MODELING THE GLUCOSE-INSULIN SYSTEM

In this assignment we explore how we can solve real world problems with computer science. The example we are investigating comes from human physiology, but the point is just that we are working with real data, so it does not matter if you don't know anything about this subject. We will provide you with all the background information you need.

For the human body to function normally, blood glucose levels must remain within a narrow range. During proper function, when glucose is ingested, B-cells in the pancreas secrete insulin, which signals to the muscles, liver, and adipose tissue that they should begin glucose uptake. When insulin is not secreted by the pancreas, or when it does not properly signal to increase glucose uptake, health issues like type 1 and 2 diabetes can occur[3].

To diagnose conditions like diabetes, clinicians may use the following markers to gauge insulin efficacy:
S_1: Insulin sensitivity index
S_G: Glucose effectiveness
ϕ_1: First phase pancreatic responsivity
ϕ_2: Second phase pancreatic responsivity

These markers can be obtained, for a given patient, as follows in a clinical setting: the patient is administered a glucose injection and then measurements of blood glucose and insulin values are taken at periodic intervals. These values are stored as data points, and then a system of differential equations is used to find a model with parameters that best fit these data points. The parameters of the best fitting equation are then used to estimate the above markers of insulin efficacy.

INITIAL INVESTIGATIONS OF SYSTEM

Differential equations allow us to explore and quantify the relationship between glucose and insulin within the human body. One such method to model the complex glucose-insulin system is the minimal model approach introduced by Pacini and Bergman[3]. This approach uses three equations:

$$\frac{dG}{dt} = -(p_1 + X(t)) * G(t) + p_1 * G_b, \quad G(0) = G_0 \qquad (1)$$

$$\frac{dX}{dt} = -p_2 * X(t) + p_3 * (I(t) - I_b), \quad X(0) = 0 \qquad (2)$$

$$\frac{dI}{dt} = -n * (I(t) - I_b) + \gamma * (G(t) - h) * t, \quad I(0) = I_0 \qquad (3)$$

Notice that unlike our example from the short assignment where the derivative *y'* depended only on *y,* the derivatives of these variables depend on the solutions to the other equations as well. This is called a system of coupled differential equations, since the variables of interest appear in more than one equation and you need all three to solve the system.

As you saw in the short assignment for the differential equation $y' = -ay + b$, changing the parameters *a* and *b* affected the shape of the resulting curve. In the glucose-insulin system case, there are quite a few more parameters (p_1, p_2, p_3, n, γ, h, I_b, and G_b) but they have the same effect of changing the shape of the solution curve, and we can utilize this to fit curves based on the data points for a specific person.

Below are some graphs representing *G(t)* and *I(t)* for two different people: one with normal glucose-insulin function and one with impaired function. The parameters for the person with normal function are listed as well.

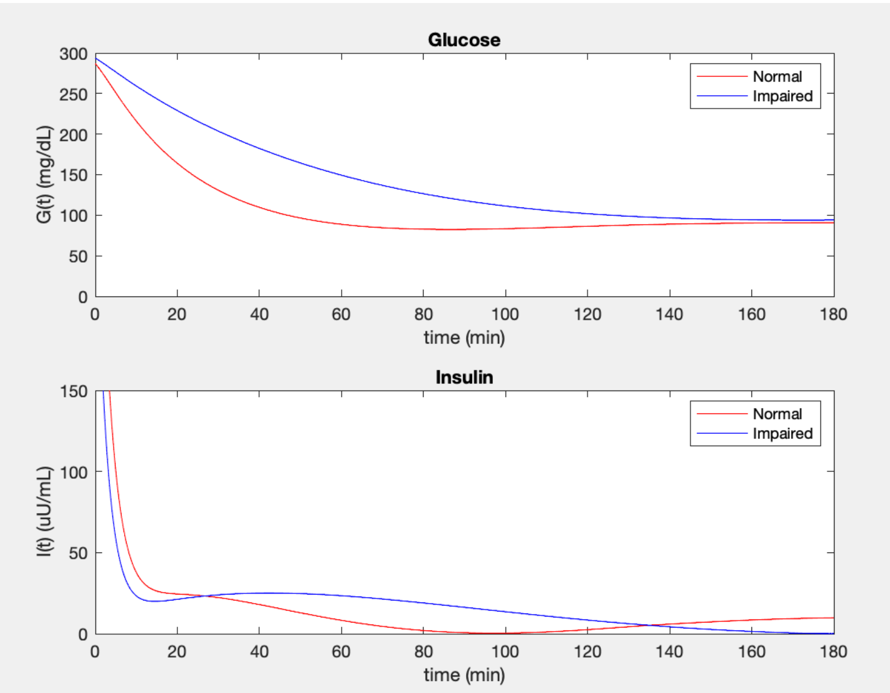


Figure 1: *G(t)* and *I(t)* are plotted for both a person with normal glucose tolerance and a person with impaired glucose tolerance. Adapted from [3]. The parameters used in the equations for a normal person (below):

```
%Normal Function Parameters
Gb = 92;
Ib = 7.3;

n = 0.3;
g = 0.003349;
h = 89.5;
p1 = 0.03082;
p2 = 0.02093;
p3 = 0.1062 * 10^-4;
```

Use the above graphs to answer the following questions:

1. Comment on the slope of *G(t)* for the normal function versus impaired function. Why might this difference occur?

2. Take a look at the parameter p_1 and the equation for *dG/dt* (Equation (1)). Based on your exploration of changing *a* and *b* in the short assignment, is p_1 more similar to *a* or *b*? What effect do you think increasing p_1 would have on the shape of *G(t)*?

3. The relationship between the parameters and the markers and normal ranges for glucose-insulin function are as follows:

$S_1 = \frac{p_3}{p_2}$ normal range: 4.0×10^{-4} to 8.0×10^{-4} (unitless) [3]

$S_G = p_1$ normal range: Greater than 2.10×10^{-1} (1/min) [1]

$\phi_1 = \frac{I_{max} - I_b}{n\,(G_0 - G_b)}$ normal range: 2.0 to 4.0 min*(µU ml^{-1})(mg dl^{-1})$^{-1}$[3]

$\phi_2 = \gamma \times 10^4$ normal range: 20 to 35 min^{-2} *($\mu U\ ml^{-1}$)($mg\ dl^{-1}$)$^{-1}$ [3]

Calculate these markers (you can use code) for the person with normal function and verify that they are in fact within the normal range. (Note that the variable g from the code in figure 1 represents γ and that you can estimate your value for I_{max} using the Insulin graph)

LONG ASSIGNMENT 3

PART 1

Write a Matlab function that performs Euler's method for the system of 3 equations used to model the glucose insulin system. Use your resulting solution to plot *G(t)* and *I(t)* on different subplots in the same figure (Note that you still have to calculate *X(t)* to solve the system of coupled differential equations, but we are not asking you to plot it). Verify that your function is working correctly by using the parameters specified in the examples above and checking whether they match the graphs provided in the example. Include your plots in your final deliverable. Use a Euler's method step size of h = 0.01 and plot from t = 0 to t = 200. One thing to note is that glucose and insulin values cannot be negative. So you should have a check in your loop and if any of the functions *G(t), I(t), or X(t)* ever go negative at a point t_i, you should set them equal to zero at that point.

PART 2

For part 2, imagine that you are working with a physician to help evaluate the insulin efficacy in patients. You will receive data points of glucose and insulin values at different time points and you will need to figure out which parameters best fit the data so that you can advise the physician on the markers for insulin efficacy (S_1, S_G, ϕ_1, and ϕ_2).

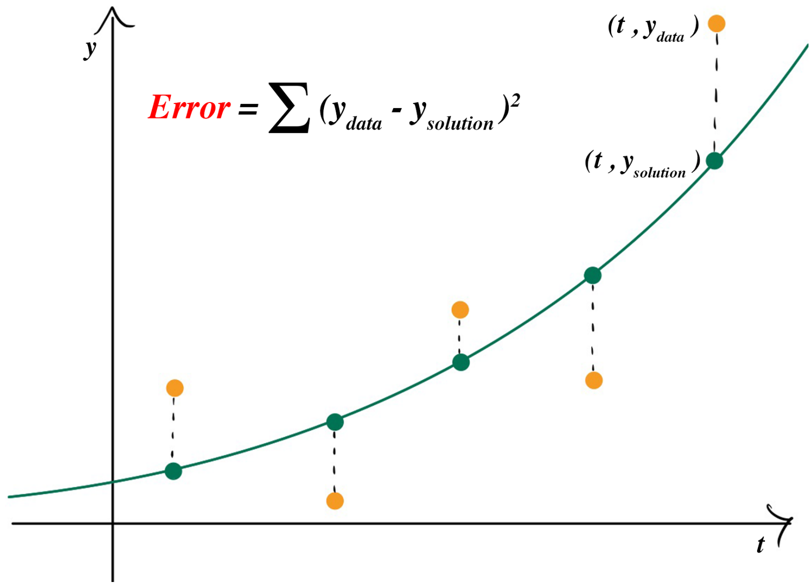
First load the provided data from Glucose.txt and Insulin.txt into Matlab. This is the patient data for Glucose and Insulin levels over a 200 second time period. You have worked with .txt files before, but using them in Matlab is different than in C. Use Matlab's help function to figure out how to use the *dlmread* function.

For this part of the assignment, you will need to find the parameters that best fit this new set of patient data. To help you, you are already given all but one of these parameters, namely:

$G_b = 95.4$	$\gamma = 0.1785 * 10^{-2}$
$I_b = 9.9$	$h = 0.1055 * 10^{3}$
$G_0 = 0.2938 * 10^{3}$	$p_2 = 0.1301 * 10^{-1}$
$I_0 = 0.241 * 10^{3}$	$p_3 = 0.4031 * 10^{-5}$
$n = 0.3606$	

Your job is to use the power of Matlab to find the best value for the parameter p_1 so that you have curves for *I(t)* and *G(t)* that best match the provided patient data. **To simplify your search, you can look for the**

best p_1 value in the range 0.01 to 0.02 in increments of 0.0001.



To evaluate how well your curve fits the data points you will use a method called *least squares* to calculate the error between your approximation and the data. This method is used in data science to find the curve of best fit to given data points. For each p_1 value, you calculate the difference between the observed y-coordinate value (provided as data for an impaired person and denoted y_{data} in the image above) and the calculated y-coordinate value (obtained through your solution of the differential equations and denoted $y_{solution}$) for each data point. You will then square each difference and sum them to find the squared error.

$$error = \Sigma\left(y_{data} - y_{solution}\right)^2 \qquad (4)$$

We want to minimize this error to find the *least squares* solution. So once you have identified the error value for each parameter choice, your objective is to identify which value for p_1 corresponds to the smallest error and therefore creates the best fit. (This is a useful model, but is simple compared to subsequent advances in the field, so it is not necessarily the most accurate. Our solution may not be completely accurate for the entire 200 second period).

Finally, calculate the markers for insulin efficacy (S_1, S_G, ϕ_1, and ϕ_2) using the equations from your exploration and your found value of p_1 and the other parameters. Display this information as an output. If any of the markers are outside the normal range as defined above, your program should print a warning message that lets the physician know which markers are abnormal.

Hints and Additional Instructions

- It may be useful to start with a smaller Euler's step size or increments for p_1 while testing your code early on since the smaller step size you use, the longer your code will take to run.
- Your script should call at least one Matlab function you have written.
- You will have more t values for your solution than for the patient data. Make sure to calculate your error using the difference between y_{data} and $y_{solution}$ for the values of t that match the patient data. You will have to think about how to find the correct index in your solution array.
- You should create some meaningful plots using the solutions you find for *I(t)* and *G(t)*. You could choose to show multiple curves in one plot, or use subplots, for example. Be sure to use excellent

labeling, titles, colors, markers, a legend, etc. Feel free to use the Matlab help function to find more features of the plot function in addition to those you learned in class.

- When plotting I(t) it is ok for the vertical axis to only go to 150 uU/mL like in the example graph in the exploration questions.
- Make sure to find your solution and then plot it (i.e. do not plot each point within a loop).

Part 3: Extra Credit Options

- Create a moving slider for values of the parameter p_1.
- Matlab has some built-in ODE solvers. Find one and figure out how to use it for the equations. **Hint:** Use the Matlab help documentation! Search for examples and explanations of ODE solvers to investigate the types of input values that can be assigned to ODE solvers. In order to ensure you do not obtain negative values for G(t) and/or I(t), you must first use the *odeset* function to evaluate one of the ODE solver inputs, *options*.
- Print out green messages for markers within normal range, and red warning messages for markers outside of the normal range.
- Write code to identify two parameter values using the least square method.
- The current method used to find the parameters of searching through each value can get pretty slow if you want to be extremely accurate. Implement a more efficient way to solve for the best parameter value. Consider exploring the Newton Raphson method (https://brilliant.org/wiki/newton-raphson-method/) and make sure to cite your sources.

References

1. C. Lorenzo, L. E. Wagenknecht, A. J. Karter, A. J. G. Hanley, M. J. Rewers, and S. M. Haffner, "Cross-sectional and longitudinal changes of glucose effectiveness in relation to glucose tolerance," *Diabetes Care*, vol. 34, no. 9, pp. 1959–1964, 2011.

2. P. Palumbo, S. Ditlevsen, A. Bertuzzi, and A. De Gaetano, "Mathematical modeling of the glucose–insulin system: A Review," *Mathematical Biosciences*, vol. 244, no. 2, pp. 69–81, Jun. 2013.

3. Pacini, G., & Bergman, R. N. (1986). "MINMOD: a computer program to calculate insulin sensitivity and pancreatic responsivity from the frequency sampled intravenous glucose tolerance test," *Computer Methods and Programs in Biomedicine*, vol. 23, no. 2, pp. 112-122, 1986.

4. *The Difference between Type 1 and Type 2*. Joslin Diabetes Center. https://www.joslin.org/patient-care/diabetes-education/diabetes-learning-center/difference-between-type-1-and-type-2