**MATLAB Review:**

Monday lab lecture covered some basics of MATLAB review. Plotting and for loops will be used in this lab. Please complete the exercises and submit a pdf with your MATLAB code, plots, calculations, and outputs on GradeScope within a week of your lab date.

**Opening a text file in MATLAB:**

*There are many ways to do this in MATLAB, feel free to use the method you are most comfortable with.*

|  |  |
| --- | --- |
| [...] = textread(‘filename.txt’, ...‘formatSpec’) | Outputs an array with specified format. formatSpec options include: %f for floating point values, %d for signed integers, %s for white space, etc. |
| fileID=fopen(‘filename.txt’, ‘r’)  A=fscanf(fileID,’formatSpec’,[m n]) | Reads file into an array, with dimensions [m n], populates in column order. |
| M=dlmread(‘filename.txt’, ‘delimiter’, R, C) | Reads file into a matrix, with commas as a default delimiter. R and C are what row and column to begin reading from. |

|  |
| --- |
| Exercise 0: (10 points)   1. Open ‘224\_MATLAB\_practice.txt’ in MATLAB. 2. Store the first column of data as x, and the second column as y. 3. Plot an x-y graph. |

**Integration commands in MATLAB:**

|  |  |
| --- | --- |
| syms | Creates fresh symbolic variables for interactive symbolic workflows. Allows for the creation of symbolic expressions. |
| diff(f) | Differentiates f with respect to symbolic variables in workflow. |
| int(f) | Finds a symbolic expression, F so that diff(F) = f.  Returns indefinite integral/ antiderivative of f with respect to symbolic variables. |
| int(f, a, b) | Finds , where x is stored as a symbolic variable. |

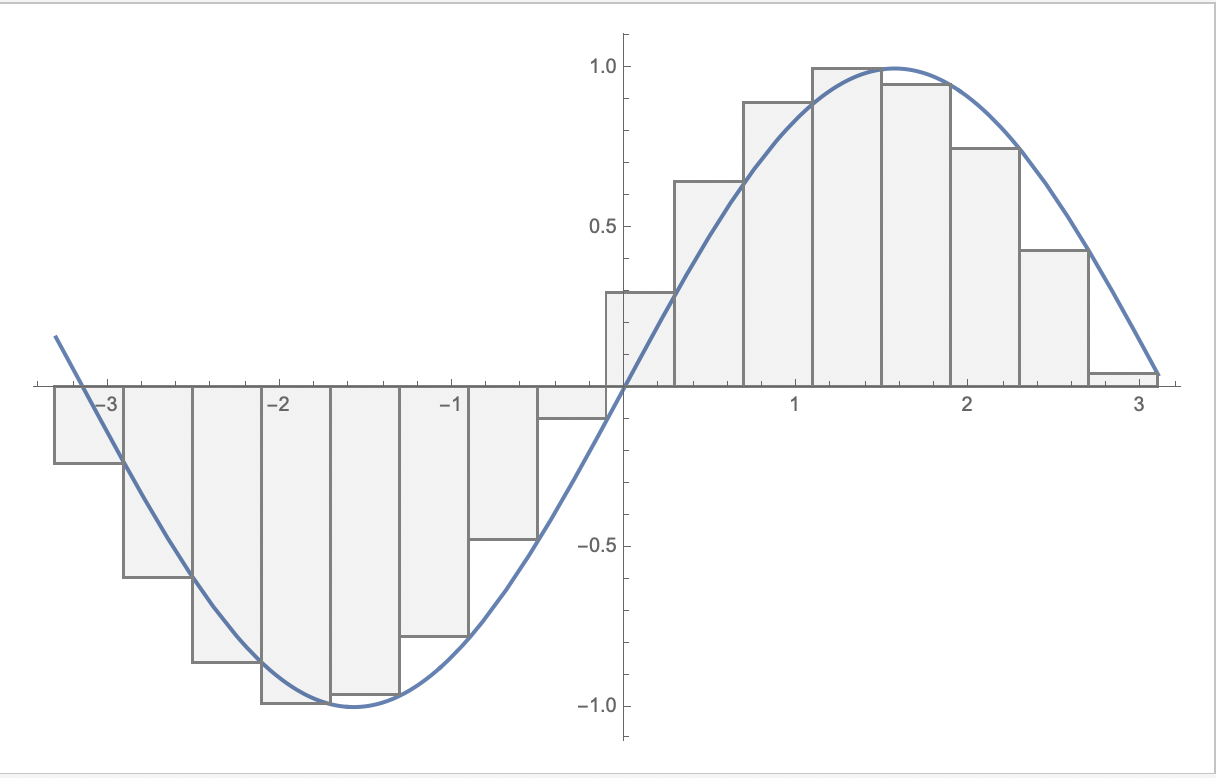
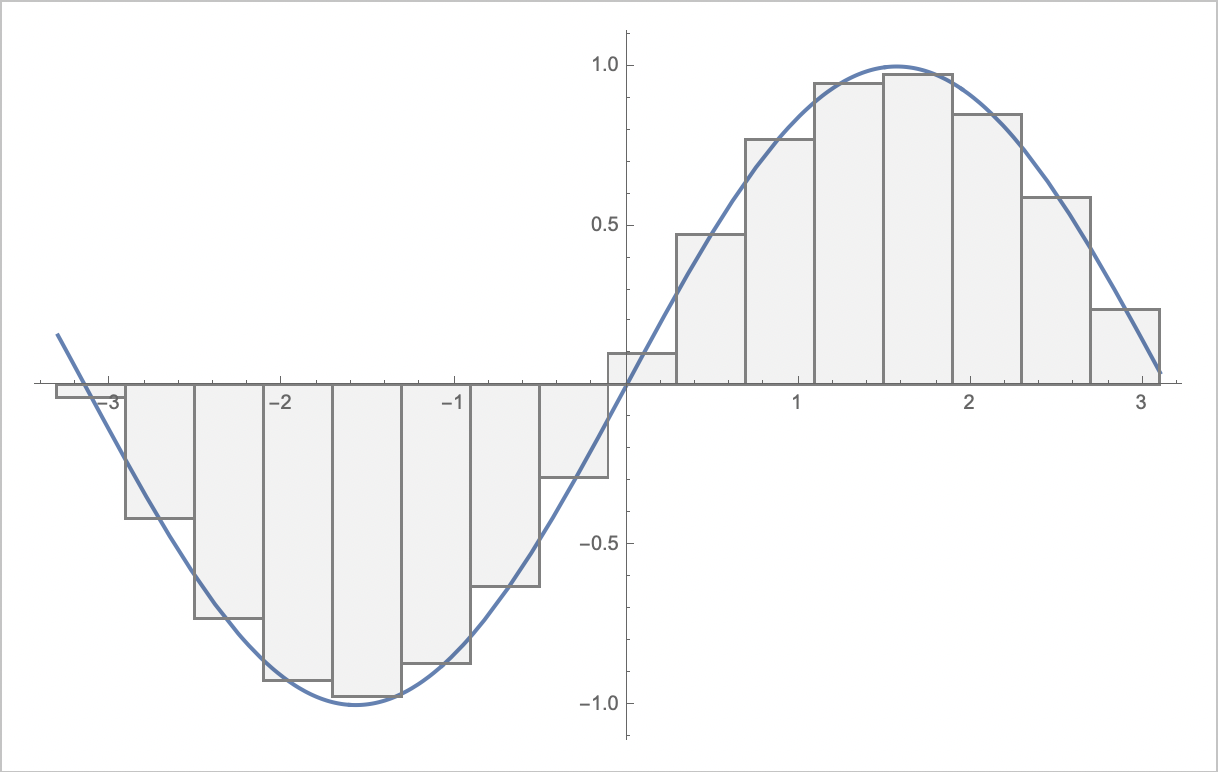
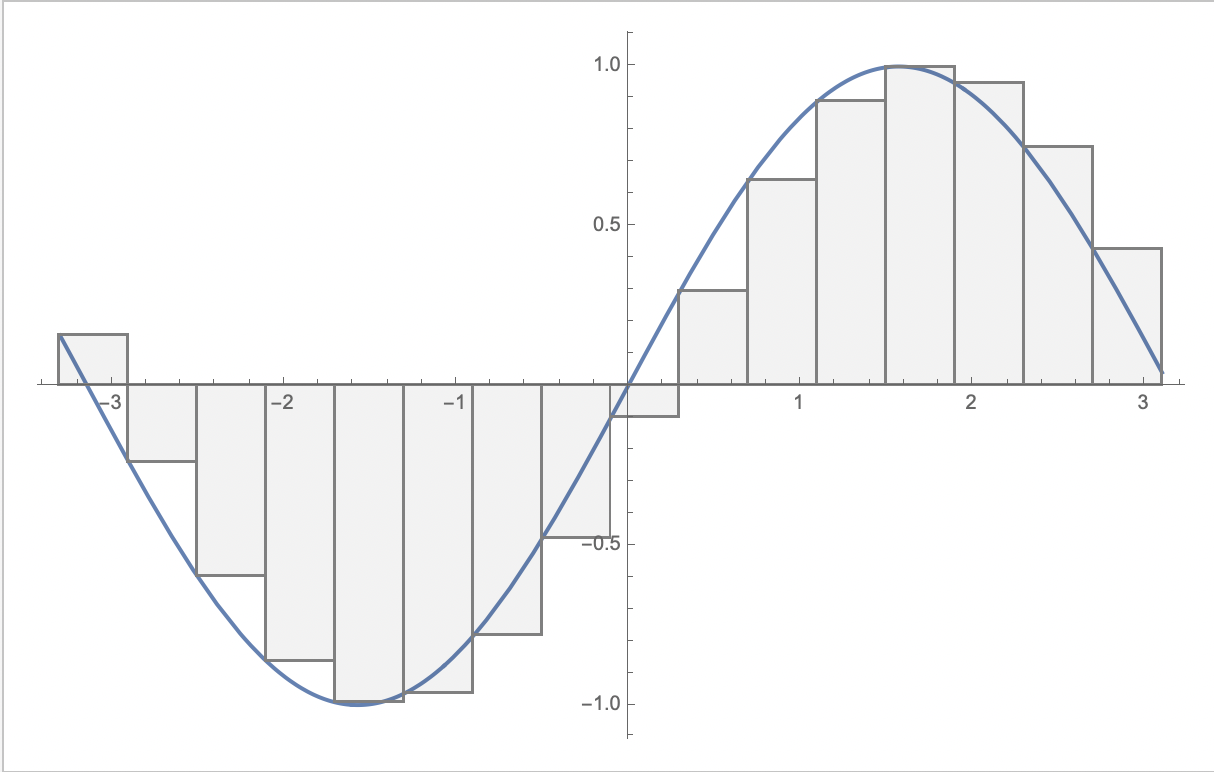
**Integral Practice:**

|  |
| --- |
| Exercise 1: (7.5 points)   1. Solve this integral by hand. 2. Solve this integral using MATLAB symbolic integration. 3. Compare your results.   Exercise 2: (7.5 points)   1. Solve this integral by hand. 2. Solve this integral using MATLAB symbolic integration. 3. Compare your results. |

**Review of Riemann sums:**

For a continuous function *f* (x) on [*a,b*], the definite integral can be computed as follows:

This approximation partitions the area under the curve into rectangles of width and height . It then calculates the area of *n* number of rectangles and adds the areas together. There are left, middle, and right Riemann sums, all of which define in different ways. These are shown in the image below.



RIGHT

MIDDLE

LEFT

For middle Riemann sums:

For partitions of equal size, dividing [*a,b*] into subintervals of equal length (i.e. ):

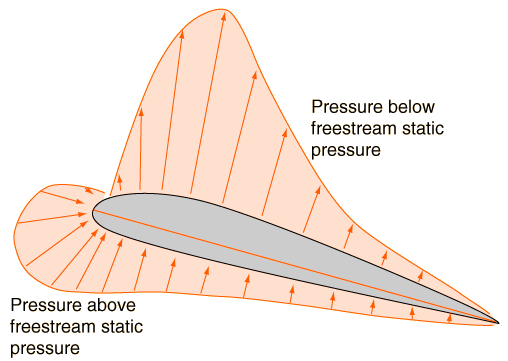
|  |
| --- |
| Exercise 3: (15 points)   1. Where , how would you define for a left Riemann sum? For a right Reimann sum? 2. In what cases will the left Riemann approximation overestimate the integral? When will it underestimate it? 3. In what cases will the right Riemann approximation overestimate the integral? When will it underestimate it?   Exercise 4: (55 points)   1. Solve this integral by hand. 2. Solve this integral using MATLAB numerical integration. 3. Plot the function in MATLAB    1. Does a left Riemann approximation overestimate or underestimate the area under the curve?    2. Does a right Riemann approximation overestimate or underestimate the area under the curve? 4. Solve this integral (in MATLAB) using middle Riemann approximations where *n = 8*. (Hint: this is best done with a for loop) 5. Solve this approximation for all *n* values from 5-20. (Hint: this is best done nesting your for loop from above inside another for loop) 6. What is the percent error for approximations where *n* = 5, *n* = 8, and *n* = 20? Use MATLABs numerical integration as your exact solution to the integral. 7. As *n* increases, does your approximation get closer or further from the exact value? Will this be the case for all functions? |

**Lab 3 Considerations:**

This type of approximation for integrals will be revisited in Lab 3. This next section acts as a short introduction to the topics of Lab 3, which will be further detailed in the upcoming weeks.

Consider body forces on an airfoil, where *P* is the pressure:

An example pressure distribution is visualized below:



From https://conself.com/blog/calculate-lift-drag-with-paraview/

|  |
| --- |
| Exercise 5: (5 points)   1. How would you write this in summation format? (Hint: this uses index notation) 2. Given that , how would you rewrite this summation into vector components? |