Week 6 Summary

Subteam 2

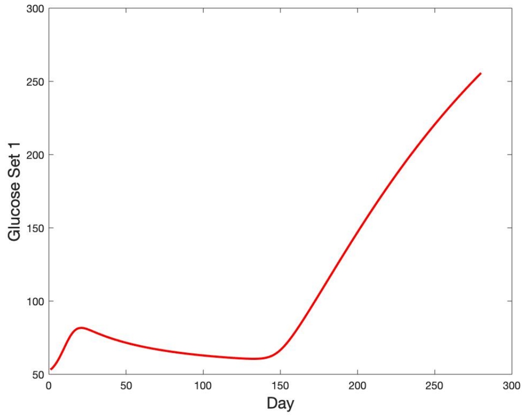
June 2020

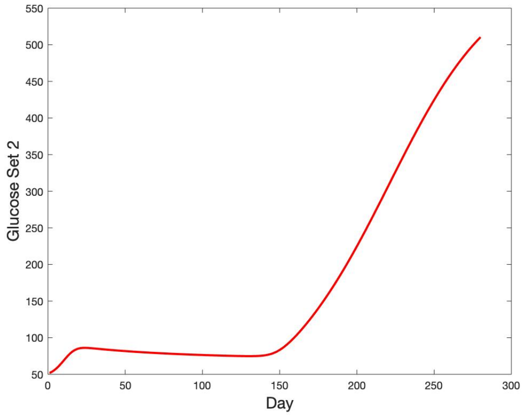
1 Introduction

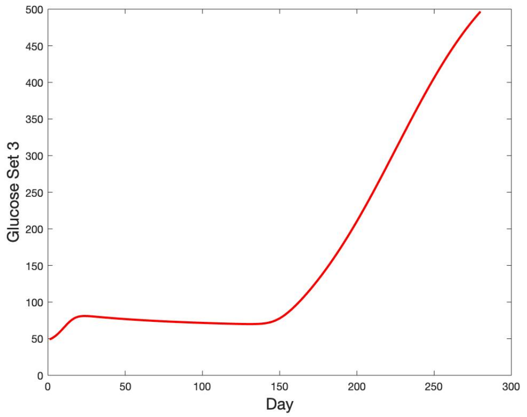
At the conclusion of last week, we were able to do parametrization using both the Joint and Dual UKF. However, the results we were getting were very inaccurate and biologically inconsistent with what we would have expected. In order to address this, this week we allowed for all parameters in the model to move, not just a select few. We first applied this technique to mouse 6 and then to other mice as well. In the process, we were able to learn a lot about the role of parameter variances in the model and their impact on performance. Additionally, we paid more attention this week to the biological aspect of the parameters we were getting in the end.

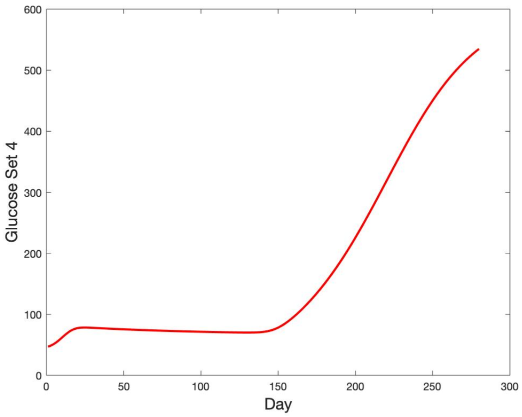
2 Initial Results When Estimating All - Mouse 6 Dual

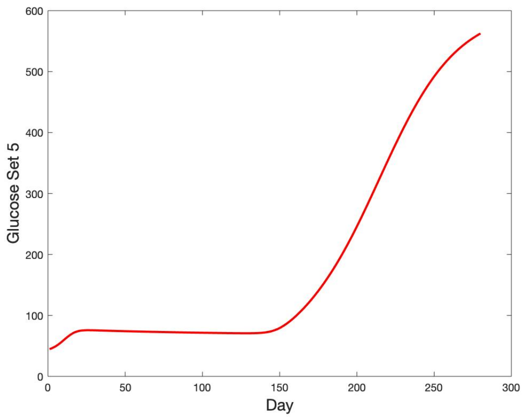
We proceeded to modify our Dual UKF algorithm to work for all parameters. This consisted of modifying matrix dimensions as well as assigning variances to the new parameters we had now added. In general, we wanted to keep variances low so to allow for "wiggle room" for parameters but not drastic shifts. To judge how well we were doing, we looked at plots of the raw glucose measurements as compared to the predicted glucose values using the final parameters (in future figures these will be on top of one another). The raw glucose data is the final figure presented. The algorithm was ran 5 on mouse 6 so to allow for ample time for the algorithm to work. The five glucose plots are found below:

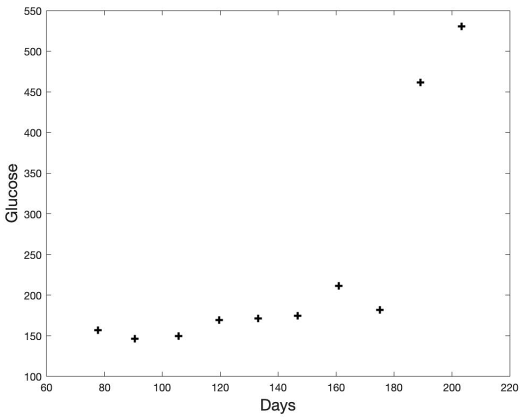












We can see that the main issues are 1) the first steady state is too low and 2) the jump to being sick happens too late.

3 Key Parameters

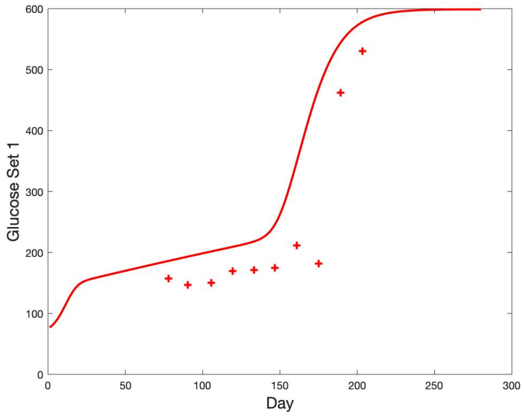
In order to address the 2 issues above, we needed to identify the parameters that are most likely to be causing the issues. In the ODE model, Professors Shtylla and de Pillis have created a variable, eta_{vary}, that is responsible for exactly our concern. It is defined as:

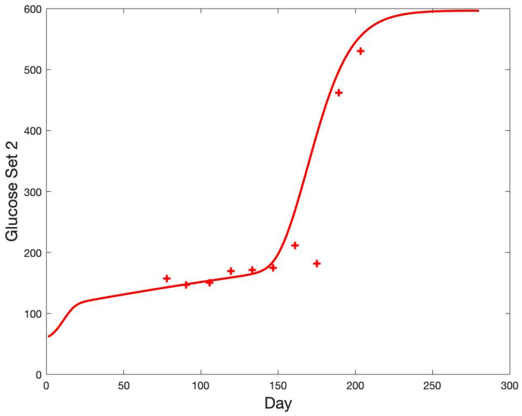
$$eta_{vary} = eta + 2 * eta * (1 + tanh(alpha_{eta} * (t - beta_{eta} * 7)))$$

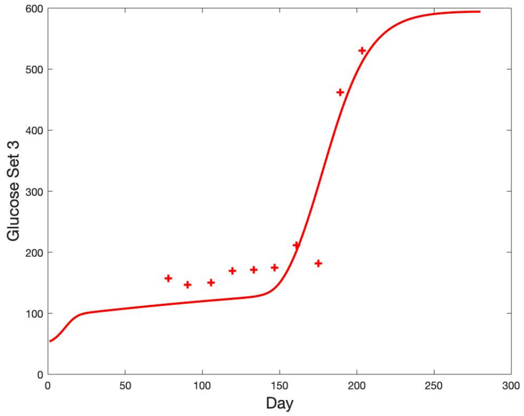
To begin with, the value eta itself is kept constant, but $alpha_{eta}$ and $beta_{eta}$ are both included in our set of parameters being estimated. Later on we began to move η itself around, which we will note once we reach that point. Since η was not yet moving, we placed extra emphasis on $alpha_{eta}$ and $beta_{eta}$. To do this we increased the variance of those two parameters, giving it much more room to move.

3.1 Results on Dual

The result, when running this on mouse 6 with the Dual UKF, is that we now got much better fits (important to note that at this point we switched from 5 to 3 runs in order to save computational time):







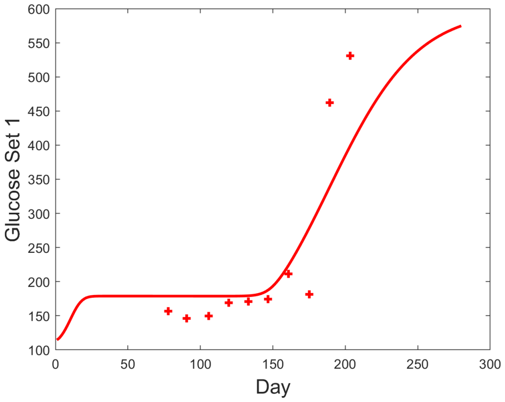
We can see that, as more iterations occur, the projected Glucose gets better and better at fitting the raw data. Apart from a visual check, it is also crucial to quantify the error of the glucose estimates. In order to do so, we use the Sum of Squares error. The error for each run is in the following table:

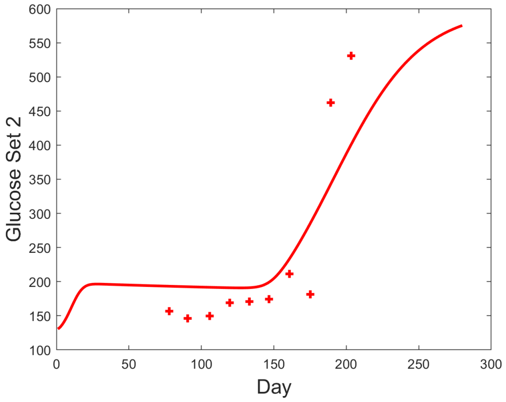
Run	Error
1	354.363
2	226.119
3	163.88

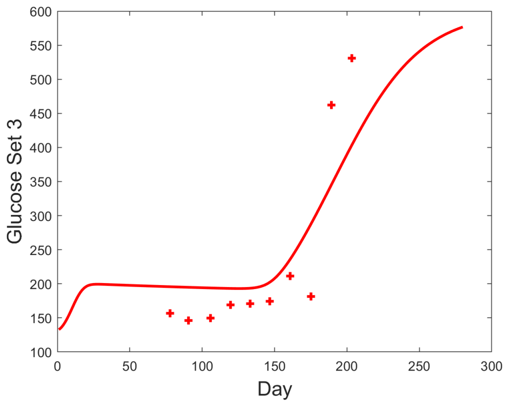
The decreasing error values confirm our visual analysis that as the number of runs increases, the error decreases.

3.2 Results on Joint

The joint was run on Mouse 6, using the same general set up, initial states and parameters, and covariance matrices as the dual. Three runs are shown, however the joint tends to hit it's best run before the third run.







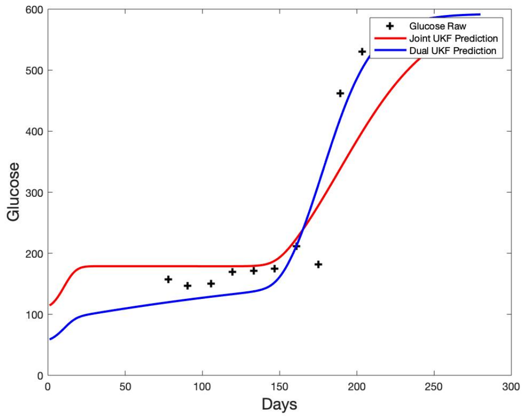
The error for each run is in the following table:

Run	Error
1	212.2441
2	221.7481
3	222.5329

We can see that the joint does indeed hit its best run the first time, but even though the error gets slightly bigger, all three errors cluster around the same value.

3.3 Comparison of Dual and Joint

In order to compare the performance of the algorithms, it is helpful to plot them together with the raw data as below:



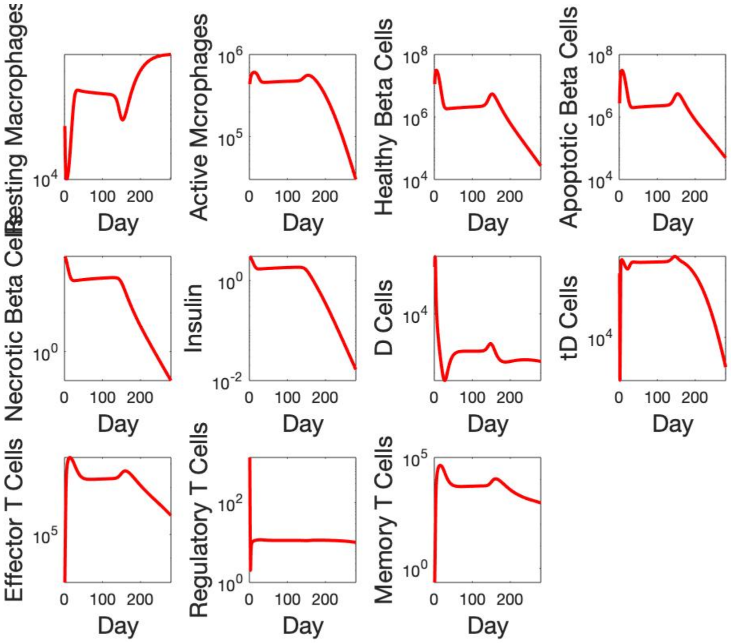
We can see that the dual appears to be performing better in this case, which is confirmed by its lower error value. This is most likely due to the amount of parameter movement. The final parameter values for the joint differ little from the baseline while they do moreso for the dual UKF.

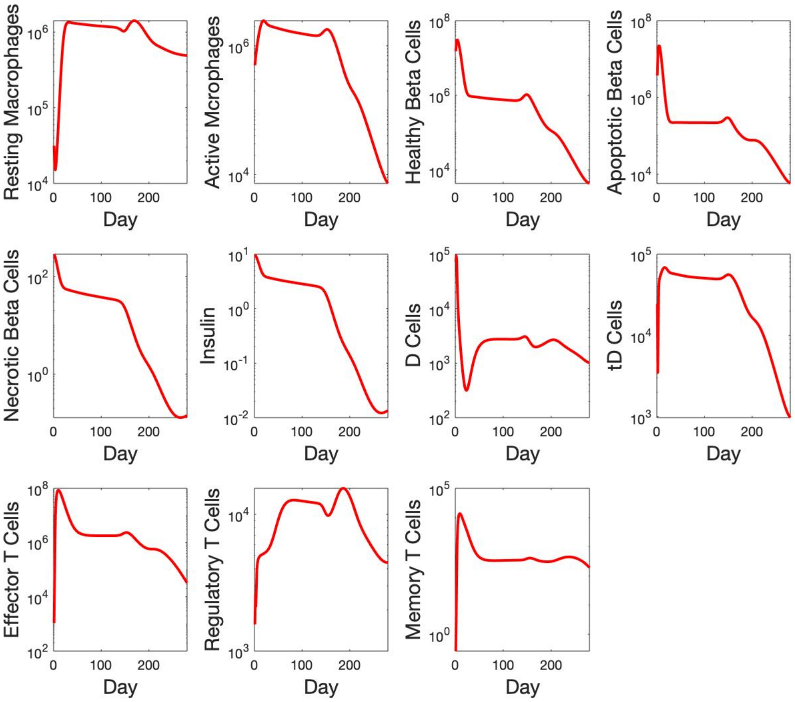
4 Other States

Up to this point we have solely been discussing our glucose estimates. However, although glucose is the only observable available to us, the parameters we generate also directly determine the behavior of the other cells in the pancreas. Thus, it is important to visually look at the plots of the other types of cells that our parameters generate in order to check their biological plausibility. As a benchmark, we will compare the plots we get to the plots generated using the baseline parameter values. Of course, we are not looking for the behavior to match exactly, but more so want to make sure our system is still biologically feasible, which we know the baseline system is.

4.1 Dual UKF

Below we can see both the states as generated by the UKF parameters (top) next to the states as generated by the baseline (bottom):

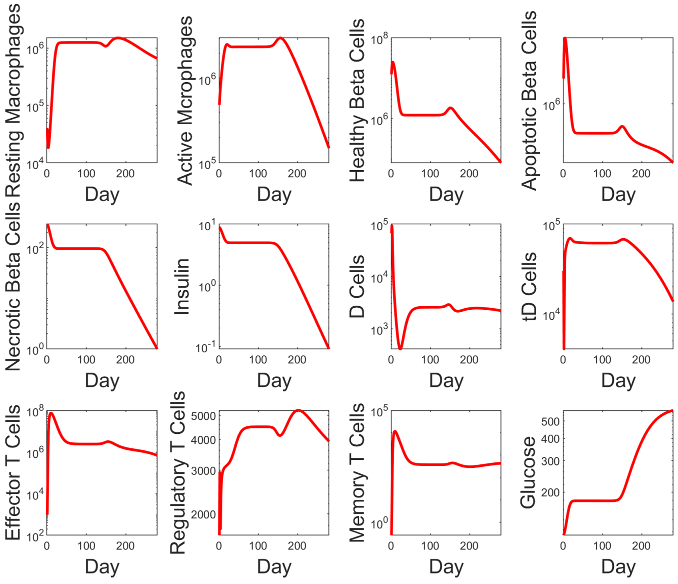


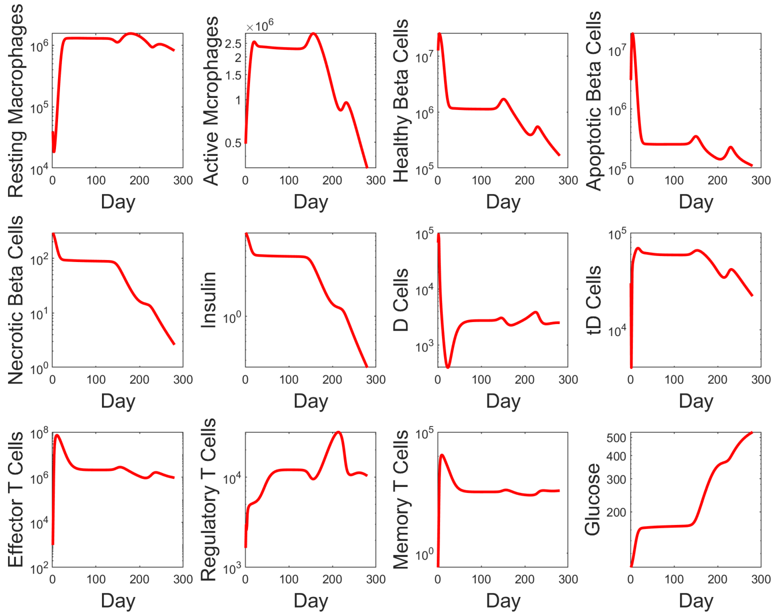


First off, it is assuring to see that none of our predicted states do anything extremely outlandish. The main difference we see is in the Regulatory T Cells. With baseline parameters the Regulatory T Cells follow more of a curved path, while under our estimates they flat line relatively early. A main parameter that controls the Regulatory, as well as Effector, is their rates of interaction μ_e and μ_r. In simplest terms, these control how good each type of T cell is at killing off the other. In the current set up, these are two separate parameters. However, in the baseline they are set to be equivalent. One idea we tried was thus to treat them as a single parameter in our UKF, however this unfortunately, and surprisingly, resulted in nearly identical results. Thus, T cell behavior continues to be one of the aspects needed to be explored further. Of course, there is a possibility that the results from the parameter estimate are actually feasible, however this could not be confirmed without raw T cell data.

4.2 Joint UKF

Below we see the states generated by the parameters from the Joint UKF followed by the states generated by the baseline parameters (for comparison).





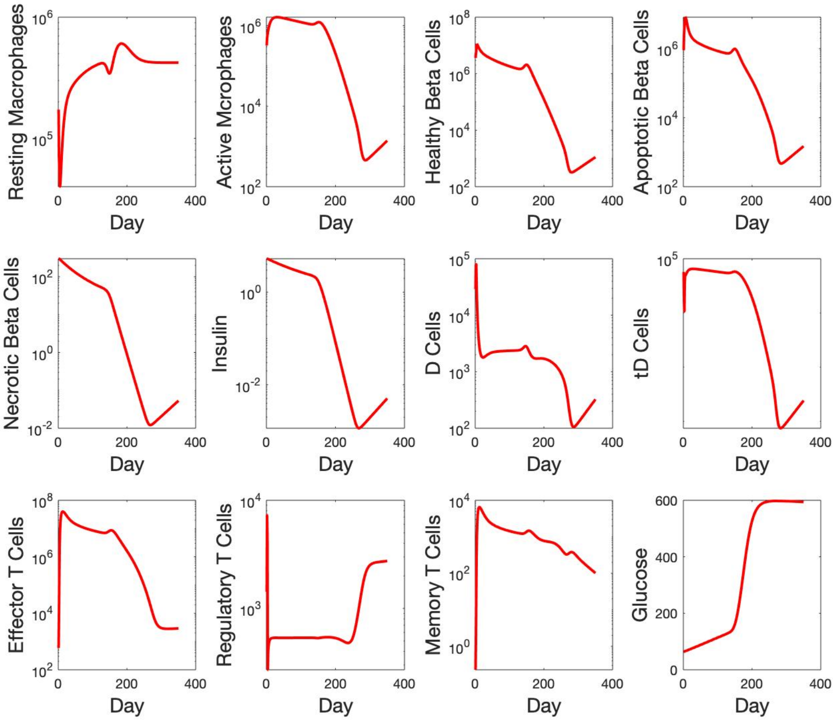
We see that in terms of state comparison, the joint is performing pretty well here. While some of the estimates are a little smoother and may lack a couple of small peak that the baseline results have, all of the state estimates look pretty similar and thee aren't any causes for serious alarm.

5 Biological Consistency

The T1D model parameters are meant to be "living on the edge". By this we mean that the mouse is meant to get sick when an apoptotic wave occurs but return to a healthy state if no wave occurs. Thus, our final parameter values must match this criteria. In order to test this, we can plot the 12 states of the model with the wave turned off. What we expect, particularly for glucose readings, is to see an initial bump but then a return to a healthy glucose state.

5.1 Dual

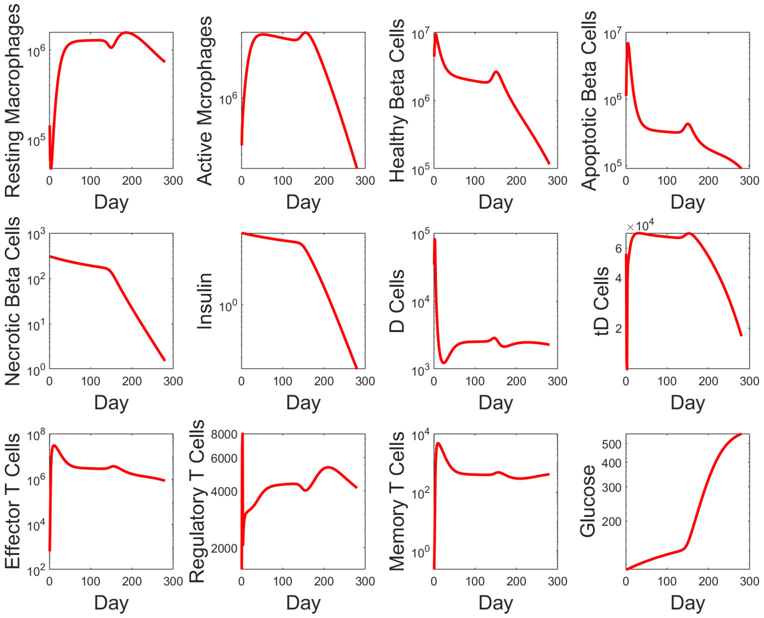
Below we see the states without a wave using parameters produced from the Dual UKF:

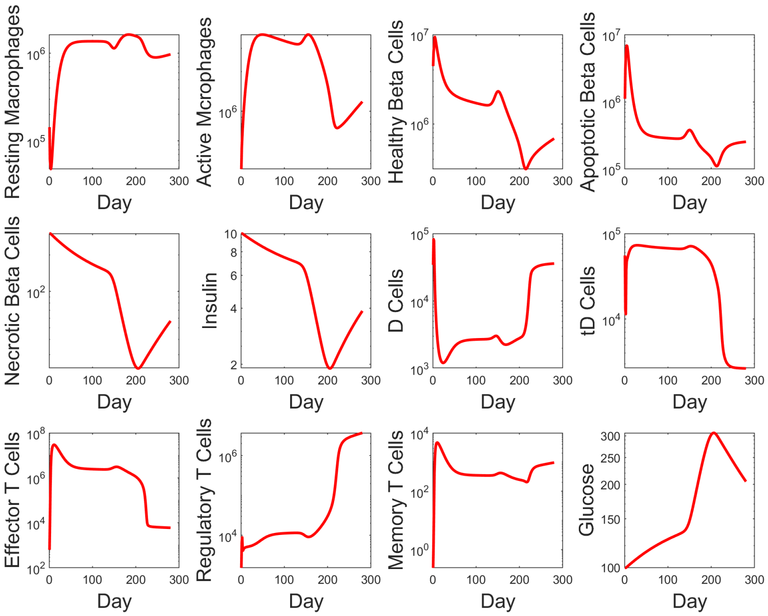


Looking at the bottom right panel, the glucose estimates, we unfortunately see that our mouse is still getting sick, even though no wave is present. Our current best guess is that this is due to the relationship between η and the macrophage clearance rates. When developing the baseline parameters, these were found to be extremely sensitive and, in the process of parametrization, it is likely that the relationship between them all is becoming distorted. Specifically, η is currently one of the parameters being estimated while the clearance rates are one of the few being held constant because of their extreme sensitivity. However, if η continues to be estimated, a decision will need to be made on how to move the clearance rates to more biologically acceptable values.

5.2 Joint

Even though the parameters from the joint have only moved slightly, are they biologically consistent? Below, we have results for running the ODE with both the final and baseline parameters, but this time without the wave.





We see that even though the parameters barely moved, the NOD mouse still gets sick even without the wave. This is an important reminder of how sensitive the system is. Even a small perturbation in some of the parameters can have a dramatic effect on the overall behavior.

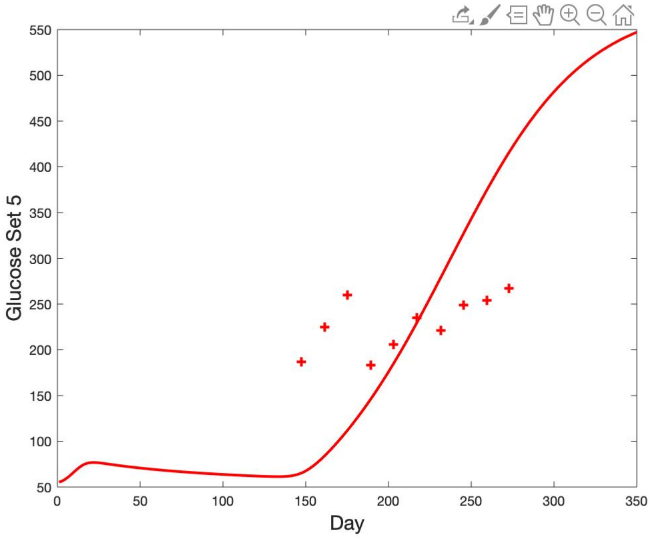
6 Other Mice Datasets - Dual

NOTE: At this point the variable η is now also being estimated instead of being held constant.

To first get the algorithm working we have focused our attention on mouse 6 due to its ”nice” behavior. However, we would like to validate our techniques on other mice as well.

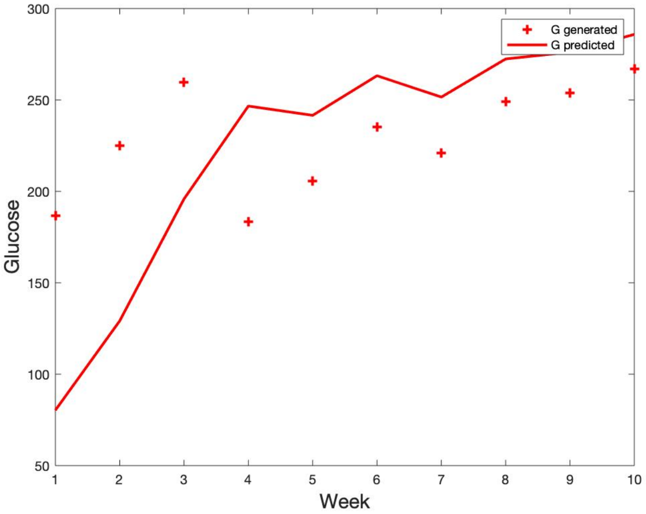
6.1 Progressive Mouse

For instance, there are two classifications of mice in the Li et al. cohort, progressive and acute. 9 of the 11 mice, including mouse 6, are acute due to their sudden onset of Diabetes. However, mouse 1 for example, progresses to a diabetic state much more slowly. Thus, we tested our algorithm on this case and found the following results:



Unfortunately, as is clear visually, our algorithm does not perform very well. The glucose fit we achieved still expects a sudden jump to a diabetic state, which does not occur in mouse 1. This is most likely due to the parameter region in which we are working. Based on initial parameter values, taken as the ”baseline”, variances are assigned to parameters, which determines the amount of room they have to move. Ultimately, in the case of a progressive mouse, we do not allow for enough movement since the parameters needed are very far from the baseline. When working with progressive mice it is likely most beneficial to develop a new starting baseline from where to begin the parameter search.

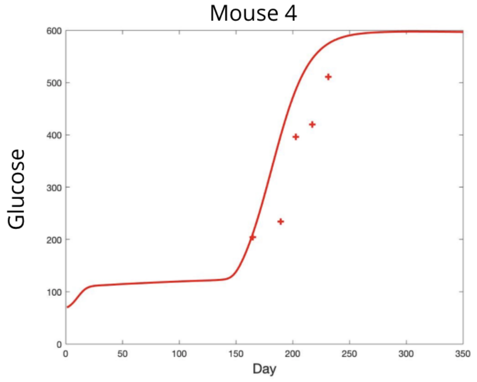
For this mouse in particular, it is interesting to look at the state estimations themselves. The state estimation for mouse 1 is found below:

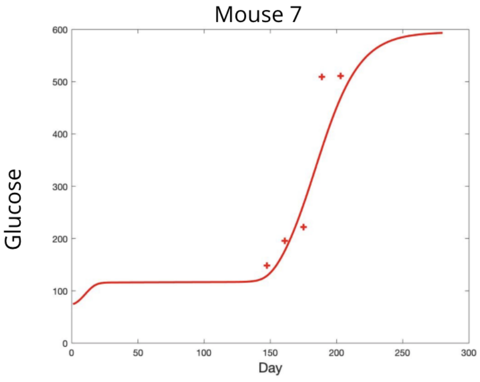


Although the state estimation does struggle, as expected, the algorithm does eventually learn that the slope of the glucose is not as drastic in this case. However, we do not see this learning reflected in the parameters themselves at this point.

6.2 Other Acute Mice

Since the majority of available data in the Li dataset is for acute mice, we thus have chosen to focus on these for the most part. Immediately, it was evident that we were not searching enough of parameter space to fit other acute mice. Since mouse 6 had optimal parameters relatively close to the baseline, we had been able to get away with low variances. However, we now needed to raise those values. After doing so, our fits improved tremendously. For example, we could get the following two fits for mice 4 and 7:

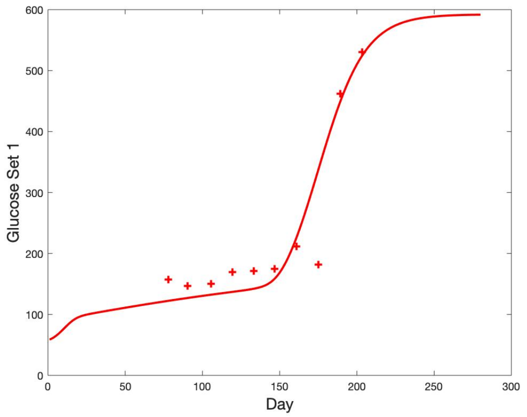


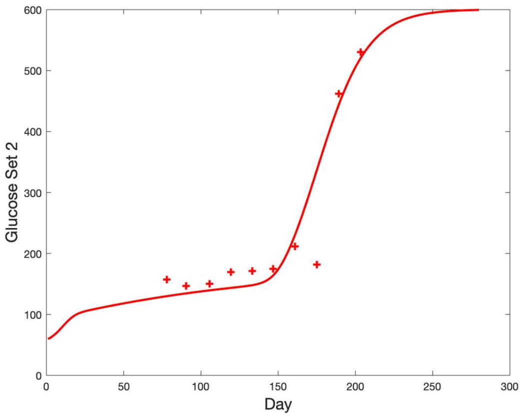


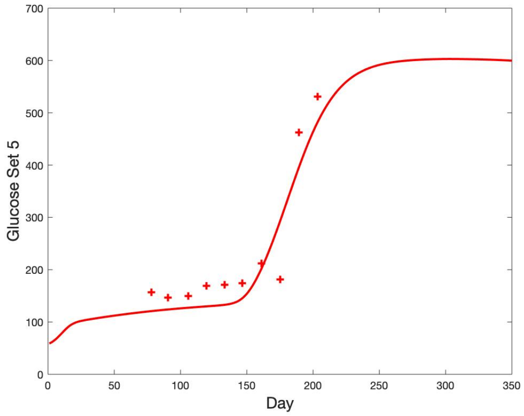
This figure exemplifies that the algorithm seems to work across a range of mice after variances are raised. However, it is necessary now to consider the possible costs of using too large of a variance. Intuitively, having a smaller search space would mean that, given the mouse has optimal parameters in that space, it would be easiest to find them. Thus, it would follow that if the search space (i.e. variance) increases, the quality of fits for mice that live closer to the baseline should decrease. As of right now, I am not able to give a definitive answer if this is the case. Rather, I currently have multiple examples in favor of this hypothesis and one against it.

6.3 Mouse 6 with Increased Variance

Let us begin with the example that goes against this hypothesis. From our previous work we know that the optimal parameters for mouse 6 live relatively close to the baseline. However, I now reran the algorithm on mouse 6 with the new, larger variances and got the following fits for the three iterations:







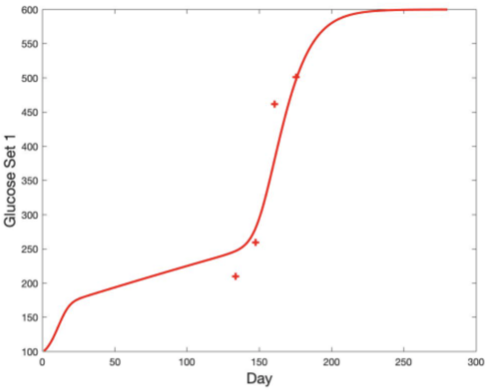
With error values:

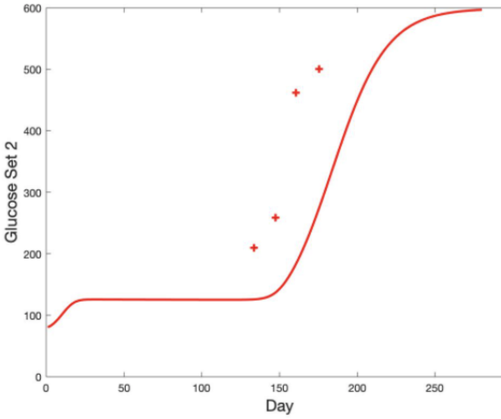
Run	Error
1	170.48
2	165.77
3	160.34

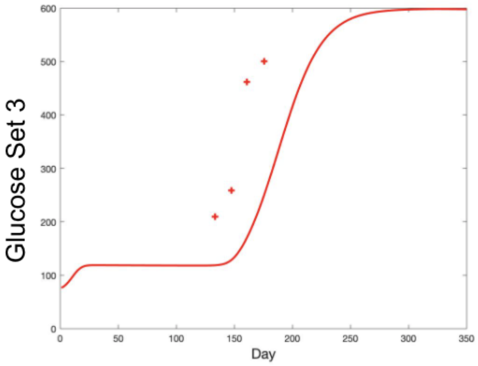
Surprisingly, the algorithm now finds a fit for the glucose both faster and with less error. This would support the claim that high variance values do not negatively impact performance. Although this is the case here, let's now look at some examples that may suggest otherwise.

6.4 Various Other Mice with Increased Variance

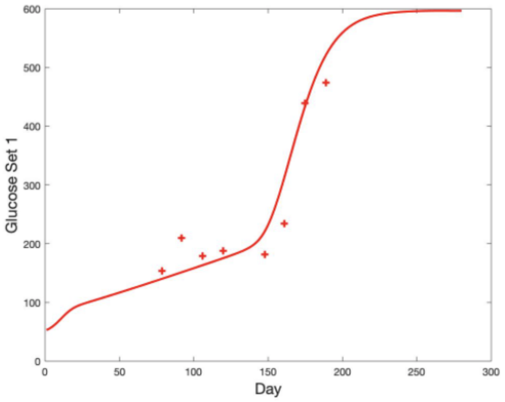
When looking at other mice we start a see a similar trend emerge: the algorithm finds a pretty good fit on the first iteration, but then when going to the second iteration and so on, the fits get much worse. The intuitive reasoning behind this is as follows: After the first iteration, the parameter variances are not changed. This means that the area of parameter space the algorithm looks stays relatively large. As a result, there is a high chance that the algorithm will leave the local area of parameter space where it found the "good" fit in iteration 1. For some examples of this concept, see first mouse 10:

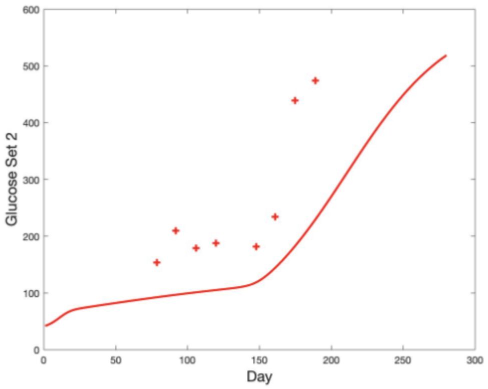


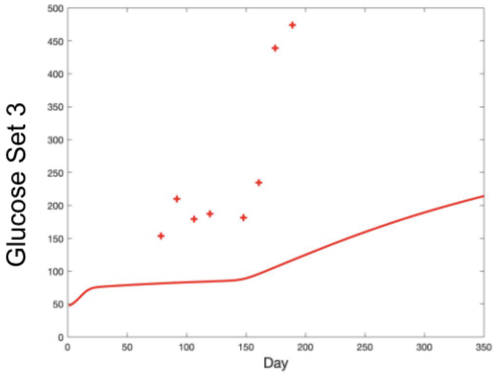




And then also Mouse 8:







As we can see, the result after run 1 is very good but their are drastic changes that drive the curve far away from the right area of parameter space.

6.4.1 Possible Solution

Currently, every iteration uses the same parameter variances. However, after initial movement on iteration 1, we may need to tighten the search area around that point. This could be achieved by systematically decreasing the variance values during each iteration, effectively urging the parameter values to converge around their current location. This is something that we would highly recommend to try moving forward.

7 Big Picture Next Steps

We now have an algorithm that can be applied to a wide range of acute mice and produce parameter estimates. This provides us with an opportunity to now try and combine these estimates with the MCMC algorithms. In particular, using the 9 parameter estimates, distributions of parameters can be made. These distributions, then, can be used as prior distributions for the MCMC algorithms. Having an informative prior is known to greatly improve MCMC performance and thus is a great possible application of the UKF algorithm.