# Project Milestone 2 – Algorithm Development

## 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. Good programming standards apply to all m-files.
5. Submit deliverables to Gradescope and to Blackboard. 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 |
|---|---|
| M2 Answer Sheet | M2_AnswerSheet_*SSS_TT*.pdf |
| M2 Algorithm(s) | M2_Algorithm_*SSS_TT*.m<br>* If your team has multiple UDFs, replace "Algorithm" with appropriate descriptor. Include team ID. |

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):** | 002_02 |

| Team Member Name | Purdue Career Account Login |
|---|---|
| Jack Swingle | jswingle |
| Matthew Imm | mimm |
| Connor Damato | Damato0 |
| Kush Gogia | kgogia |

## 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 |
|---|---|---|

| | | |
|---|---|---|
| Connor Damato | Created the user defined function to calculate the v0. And helped with main function | 25 |
| Jack Swingle | Created the main function to call user defined functions and initialize data | 25 |
| Kush Gogia | Helped create the user defined function to calculate v0 and Km. Wrote the algorithm reflection. | 25 |
| Matthew Imm | Helped create the user defined function to calculate v0 and Km. Wrote the M1 feedback, and algorithm plan. | 25 |

## Part 0: M1 Feedback Review

Reflect on your M1 feedback for the purpose of improvement. Your reflection should provide a clear, useful summary of your M1 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 M1, identify at least one strength and one limitation of your team's approach or process you created in M1. Consider how the feedback from M1 could lead to improvements in your work.**<br>The feedback we received for our M1 gives us a lot to work with. We did not lose any points on the assignment, which shows our approaches, and our format for answering the questions is correct. One thing that was given was that our approaches may be easier to justify and explain with images because our explanations could be a little wordy at times. In the future, when we are justifying our approaches, we will likely include images to help aid our explanations. Overall, our M1 was well done and our main strength is the approaches for finding the different variables. |
| **Part B: Explain how you will incorporate the M1 feedback to improve your parameter identification** (do not just reword your response from Part A, include concrete actions you will take to improve).<br>We will use our M1 feedback to improve parameter identification by plotting our results to ensure that it is accurate. We will use graphs to check our work to see how accurate we are as well as finding the r^2 values. |

## Part 1: Algorithm Development

In Milestone 1, you developed approaches for identifying $v_{0_i}$ and approaches for identifying $V_{max}$ and $K_m$. Now, you will take your best ideas from M1 and use them to create an algorithm (which can be one or more user-defined functions) coded in MATLAB to perform parameter identification on enzyme reaction data.

You must develop a plan for your algorithm before you start coding. Outline your algorithm using pseudocode (i.e., plain English text, not MATLAB code). Remember, it is valuable to develop and organize your programming ideas and solutions *before* you code. A well-developed plan reduces coding frustrations.

As you develop your function parameter identification plans and algorithm, keep in mind that it will be called in future milestones by a main function. This main function will load your data, assign the correct variables, and call your user-defined function parameter identification algorithm to determine the parameters for the data. Use this description of the main function to help you determine appropriate inputs and outputs for your algorithm.

Complete your plans in Table 2 below.

### Table 2. Algorithm plans

| **Plan for Algorithm** |
|---|
| In our algorithm we will use 3 different Matlab programs. We will use 2 user defined functions, and one main function. Our first user defined function will calculate v0 by linearizing the data and finding the slope for an increasing domain until the r^2 value is at the max. The user defined function will automatically return the v0 that is calculated using the highest r^2 value. Our second user defined function will use the v0 calculations and the lineweaver burk linearization method in order to calculate the vmax and the km value. The main function will first initialize the data while removing the non numeric sections of the data. Then it will call the first user defined function in order to calculate the v0 for each set of data. After it has all v0s calculated, it will average the v0 for the original and replicate data that is the same enzyme. Then with the newly averaged v0, it will call the second user defined function in order to calculate the vmax and the km values. |

After you complete your plan, translate your plan into **two** (2) user-defined functions. Your subfunction should use your algorithm to perform parameter identification $v_{0_i}$, $V_{max}$ and $K_m$) on enzyme reaction data. The subfunction should compute the parameters for 1 enzyme at a time. Your main function should load the data, segmenting the data, and call the subfunction once for each enzyme in your data set. Use the filename **M2_Algorithm_SSS_tt.m** for your subfunction and the filename **M2_Main_SSS_tt.m** for your main function. Make sure that your algorithm follows ENGR 132 Programming Standards and is clearly commented throughout.

## Part 2: Algorithm Reflection

In Table 3, discuss your choice of algorithm, your process for debugging your algorithm, and the strengths and limitations of your algorithm at this point. See the directions in each part of the table.

### Table 3. Algorithm reflections

| **Your choice of your algorithm in M2** |
|---|
| Describe your process for choosing how you would develop your algorithm? How did you use your data in this process? |

We chose to develop our algorithm in 3 separate parts. We wanted to have a main function that calls user defined functions in order to do calculations. We decided to use 2 separate user defined functions because there are two main calculations that need to be done. The v0 calculation is the first calculation, and then after the v0 is calculated, the km and vmax can be calculated from there. We used our data to help us decide how to set up our v0 calculation.

### Debugging your algorithm in M2

Describe your process for making sure your algorithm is meeting the needs of the client and running smoothly. What did you do to debug your algorithm? How did you use your data in this process?

Our process for ensuring our algorithm meets the needs of the clients was fairly basic. We used an r^2 calculation to ensure the highest amount of accuracy when calculating v0. When we calculated Km and Vmax using the lineweaver burk linearization, we would graph our findings in order to check to make sure they look accurate. Using these methods to check our program helped us debug and optimize our program to run smoothly. We used our data in the r^2 calculations and in our graphs.

### Strengths and limitations to your algorithm in M2

Identify at least one strength and one limitation of your team's algorithm you created in M2.

One strength of our algorithm is the accuracy of the v0 calculation. We used a loop to make sure the v0 was calculated with the optimal domain and had the highest r^2 value, which helped us to eliminate noise. One limitation of our algorithm is the km and vmax calculations. Because we used the lineweaver burk linearization method, there is an inherent error that is not able to be avoided. Because we calculated v0 of a noisy plot, there is some error and that error is magnified in the lineweaver burk linearization.

## How to Submit

1. Save this answer sheet as a PDF named **M2_AnswerSheet_SSS_*TT*.pdf.**
2. Select one person to submit the assignment for the team. That person should
   a. Log into Gradescope and submit **M2_AnswerSheet_SSS_*TT*.pdf**, **M2_Algorithm_*SSS*_*TT*.m**, and **M2_Main_SSS_tt.m** to the **M2** assignment.
   b. Select all team members for the group assignment.
   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 and that all parts of the answer sheet were properly tagged.
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 problem solving/design approach.
PA02. Identify limitations in the approach used.
PA03. Identify potential behaviors to improve approach in future problem solving/design projects.

**Idea Fluency (IF)**

Generate ideas fluently. Take risks when necessary.
IF03. Generate testable prototypes (including process steps) for a set of potential solutions.

**Evidence-Based Decision Making (EB)**

Use evidence to develop and optimize solution. Evaluate solutions, test and optimize chosen solution based on evidence.
EB03. Clearly articulate reasons for answers with explicit reference to data to justify decisions or to evaluate 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.

**Engineering Professional Skills**

PC05. Fully address all parts of assignment by following instructions and completing all work.
EPS01. Use professional written and oral communication.

**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.