General Instructions

Due Date:
Saturday, April 15 at 11:00pm (submit via blackboard)

Assignment Summary Instructions:
This assignment has one problem with several tasks. You will use MATLAB as a tool to solve the problems for the given test cases provided and that it is flexible for any additional test case that might be used to evaluate your code.

- Problem #1: Area between curves

Submission Instructions:
You will submit a **.zip file** that will contain the following files:

1. **.m file – Matlab file (separate sections using %% for each problem)**. Your final script should be in the zipped folder and named as **MA#_USERNAME.m (for example, MA8 _dwburles.m).**
2. .pdf file - scan of your algorithm sheet (use the template) for Problem #1

Submit your **.zip file** to blackboard using the Mastery Assignment 1 link. Your final submission should be a zipped folder named as **MA#_USERNAME.zip (for example, MA8_dwburles.zip).** The file must be completely submit before 11pm on April 15 for credit. **No late work is accepted and will result in a zero on the assignment**. It is your responsibility to make sure the file is completely submitted prior to the deadline. You are provided 2 upload opportunities in case your first upload is incomplete or a mistake.

Academic Honesty Reminder:

The work you submit for this assignment should be your work alone. You are encouraged to support one another through collaboration in brainstorming approaches to the problem and troubleshooting. **However, all support should be only verbal in nature**. Sharing files and showing other students your file is considered physical and visual help and is considered an academic honesty violation.

Some examples of academic misconduct in ENGI 1331 include but are not limited to the following actions:

1. Picking up and using or discarding another student's written or computer output;
2. Using the computer account of another student;
3. Representing as one's own the work of another on assignments, quizzes, and projects;
4. Giving another student a copy of one's work on an assignment before the due date.
5. Copying work from online resources (Chegg, google forums, etc.)
6. Posting work to online resources where other students can view your work

This assignment will be checked for similarity using a MATLAB code. The similarity code will check each submission for likeness between other student submissions, past student submissions, the solution manual, and online resources and postings. If your submission is flagged for a high level of similarity, the ENGI 1331 faculty will review the files, and then the guilty parties will be turned in for an academic honesty violation if deemed appropriate.

NOTE: Since this is an automated system for all sections, if any of your work is not your own, you will be caught. Changing variable names, adding comments, or spacing will not trick the similarity algorithm and will result in a violation.

Problem #1: Area between the curve

Background: You need to analyze a series of test runs by calculating the area defined by the intersection points of multiple curves. The data file, ***area_data.csv*** file contains the coefficients for 3 separate functions listed below,

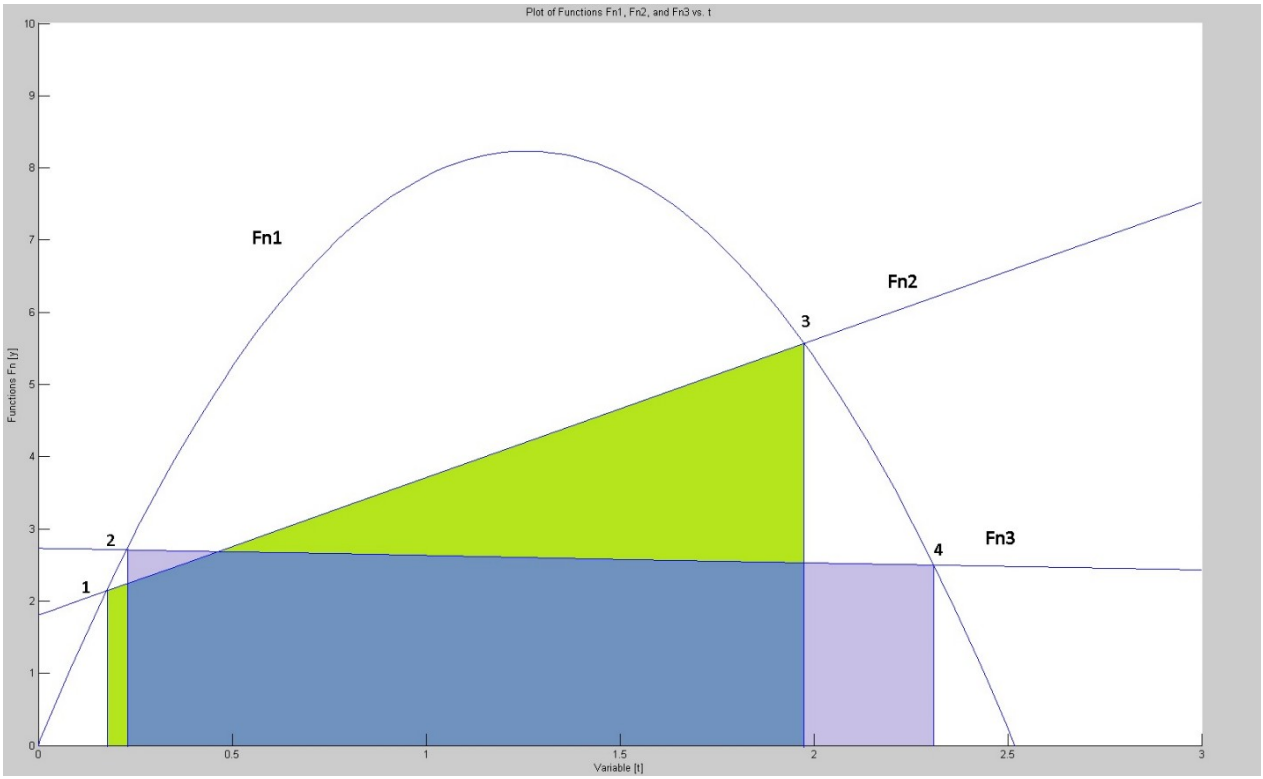
$$fn1(t) = At^2 + Bt + C, \quad (1)$$
$$fn2(t) = Mt + N, \quad (2)$$
$$fn3(t) = Zt + W, \quad (3)$$

where [A,B,C,M,N,Z,W] correspond to columns [1,2,3,4,5,6,7] of the data file.

Using user defined functions, define the 3 functions ***fn1***, ***fn2***, and ***fn3*** in MATLAB and allow for the coefficients above to be input as variables. Note that in every single test run, functions ***fn2*** and ***fn3*** intersect ***fn1***.

The goal of your program is to analyze the data to determine for each case which function, ***fn2*** or ***fn3***, produces the most area between its intersection points with ***fn1***. You will compare this value with a Riemann sum and you will also display which function has a higher value along the left and right side of ***fn1***. (For a brief review on Riemann sums, see the document "Riemann Sum.pdf" on the course website attached to MA8.)

See image below that defines the regions to calculate.



You will need to determine the points 1-4 (intersection points) then calculate the area defined only between the intersection of each function ***fn2*** and ***fn3*** with function ***fn1*** upon each iteration.

For the Riemann sums, you should calculate the Left Hand, Right Hand and Midpoint sums using a user input number of rectangles. To define your rectangle limits, divide the width of the intersection points for each function by the number of rectangles. It may be useful to define another function that computes the Riemann sums that may be called by other functions.

To determine which function is higher along the left and right edge of ***fn1*** you will need to evaluate each function, ***fn2*** and ***fn3***, at its intersection points and compare them to find the max. You will need to display a string value in the table.

Write a function named ***Compute_area*** that takes in one full set (single row) of the coefficients from the data file along with the number of rectangles to compute for the Riemann sums. Your function should output to the command window a single row of computed area for each test run, the calculated Riemann sum values, and will output which function has the highest value along each edge with the associated value. You should then output the following table below for several sample rows calculated outside of your function.

fn2	LH	RH	MP	Area fn3	LH	RK	MP	Left edge	Right edge
5.65	5.55	5.76	5.65	3.48	3.54	3.43	3.48	Fn3	Fn2
1.61	1.59	1.63	1.61	3.91	3.96	3.87	3.91	Fn3	Fn3
3.72	3.65	3.80	3.72	5.20	5.23	5.18	5.20	Fn3	Fn2
0.85	0.84	0.87	0.85	5.59	5.60	5.58	5.59	Fn3	Fn3
3.91	3.80	4.02	3.91	2.59	2.68	2.51	2.59	Fn3	Fn2

Your program should ask the user to input the number of rectangles to compute when estimating the Riemann sums make the appropriate calculations for each set of coefficient values from the supplied data file and will call ***Compute_area*** to output a single row of calculated area, Riemann sums, and display the highest left and right edge. Sample output for the data provided below. NOTE: Your code should be flexible for data file with a different number of test runs with intersections in different ranges.