Chapter five from the *Theoretical Neuroscience* book started the discussion about model neuron and introduced the basic integrate-and-fire model. The model brought concepts of electrical circuit and neuron’s membrane and synaptic conductance together. It introduces the properties of neurons in an electronic context where the neurons have intracellular and membrane resistance, capacitance and voltage dependence. The synaptic conductance could be solved like we solve for an equilibrium in a physics problem. The model that describes the membrane potential is a single-compartment model, which was a classic example we have practiced in the differential equation class. It is interesting to see the a series of variations derived from the basic model and how they could be applied to model the neuron membrane potential activities. The modeling I did in the lab is not so difficult as the models introduced in the book. The mnemonic model is a descriptive model where we use the model to help us make sense of the data and helps us to explain the story to others who are novel to the task. The model techniques in traduced in the book is constructed based on theory and our understanding of the neuron. The goal for these models is not only to describe the neuronal systems but also to predict the responses to certain stimulus.

Chapter five from the MATLAB book extended the discussion on good code documentation, which is a good reminder of my current task. I started to integrate timing information of practice sessions of the TMR study into the model last week. Within one week, I have created three folders with timing, linear regression parameter, and model fitting values information. All the text files contain just columns of numbers and good documentation would be very necessary in this case to keep track of the data and use them correctly. I have to make a separate “readme” file to record the column syntax and orders of all the new files I am generating. A large amount of my time last week was to carry out calculations with the new data and go back using a different set of number or a different equation to check the validity of the calculation. I spent a whole day trying to figure out why two sets of decay parameters generated from the model did not match when I thought they should be, but only to find out that the two numbers should indeed be different due to a minor difference in the order of the average calculation.

Chapter five of the MATLAB book also introduced a few visualization tools in MATLAB. During the lab meeting, we discussed that Python is used more often in the data science research field while MATLAB is a little limited in academia research only. So I think I will devote more time to python rather than MATLAB.

Last week, I was reconstructing the timeline for the first two experiment and now we have the practice, pre-nap test and post-nap testing events with a precision exact to ms. I then did a linear regression on the data of the practice session to predict a pre-nap test accuracy, which we hope would decrease the effect of measurement error. However, with this technique, the decay parameter would have to be subject and picture specific. Our next question is how to generalize the decay parameter estimation and get one graph or one TMR estimation for the whole population (entire data set of experiment one and two). I have not been able to move forward too much with the additional information in the model, because I tried to do several checks on any new numbers to make sure they are correct and make sense before using them in the consequent analysis.

When assisting the lab for the DARPAR biweekly report, I made a mistake on one graph Dr. Reber used in their progress report meeting. I had a typo in one line of the code which produced a memory trace graph with a wrong post-nap memory strength. That made me really nervous, but Dr. Reber said it was totally fine as the progress report was not very official and making mistakes showed that we were actually doing works. I found it nerve racking that when you work on your own project, you are the only one who is dealing with the data and with so many lines of code, it is so easy to make mistakes. What if the mistake happens at the first step when clean up the data, then every analysis afterwards would be untenable? How could we assure the research and analysis is foolproof?