Project Milestone 3 – Algorithm Evaluation and Improvements

Instructions

1. Read this document carefully. You are responsible for following all instructions in this document.
2. Read the Learning Objectives at the end of the document to understand how your work will be graded.
3. Use professional language in all written responses and format all plots for technical presentation. See EPS01 and EPS02 for guidelines.
4. Apply good programming standards to all m-files.
5. Submit deliverables to Gradescope. Name your files to match the format in the table below, where *SSS_TT* is your section and team ID (e.g., 001_03 is Section 001, Team 3)

Item	Deliverables
M3 Answer Sheet	M3_*SSS_tt*.pdf
M3 Algorithm	M3_Algorithm_*SSS_tt*.m
Reference Enzyme Analysis	M3_MM_PGOX50_*SSS_tt*.m
M3 Main Function	M3_main_*SSS_tt*.m

   See submission requirements on the last page of this answer sheet.

6. Complete the Assignment Header before starting the answer sheet.

Assignment Header

Section and Team ID (SSS_TT):	019-24

Team Member Name	**Purdue Career Account Login**
Sergio Monge	smonge
Greg Szymchack	gszymcha
Seena Pourzand	spourzan
Nathan Thorson	njthorso

Role of Each Team Member

In this section, put each team member's name who worked on this milestone. In the Detailed Description of Work, each person on the team should write their own description of how they contributed to this milestone. Be very detailed here. Then in the last column, your team should estimate the percentage of the work that each team member did on the milestone. This column needs to add up to 100%. We know that on any given milestone that this will vary, but one person in the team should not be doing significantly more than the others throughout the whole project. Use this column as a way for you to make sure your workload is balanced throughout the project.

Team Member Name	**Detailed Description of Work**	**Percent of Work**

Sergio Monge	General revision of document. Contributed on the feedback review and improvements section. Revised code and general commenting.	25%
Greg Szymchack	Formatted the figure display of the PXOX50 models to make for easier reading, tested the program to defects and or inefficacies that would harm the overall running of the program, and basic grammar edits	25%
Seena Pourzand	Lead the effort of creating the main, revised m3, and pxox50 functions. Helped with reflection as well	25%
Nathan Thorson	Revised writeups, helped with team planning efforts, formatted template. Code editing and revision.	25%

Part 0: M2 Feedback Review

Reflect on your M2 feedback for the purpose of improvement. Your reflection should provide a clear, useful summary of your M2 feedback and provide a clear and practical plan to address the issues. Complete table 1 below.

Table 1. Feedback summary and plan

Part A: Based on your feedback from M2, identify at least one strength and one limitation of your team's algorithm you created in M2. Consider how the feedback from M2 could lead to improvements in your work. One of our strengths in our team algorithm was that the code was very organized, easy to understand, and give out the desired outputs for the client. As previously stated, we compared various approaches for the parameter identification and ended up with a fairly straightforward algorithm. One of our limitations was our decision to hardcode the substrate concentration (Ex. [3.75 7.5 …. 2000]) which could pose problems for datasets that do not follow a similar concentration pattern.
Part B: Explain how you will incorporate the M2 feedback to improve your parameter identification algorithm (do not just reword your response from Part A, include concrete actions you will take to improve). As mentioned above, our main source of feedback was the hardcoding made in our UDF for the substrate concentration. We can very easily fix this by creating a vector that obtains the substrate concentration from the dataset and passes it in as parameters into the UDF. An Issue that we might face but wasn't mentioned in our feedback is that our current UDF operates with duplicate tests. However, the PGO-X50 dataset only contains 10 tests and no duplicates. If we were to pass in the enzyme matrix from PGO, it would have faulty results or throw an error as the function expects there to be 10 more columns/tests that it would use to average with the original 10 tests. To avoid this, we will create our algorithm to work without duplicates for this example as there are none in this case.

Part 1: Algorithm Development

In Milestone 2, you developed an algorithm for identifying the enzyme parameters v_{0_i}, V_{max}, and K_m. Incorporate any improvements based on the feedback you received on M2 and the information in Table 1. Name the updated algorithm **M3_Algorithm_*SSS*_*tt*.m**. The algorithm function must have appropriate inputs and outputs.

Part 2: Output Comparison

The next step is to examine how well your algorithm performs using real data. NaturalCatalysts has provided data in a .csv file for a well-understood enzyme in their catalog, PGO-X50. You will use these data to examine the performance of your algorithm.

Part 2A: Understand Michaelis-Menten model for PGO-X50

Before you evaluate your algorithm, you need to understand how well the PGO-X50 data follow the Michaelis-Menten model. This information will help you understand how much error is associated with the data provided versus the parameter identification of your algorithm. Part 2A is just looking at how much error is inherent in the data, as the PGO-X50 enzyme is well-known. Therefore, in Table 2, you will find the expected parameters for PGO-X50 if the data perfectly follow the Michaelis-Menten model. You will use these reference values for the PGO-X50 enzyme following the procedures described below to determine the error in the measured data provided to you from NaturalCatalysts.

Table 2. PGO-X50 reference values

Parameter (μM/s)	PGO-X50 Reference Values
v_{0_1}	0.025
v_{0_2}	0.049
v_{0_3}	0.099
v_{0_4}	0.176
v_{0_5}	0.329
v_{0_6}	0.563
v_{0_7}	0.874
v_{0_8}	1.192
v_{0_9}	1.361
$v_{0_{10}}$	1.603
V_{max}	1.806

K_m (µM)	269.74

Use these reference values to create a user-defined function, named **M3_MM_PGOX50_*SSS_tt*.m**, that quantifies how well the PGO-X50 data follow the Michaelis-Menten model. This function must perform the following analyses.

1. Create a figure or set of figures to display the ten concentration curves from the given data (found in Data_PGOX50_enzyme.csv). Overlay the initial velocity, v_{0_i}, tangent lines on these data. Because of the large number of data points and because the data points are sequential, you may display the measured data with a line (without using markers too). Use different styles to differentiate between the data and the tangent line. When deciding on number of figures and plots, consider what may be useful for the client to see.
2. Create a figure that displays the initial velocities (v_{0_i}) as a function of substrate concentration for the data. Plot these 10 points with data markers, since they represent the original data. Using the reference V_{max} and K_m, calculate a vector of reaction velocities using the Michaelis-Menten equation (see the *ENGR 132 Project – Background and Schedule* document). Plot the model (using a solid line) along with the ten points (using markers).
3. Estimate the error between the reference initial velocities (v_{0_i}) and the reaction velocities (v_i) predicted by the Michaelis-Menten model. Use the Sum of Squared Errors (SSE) method.
4. Analyze how well the PGO-X50 data follow the Michaelis-Menten model using evidence-based rationales.

This MATLAB reference page shows how to add Greek letters and subscripts to plot displays.

Report your figures and SSE values in Table 3. Justify your plot display choices for Steps 2A.1 and 2A.2.

Table 3. Results for Part 2, Section A

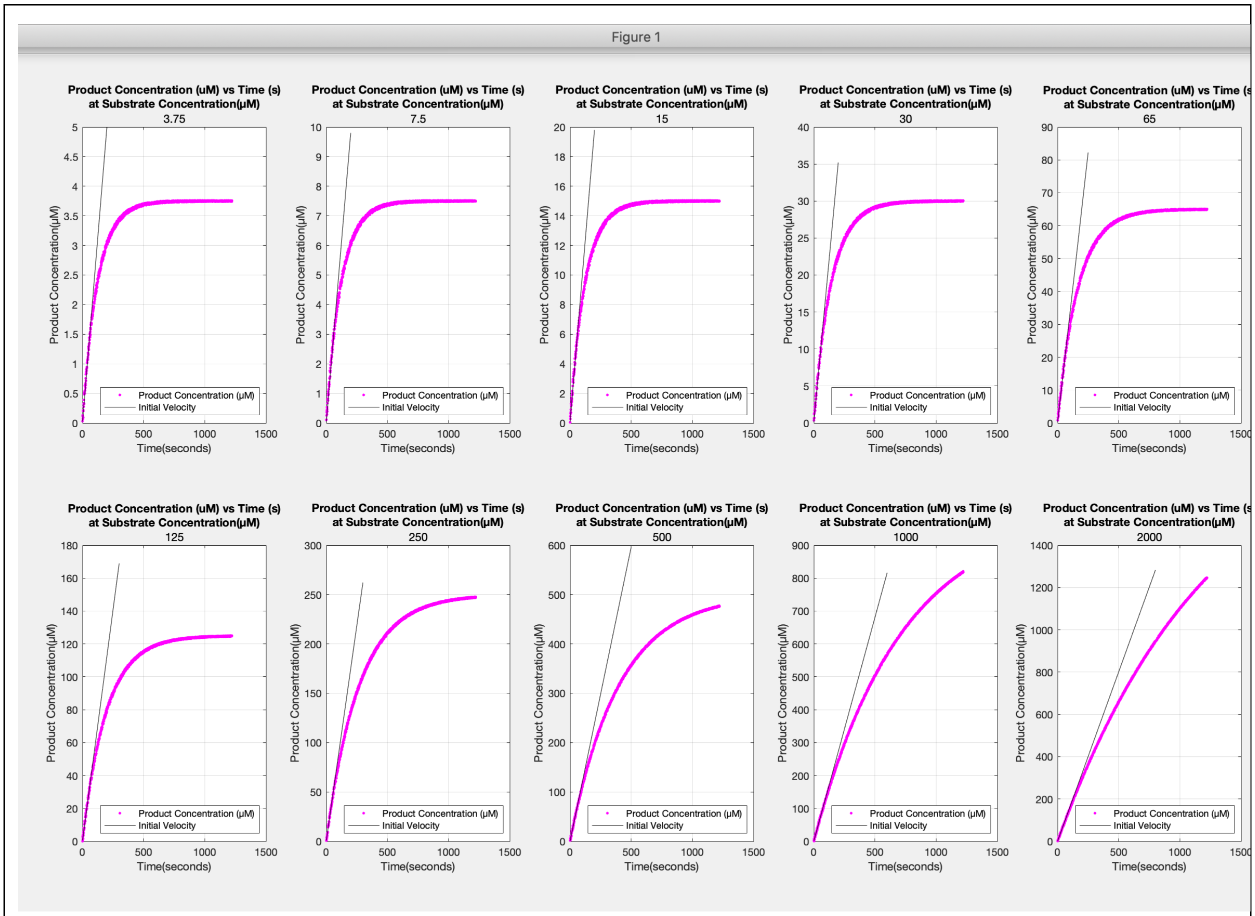
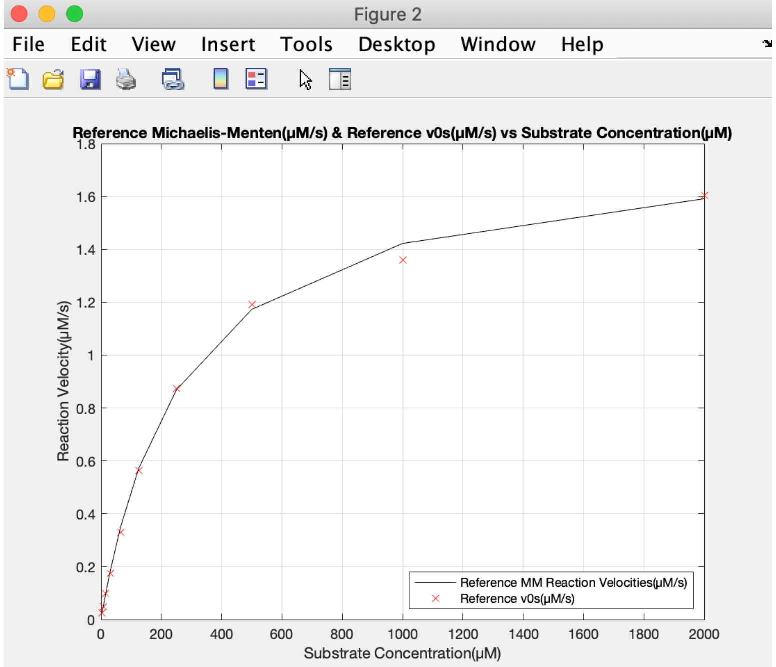


Figure 1
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 3.75
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 7.5
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 15
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 30
Product Concentration (uM) vs Time (s at Substrate Concentration(μM) 65
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 125
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 250
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 500
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 1000
Product Concentration (uM) vs Time (s at Substrate Concentration(μM) 2000
Product Concentration(uM)
Time(seconds)
0
500
1000
1500
Product Concentration (μM)
Initial Velocity
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
0 1 2 3 4 5 6 7 8 9 10
0 2 4 6 8 10 12 14 16 18 20
0 5 10 15 20 25 30 35 40
0 10 20 30 40 50 60 70 80 90
0 20 40 60 80 100 120 140 160 180
0 50 100 150 200 250 300
0 100 200 300 400 500 600
0 100 200 300 400 500 600 700 800 900
0 200 400 600 800 1000 1200 1400



Justify your plot display choices for Steps 1 and 2: We decided to use one large figure full expanded that had 2 rows and 5 columns to display our graphs as we felt it did the best job displaying the data. We experimented with 5 rows and 2 columns but for our graphs were stretched too much vertical by MATLAB and how it decides to display things. We opted to have one figure to show all of the concentrations and v0s as we felt having it all in one figure as opposed to multiple makes it easier to understand from the perspective of the user as makes a more unified display. Displaying it in multiple figures could make the user assume they are not related and parts of separate calculations/steps. As for the second figure, our decision to display it the way we have was much more straight forward as it simply is just the graph of the Michaelis Menten equation using the reference parameters vs the reference v0s, we followed the instructions of using markers for our data and a solid line for the reference model.
SSE for data: **0.0049**
Analysis of SSE: When computing SSE, the lower the value the better the fit. Personally, we believe the SSE is good as while extremely close to zero, and the closer to zero the better our ability to predict accurately from the model is. Based on our 0.0049 SSE, we can tell that the values that were determined by the MM equation closely resemble that of the reference values for the v0s. This is also supported by the second figure as the markers are extremely close to, if not laying on, the line of the model.

Part 2B: Main function using your algorithms

Now you must create a main function that will call your algorithm UDF (from Part 1) so that you can examine how well your algorithm identifies the parameter values from the Michaelis-Menten model, as compared to the reference values given to you in Part 2A. Using the data for PGO-X50, determine the enzyme parameters using your algorithm. Name the main function **M3_main_*SSS*_*tt*.m**. Remember, a main function has no input arguments and no output arguments. The function must perform the following tasks:

1. Load the PGO-X50 data.
2. Call your parameter identification algorithm to determine the parameters for the PGO-X50 data.
3. Plot the data and the Michaelis-Menten model using the steps 1 and 2 described in Part 2A (above). Remember to consider readability and usefulness to the client.
4. Determine the SSE for the PGO-X50 data.

Report your figures in Table 4.

Table 4. Figure displays for your Algorithm

Figures for Algorithm

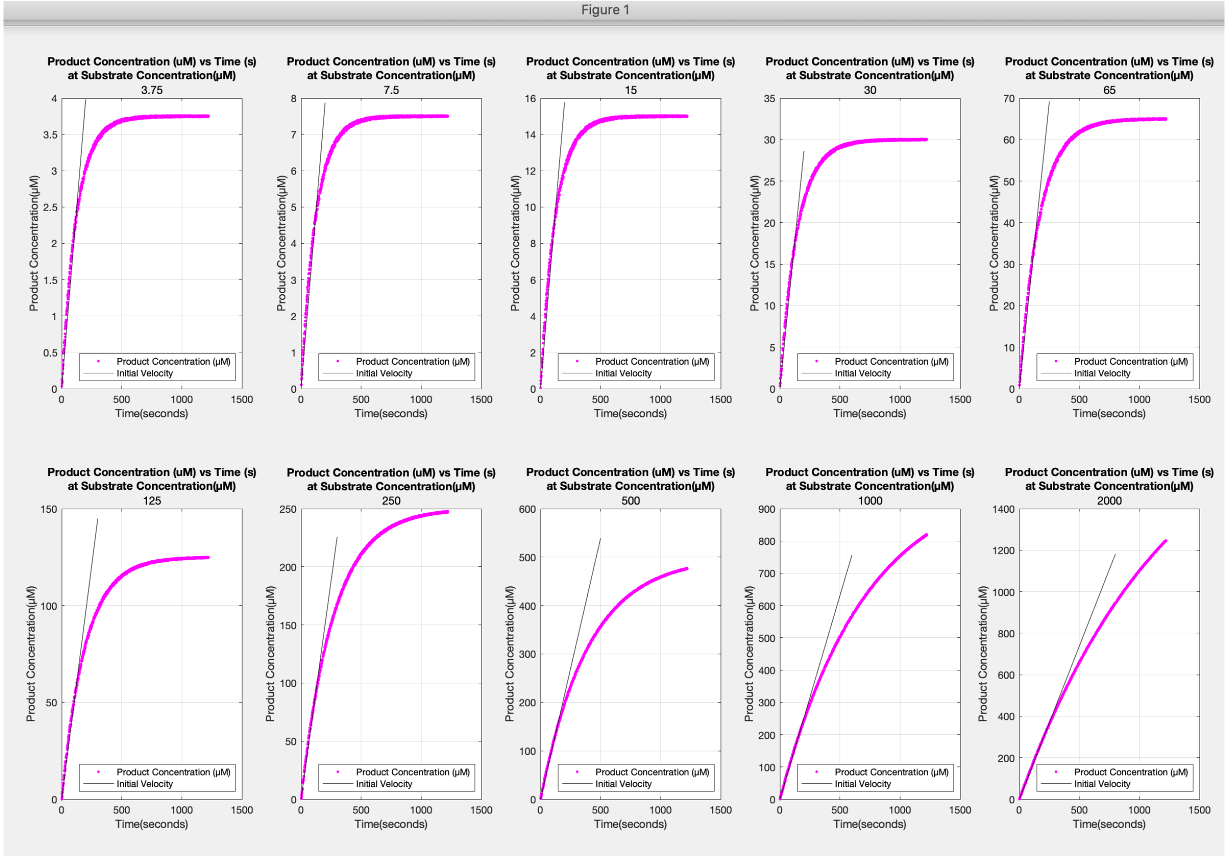


Figure 1
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 3.75
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 7.5
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 15
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 30
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 65
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 125
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 250
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 500
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 1000
Product Concentration (uM) vs Time (s) at Substrate Concentration(μM) 2000
Product Concentration(μM)
Time(seconds)
0
500
1000
1500
Product Concentration (μM)
Initial Velocity

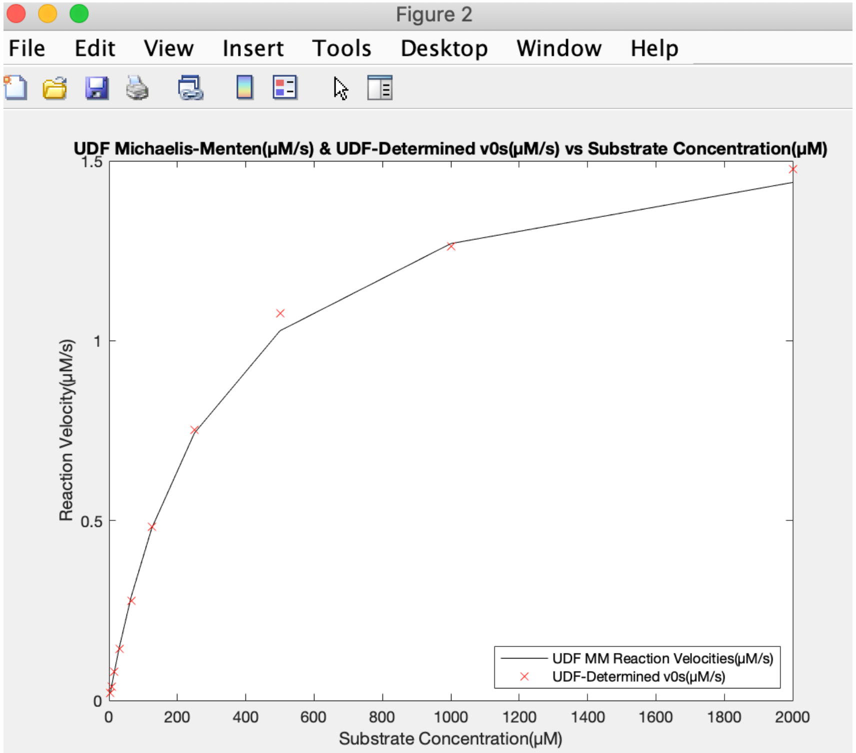


Figure 2
File
Edit
View
Insert
Tools
Desktop
Window
Help
UDF Michaelis-Menten(μM/s) & UDF-Determined v0s(μM/s) vs Substrate Concentration(μM)
Reaction Velocity(μM/s)
0
0.5
1
1.5
Substrate Concentration(μM)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
UDF MM Reaction Velocities(μM/s)
UDF-Determined v0s(μM/s)

Complete Table 5 using the parameters and results from your algorithm. Include the SSE results from Table 3 for easier comparison.

Table 5. Algorithm comparison using data results

Parameter (μM/s)	PGO-X50 Reference Values	M3_Algorithm
v_{0_1}	0.025	0.0199
v_{0_2}	0.049	0.0394
v_{0_3}	0.099	0.0788
v_{0_4}	0.176	0.1431
v_{0_5}	0.329	0.2767
v_{0_6}	0.563	0.4828
v_{0_7}	0.874	0.7521
v_{0_8}	1.192	1.0772
v_{0_9}	1.361	1.2618
$v_{0_{10}}$	1.603	1.4769
V_{max}	1.806	1.6625
K_m (μM)	269.74	309.0708
SSE	0.0049	0.0042

Part 3: Observations and Improvements

Based upon your observations of your plots and the SSE results for your model, suggest at least two ways you believe your algorithm could be improved. Briefly explain each suggestion below using evidence-based rationales. You do not need to code these changes; at this point, simply describe changes you think might be useful.

Be sure to:

- explain which parameter(s) your improvement will target,
- explain the improvement with a level of detail that can be understood by others (provide sketches or flowcharts as necessary to clarify your improvement),
- describe the performance metrics you will use to determine whether your proposed improvement really does improve your solution, and
- provide evidence-based rationales for each proposed improvement and the metrics selected. Your rationales should answer the questions:
  - What is your evidence that this improvement is necessary?
  - Why is this method for making the improvement a good idea - what is your evidence?
  - Why is this metric a good idea - what is your evidence?

Discuss improvements that resolve issues that you see in your current algorithm in the Table 6 below. Cite any external references in Table 8.

If, based on your PGO-X50 analysis, you feel that you do not need to make changes to your algorithm, indicate your rationale in Table 7. Cite any external references in Table 8. You should start thinking ahead to the data sets you will work with in M4 (data from ten concentration tests plus ten duplicate tests for 5 enzymes [i.e., the data from earlier milestones]).

Table 6. Algorithm improvements (add improvement blocks as needed)

Improvement 1. Parameter(s) Targeted: v0
Description If we operate under the assumption that the reference v0s are the true v0s or values that we should strive towards, then in that case, we should look to improve of v0s as when comparing them to the reference values, they are often a bit less than the reference. This likely means we are factoring in too many data points in v0s calculation to the point where it is decreasing the initial velocity that we calculate
Metrics to Determine Improvement To determine our improvement, we could simply check how our v0 values change as we adapt our algorithm and continually compare it against the reference numbers until they become within 0.02 micro-Molarity/second.
Rationale for Improvement and Metrics To improve our v0 calculation we should incorporate less values or have a smaller range of values that we look to use when calculating. As the more values we incorporate the smaller the magnitude of the corresponding initial velocity becomes as we start to include more portions of the graph that would not be classified as the initial/beginning. The reverse is true and would help us increase our v0s to match the values of the references. So by limiting the number of values we utilize in our v0s calculations, we would likely result in larger v0s. The rationale behind the metric is simply a value that we felt would be close enough to the reference values were it would be approximately similar enough where our further calculations would be indistinguishable from that of the reference values. As it stands the typical difference between our v0s and reference is in the neighborhood of 0.1 micro-Molarity/second or more. So our main goal is to decrease that as much as we can and we felt 0.02 was appropriate as it would decrease our current error by 80%, however, ideally, we want to match them exactly assuming the reference values are the numbers that we should be outputting.
Improvement 2. Parameter(s) Targeted: vMax + Km
Description Our vMax and Km values are also a bit off the mark when comparing it to the reference. However, these values are largely related to the v0s as we utilized the v0s in their calculations.

One of the main things that makes us believe that the error comes v0s is that the Km value is slightly bigger than the reference one but in the other hand the Vmax falls short. The way we calculated such values involves taking the inverse of the other (inverse variation), so it makes sense that when one is bigger the other value becomes smaller (Burns, 2021). Had both the parameters been larger than the reference, then the issue might have been due to a mathematical error but since this pattern exists, we know if is due to the v0s and how the error in v0s compounds after we convert them into the parameters.

Metrics to Determine Improvement

Similarly to the v0s, we ideally would like to exactly match our values to that of the reference, but if that is not feasible, we think having our vMax within 0.02 micro-Molarity/second and our Km within 1-2 micro-Molarity their respective reference values is appropriate metric to determine improvement.

Rationale for Improvement and Metrics

We don’t have a method to improve these two parameters specifically as if our v0s become more accurate, it is likely our vMax and Km that we compute will be more accurate. This is because our entire process of determining those two parameters is using a Lineweaver burke plot to linearize and then eventually find the non-linear regression line which we alter to get Vmax and Km. So, since this process largely relies upon finding a linear regression line from the v0s, any changes we make to improve the accuracy of the v0s will be reflected in the changes in the other parameters (i.e. km and vMax). In this situation, we are assuming our process of obtaining the parameters is mathematically sound/correct, which is what we truly believe but there could be an error present in there that might be responsible for the issues in our values, but we have yet to find this error if it even exists. As for the metrics, they serve as goals more than anything

Table 7. Justification for not needing improvements to algorithm

Apart from the minor improvements mentioned above we believe this nothing else really left to improve, we're very satisfied with the overall performance our algorithm is calculating. The SSE we calculated for our algorithm shows an extremely low error, because as explained before, the closer to 0 (the value for SSE) the better in terms of accuracy.

After some discussion we also thought about the idea that the values given by the company may have a small error. The company describes this data as a “well-known" data, but in this type of experiments, there is no certainty as any small change to the testing facility or data could cause a variation in the final values. Having said that, we acknowledge this is a very improbable situation, and we don’t have any evidence to back this claim, so for this reason, we are going to assume that there

isn’t much error in the data given and we are going to change our algorithm for it to meet such values.

Table 8. References used in evidence-based rationales

Burns, C. (2021, April 24). Direct and inverse variation. Retrieved April 26, 2021, from https://www.onemathematicalcat.org/algebra_book/online_problems/direct_inverse_variation.htm#:~:text=when%20one%20gets%20bigger%2C%20the,y%20varies%20inversely%20as%20x%20'.

How to Submit

1. Save this answer sheet as a PDF named **M3_*SSS*_*tt*.pdf** where ***SSS*** is your section number (e.g., 001 for section 001) and ***tt*** is your team number (e.g., 07 for team 7).
2. Select one person to submit the deliverables for the team. That person should
    a. Log into Gradescope and submit all deliverables to the **M3** assignment.
        i. M3_*SSS*_*tt*.pdf
        ii. M3_Algorithm_*SSS*_*tt*.m
        iii. M3_MM_PGOX50_*SSS*_*tt*.m
        iv. M3_main_*SSS*_*tt*.m
    b. Select all team members for the group assignment and submit.
    c. Double-check that all team members are assigned to the submission.
3. Each team member should confirm that they are part of the submission.
4. After submission, distribute the submitted files to all team members. *Ensure all members of the team have copies of the submitted files.*

Learning Objectives

Teamwork (TW)
Contribute to team products and discussions
TW02. Document all contributions to the team performance with evidence that these contributions are significant.

Process Awareness (PA)
Reflect on both personal and team's problem solving/design approach and process for the purpose of continuous improvement.
PA01. Identify strengths in the approach used.
PA02. Identify limitations in the approach used.
PA03. Identify potential behaviors to improve approach in future problem solving/design projects.

Evidence-Based Decision Making (EB)
Use evidence to develop and optimize solution. Evaluate solutions, test and optimize chosen solution based on evidence.
EB01. Test prototypes and analyze results to inform comparison of alternative solutions.
EB03. Clearly articulate reasons for answers with explicit reference to data to justify decisions or to evaluate alternative solutions.
EB05. Present findings from iterative testing or optimization efforts used to further improve aspect or performance of a solution.
EB06. Clearly articulate reasons for answers when making decisions or evaluating alternative solutions.

Solution Quality (SQ)

Design final solution to be of high technical quality. Design final solution to meet client and user needs.

SQ01. Use accurate, scientific, mathematical, and/or technical concepts, units, and/or data in solutions.

Data Visualization and Analysis (DV)

Visually represent data and derive meaningful information from data. Demonstrate ability to accurately describe data sets through foundational descriptive statistics and then perform simple inferential statistics through understanding probability concepts and linear regression.

DV03. Justify graphical representation based on data characteristics.

Information Literacy (IL)

Seek, find, use and document appropriate and trustworthy information sources.

IL03. Support all claims made with that is either generated or found

Engineering Professional Skills

PC05. Fully address all parts of assignment by following instructions and completing all work.

EPS01. Use professional written and oral communication.

EPS02. Format plots for technical presentation.

Programming

MAT01. Develop code that follows good programming standards

MAT08. Debug scripts and functions to ensure programs execute properly, perform all required tasks, and produce expected results.