

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	Project_M2_AnswerSheet_SSS_TT.docx
M2 Algorithm	M2_Algorithm_SSS_TT.m
Gradescope Submission	M2_SSS_TT.pdf

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):	005-19
--------------------------------------	--------

Team Member Name	Purdue Career Account Login
Annabelle Johnson	john2445@purdue.edu
Nic Ballesteros	nballes@purdue.edu
Alan Camacho	Camachop@purdue.edu

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
Annabelle Johnson	Worked on Tables 1,2,3 and plotting the data	33%
Alan Camacho	Worked on Table 2 and commented code	33%

Nic Ballesteros	Worked on Table 2 and translated the plan into code	34%

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.

Looking at the feedback we have received on M1, we can see that one of our team's approach is the ability to produce a near correct output through very little coding. This will help us avoid silly mistakes and errors when producing our code. One limitation to our approach, however, is determining which times to calculate our v_0 values. Due to noise and other issues, it is hard to determine which small step in time will create the best response for v_0 . We can use this feedback to determine which time interval would be the best to take our initial slope and how we can avoid taking the slope of the noisy data instead of a smoother curve.

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).

We will take our feedback mentioned above to improve our parameter identifications by making sure we get rid of noise so we can get as close to our initial slope as possible. To do this we will be linearizing our initial data and then creating a model line to use as our time vs substrate concentration curve. After we do this, we can find the slope of our model line from initial time 0 and a small step such as time = 0.01s. This will provide us with a secant line that is very close to being the original tangent line and will provide a v_0 value that is very close to the actual value.

Algorithm Development

In Milestone 1, you developed approaches for identifying v_{0i} and approaches for identifying V_{max} and K_m . Now, you will take your best ideas from M1 and use them to create a single algorithm that is 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
<ol style="list-style-type: none"> 1. Remove all the NaN values from our columns 2. Store the data for the concentration tests and their duplicates 3. Smooth the data by creating a model 4. Find the v_0 values of each initial concentration test and their duplicates 5. Store these values for easy access later 6. Using the Hanes-Woolf linearization method, linearize the v_0 values 7. Determine the coefficients of the linearized model 8. Use the coefficients to solve for V_{max} and K_m 9. Create a model for the v_0 graph 10. Graph the v_0 values and their model

After you complete your plan, translate your plan into a user-defined function. Use the filename **M2_Algorithm_SSS_tt.m**. Make sure that your algorithm follows ENGR 132 Programming Standards and is clearly commented throughout.

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
<p>Describe your process for choosing how you would develop your algorithm? How did you use your data in this process?</p> <p>We wanted to create our algorithm so that we could easily get rid of noise and still be able to have a curve that is easily related to the data given. We also wanted to consider that not all tests ran for the same amount of time. When looking at our data, we saw that we would need to get rid of NaN values to insure that our program would function properly for each test without considering the nonexistent values.</p>
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?

We started by graphing the results to make sure our models still matched up with the data. We then produced the code that would calculate the parameters requested by the client. We made sure that everything in the code was labeled and easy to understand so that our client would be able to understand where our values came from. When we did have errors come up with our code, the team would sit down and reread the code individually to give a fresh set of eyes to different code and see if mistakes pop out at them. We would then run the code and if an error popped up in the command window, we would work to fix that. We used the data in this process by cross referencing our models with the data to see if the model created the same overarching shape that the data created.

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.

Our algorithm has the strength in that it provides both the parameters and graphs justifying the parameters. Our algorithm also uses linearization multiple times to make sure the data is as smooth as we can make it and provide the best results for our parameters. Even with these strengths, our algorithm still has some limitations. Our linearization of the v_0 values tend to bias towards the smaller initial concentrations making the model not as accurate towards higher concentrations.

How to Submit

1. Rename this answer sheet to be **Project_M2_AnswerSheet_SSS_TT.docx** where **SSS** is your section number (e.g., 001 for section 001) and **TT** is your team number (e.g., 07 for team 7). Save the answer sheet as a PDF.
2. Publish the function you created in this milestone.
3. Merge the answer sheet PDF and the published code PDF into one PDF file named **M2_SSS_TT.pdf**.
4. Select one person to submit the PDF for the team. That person should
 - a. Log into Gradescope and submit **M2_SSS_TT.pdf** to the **M2** assignment.
 - b. Indicate which pages correspond to each part of the milestone.

Failure to tag the items appropriately will result in your work receiving partial credit for all wrongly tagged items.
 - c. Select all team members for the group assignment.
 - d. Double-check that all team members are assigned to the submission.
5. Each team member should confirm that they are part of the submission and that all parts of the answer sheet were properly tagged.
6. Select one person to submit all m-files, data files, and answer sheet to the **M2 dropbox** on Blackboard.

Failure to submit your files to Blackboard may result in every team member receiving a zero on this assignment.

7. 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.

Table of Contents

.....	1
.....	2
INITIALIZATION	2
.....	2
CALCULATIONS	2
.....	5
FORMATTED TEXT/FIGURE DISPLAYS	5
.....	5
COMMAND WINDOW OUTPUT	5
.....	5
ACADEMIC INTEGRITY STATEMENT	5

```
function [Km,Vmax] = project_function(time, enzymeData);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
%   This program estimates the Michaelis-Menten parameters, Km and
%   Vmax, for
%   a given enzyme's data.
%
% Function Call
%   [Km, Vmax] = project_function(time, substate_data);
%
% Input Arguments
%   time: the time variable for each given data set.
%   enzymeData: First row is the initial concentrations of the
%               substrates for
%               each given test. The rest of the rows are the data points for each
%               test; each
%               test in a separate column.
%
% Output Arguments
%   Km: Outputs the estimated Km value for the enzyme
%   Vmax: Outputs the estimated Vmax value for the enzyme
%
% Assignment Information
%   Assignment:      M02, Problem 1
%   Team member:    Nic Ballesteros, nballes@purdue.edu
%   Team member:    Annabelle Johnson, john245@purdue.edu
%   Team member:    Alan Camacho, @purdue.edu
%   Team ID:        005-19
%   Academic Integrity:
%   [x] We worked with one or more peers but our collaboration
%       maintained academic integrity.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION

```
% Each test is stored in the following manner
% to get the data for a test the following command is used :
%     test(test#).data;
% to get the duplicate data for a test the following command is used:
%     test(test#).dupData;
% to get the concentration of the test use:
%     test(test#).concentration;

%Gets rid of times where reactions are no longer occurring
for i = 1:10
    test(i).data = rmmissing(enzymeData(2:end, i)); %get all not NaN
    values in each col for each test
    test(i).dataSize = size(test(i).data); %Determines the number
    of seconds that pass before the reaction stops for each initial
    substrate concentration
    test(i).time = time(1:test(i).dataSize(1)); %Creates a matrix of
    times where the reaction was occurring for each initial substrate
    concentration
    %store the duplicate data
    test(i).dupData = rmmissing(enzymeData(2:end, i + 10)); %get all not
    NaN values in each col for each duplicate test
    test(i).dupDataSize = size(test(i).dupData); %Determines the number
    of seconds that pass before the duplicate reaction stops for each
    initial substrate concentration
    test(i).dupTime = time(1:test(i).dupDataSize(1)); %Creates a matrix
    of times where the duplicate reaction was occurring for each initial
    substrate concentration
    %store the concentration
    test(i).concentration = enzymeData(1, i); %Creates a matrix of
    substrate concentrations
end;

mmData = zeros(20, 2); %Michaelis-Menten data that will eventually be
    plotted

Not enough input arguments.

Error in project_function (line 45)
    test(i).data = rmmissing(enzymeData(2:end, i)); %get all not NaN
    values in each col for each test
```

CALCULATIONS

```
% -----
% find the v0 data
```

```

% -----

for i = 1:10
    test(i).v0 = [0 0];
    x = test(i).time;
    y = test(i).data;

    x(1) = []; %had a divide by zero error
    y(1) = []; %to line up both vectors

    x = x(1:500); %linearize only the first 500 values
    y = y(1:500); %linearize only the first 500 values

    y = x ./ y; %linearize the product data

    %find line best fit
    xline = mean(x);
    yline = mean(y);
    xyline = mean(x .* y);

    a = (xline * yline - xyline) / (xline ^ 2 - mean(x .^ 2));
    b = yline - a * xline;
    %done with best fit

    a = 1 / a;
    b = b * a;

    %make dataset off of modeled line
    xDataPoints = 1:500;
    yDataPoints = (a * xDataPoints) ./ (b + xDataPoints);

    %use line to find initial velocity
    test(i).v0(1) = (yDataPoints(2) - yDataPoints(1)) / (xDataPoints(2)
- xDataPoints(1));

    %add it to the Michaelis-Menten dataset
    mmData(2 * i - 1, 1) = test(i).concentration;
    mmData(2 * i - 1, 2) = test(i).v0(1);

    %do the same thing as above but for the duplicate data
    x = test(i).dupTime;
    y = test(i).dupData;

    x(1) = []; %had a divide by zero error
    y(1) = []; %to line up both vectors

    x = x(1:500); %linearize only the first 500 values
    y = y(1:500); %linearize only the first 500 values

    y = x ./ y; %linearize the product data

    %find the line best fit
    xline = mean(x);

```

```

yline = mean(y);
xyline = mean(x .* y);

a = (xline * yline - xyline) / (xline ^ 2 - mean(x .^ 2));
b = yline - a * xline;

a = 1 / a;
b = b * a;

%use the model to make a dataset
xDataPoints = 1:500;
yDataPoints = (a * xDataPoints) ./ (b + xDataPoints);

%use data set to find the initial velocity
test(i).v0(2) = (yDataPoints(2) - yDataPoints(1)) / (xDataPoints(2)
- xDataPoints(1));

%add the initial velocity to the Michaelis-Menten dataset
mmData(2 * i, 1) = test(i).concentration;
mmData(2 * i, 2) = test(i).v0(2);
end;

%-----
%implementing Hanes-Woolf Linearization
%-----

data(:, 1) = mmData(:,1);
data(:, 2) = mmData(:, 1) ./ mmData(:, 2);

%data = rmoutliers(data);

Y = mmData(:, 1) ./ mmData(:, 2);
X = mmData(:, 1);

Xline = mean(X);
Yline = mean(Y);
XYline = mean(X .* Y);

a = (Xline * Yline - XYline) / (Xline ^ 2 - mean(X .^ 2));
b = Yline - a * Xline;

fx = X * a + b; %Hanes-Woolf Line

Vmax = 1 / a; %calculate Vmax from Hanes-Woolf
Km = b * Vmax; %calculate Km from Hanes-Woolf

% make a dataset that follows the model
numberOfDataPoints = 100;
seperation = (2000 - 3.75) / numberOfDataPoints;
xmodel = 3.75:seperation:2000;
MichaelisModel = Vmax * xmodel ./ (Km + xmodel);

```

FORMATTED TEXT/FIGURE DISPLAYS

```
%Plots the Calculated Reaction Velocities against the Model Reaction
%Velocities
figure;
subplot(2,1,1);
plot(mmData(:,1), mmData(:, 2), 'ko'); %Calculated Reaction
Velocities
title('Reaction Velocity as Initial [S] changes');
xlabel('Initial Substrate Concentration [S] (uM)');
ylabel('Reaction Velocity (uM/s)');
hold on;
plot(xmodel, MichaelisModel, 'r--'); %Michealis Model curve
legend('Calculated Reaction Velocities', 'Michaelis
Model', 'location', 'best');

subplot(2,1,2);
plot(X,Y, 'ro');
hold on;
plot(X, fx, 'b-');
xlabel('Initial Substrate Concentration (uM)');
ylabel('Velocity / [S]');
title('Hanes-Woolf Linearization');
legend('Linarized Velocity Data', 'Best Fit
Line', 'location', 'best');
```

COMMAND WINDOW OUTPUT

```
fprintf("Vmax: %.3f\n", Vmax);
fprintf("Km: %.3f\n", Km);
```

ACADEMIC INTEGRITY STATEMENT

We have not used source code obtained from any other unauthorized source, either modified or unmodified. Neither have we provided access to my code to another. The program we are submitting is our own original work.

Published with MATLAB® R2019b