as well as the time points when the treatment was given as well as the specifics on what treatment were given
 - Also in the above paper, they have a version of their data available, but it is sort of janky and weird. So worst case, we can try to work with that, but ideally we could steal some from NYU
 - Also, for now, I think working with synthetic data just for a workflow perspective probably makes sense!
- For the fitting of the models themselves, I think we can follow the same path that they put forth in that paper

Fitting and implementation

All model fitting procedures were implemented in Python 3.7. In particular, we used differential evolution to generate the initial data points for the differential equations. Differential evolution is a stochastic population based method which is used for global optimization problems [36].

Based on these initial guessed parameter values, the “Curve_fit” function (from the python package “scipy”) is used to fit the model parameters to the experimental data. This function uses the Trust Region Reflective (trf) algorithm with the non-linear least squares loss function to find an optimal fit of the model parameters to the data points. The inputs to this function are the sorted time and its corresponding tumor volume measurements, the respective mathematical function to fit the data and the maximum number of iterations (we used 1000 iterations in this study). The output of the “Curve_fit” function is the calculated optimum parameters for the selected mathematical function. Having these parameters, it is possible to predict the volume values for each time point and then evaluate the goodness of the fit. The source codes are publicly available at <https://github.com/KatherLab/ImmunotherapyModels.git>.

- publicly available at <https://github.com/KatherLab/ImmunotherapyModels.git> (here's the above link but actually as a hyperlink)
- Python should be able to do this for us (although if we want to use not Python that's fine by me, I just think it's got some useful libraries for some of this stuff and I'm not as familiar with other languages tbh)
 - Differential evolution to get initial guess parameters
 - Non-linear least squares loss function using the “scipy” package in Python with the trf algorithm to pick the parameters afterwards based on the datapoints given
- If we want to play with other methods of fitting the models, we can do that too! I just think it might be easiest to at least start here, but I'm not married to the idea
- The equations we want to use:
 - **Exponential** $\frac{dV}{dt} = rV$, where r is the rate of growth (so a birth rate minus death rate of the cells essentially)
 - **Logistic** $\frac{dV}{dt} = rV\left(1 - \frac{V}{K}\right)$, where K is the carrying capacity
 - **Gompertz** $\frac{dV}{dt} = rV \ln\left(\frac{K}{V}\right)$, here K is essentially still the carrying capacity, but it is an asymptote versus in the logistic where you'll actually reach K and then stop having any growth, but really similar idea
 - **General Gompertz** $\frac{dV}{dt} = rV \left[\ln\left(\frac{K}{V}\right) \right]^\alpha$, where α is there to be able to modulate the steepness of the curve as we start to approach K . It just

gives us some more control in the curve we're making really. When $\alpha > 1$ we have a more sharp slowdown as we get to K , versus if $\alpha < 1$ then it's a lot more gradual of a slowdown. α should always be positive.

- **Classic von Bertalanffy** $\frac{dV}{dt} = aV^{\frac{2}{3}} - bV$, where a is called an anabolic growth constant and b is a catabolic loss constant. In this case, since the a is attached to $V^{\frac{2}{3}}$, we're saying that growth is proportional to a surface area kind of thing of the tumor, and that the death is more proportional to just the volume itself per unit time
- **General von Bertalanffy** $\frac{dV}{dt} = aV^m - bV$, this is like the classic, but now we have m that lets us say that growth might be more proportional to the tumor in a surface area way, some sort of linear way, or any other host of weird ways that we can come up with that seem to fit the data more appropriately. It gets a little weird and convoluted here
- These should all be able to be loaded into Python already
- They are pretty standard ones used in biological modeling and have been used a lot in tumor modeling in particular
- They are also the 6 used by Laleh paper, like I mentioned above
- They are not super complex by any means, but they are common and give a good standard for comparing the ODEs and IDEs in my opinion
- For putting the IDE portion in, I would say we should do something so that at each point of time we are giving treatment (that could be once a day, 5 days a week, for 4 weeks or something for a radiation schema, or it could be like once a week for 6 weeks for some kind of immunotherapy, etc.), we do that at that specific one point in time:
 - $V_{+i} = (1-\delta)V_{-i}$, where V_{+i} is the volume after the impulse, V_{-i} is the volume right before the impulse, and δ is some factor that we correlate to the particular treatment for how much tumor death is occurring at that moment because of the treatment

- We would probably also determine δ in the same way mentioned above – although I need to give this a little more thought!
- Otherwise, at all other times, the IDEs follow the same way as the ODEs!

Experiment 1: Test for goodness of fit when applying each of the 12 models to the whole set of data

- Feed all data points to the model and appropriately select values for the variables to fit to the best extent that it can
- Probably evaluate goodness of fit using MSE or MAE or something; also probably want to make some kind of plot of residuals to see if it is massively horrible (i.e. there are patterns in there)
- This just sees if the models themselves have any resemblance to actual patterns, it isn't super useful, but is probably a good thing to include – less exciting than experiment 2 imo

Experiment 2: The big kahuna. This is where we test the predictive capabilities of each of the models

- We will give each of the models the first maybe 30% or so of the datapoints, have them fit the best values to the parameters that they can given that data, then we plot it out and see how well the values correspond to the future points we didn't provide
- Again we would probably look at MSE or MAE or something (but obviously only on the points that we are predicting, not on the ones we already gave to it)
- We might also be able to run this experiment a couple times giving different amounts of points (i.e. 30% of them versus 50% of them, etc.) before we make the thing predict stuff for us to see how it affects stuff

Conclusions:

- Determine who can be fit best (the IDEs or ODEs)
- Determine who has the best predictive ability (IDEs vs ODEs)
- See which of the 6 models is the best (Gompertz did the best in the Laleh paper, so it'll probably be our best one too for ODEs and IDEs)
 - General Gompertz and General Bertalanffy also did well for them, so we can probably expect similar results in that regard as well
- I think maybe we will see that the IDE versions might do slightly better in their predictions! If not, at least we will have learned something :)
- We might also consider doing a paired test using Wilcoxon signed-rank test (or can try a paired t-test but it depends on the data distribution) to see the ODE vs IDE comparison for each patient we look at – but we can look more at this later on once we get things going

Tasks:

- Create some false datasets of tumor volumes with pretend treatment regimens
- Set up all 12 of these models
- Run experiments 1 and 2
- Plot residuals, look at MSE, MAE and compare performance
- Plot the curves generated by each of the 12 models in experiments 1 and 2 compared to the real data