Carlise Moreland and Joshua Pollock

CS 122L-1

Lab 10: Riemann Sums

23 April 2018

**Task Description**

In this lab, we were tasked to write code to calculate the Riemann sums of a function. Riemann sums are used to estimate the area beneath a function. There are four ways to calculate a Riemann sum: left, right, middle and trapezoid. Left is an underestimate when the graph is increasing. Right is and overestimate when the function is positive and increasing. Middle and trapezoid are more accurate because they do not work with one side.

**Learning Objectives**

We will work with Riemann sums to approximate the integral of a function. We will also practice working with functions. To create the code, we will be setting conditional statements and use loops.

**Approach**

We will start by setting our conditions. User input will be needed to identify the lower and upper bounds along with the number of rectangles that will be used. Once these are set, we will create a parameter and then form a loop to calculate the approximation.

**Mathematical Concepts**

Mathematical concepts include finding the width of the rectangles by finding delta X. We will also calculate a formula to find the height of the rectangle. Once these are found, we will calculate said rectangle and add all rectangle together. The height will depend on what type of Riemann sum we are using.

**Program Inputs**

**Lab10.m**

This file asks for user input for the variables a,b, and n. It also asks for the user to input a function, but we have been told to just input the function outselves. The function is: [f=@(x)2.\*x.^5-5.\*x.^3-10.\*x+9](mailto:f=@(x)2.*x.%5e5-5.*x.%5e3-10.*x+9);. The variable a is the lower limit of integration, b is the upper limit of integration and n is the interval/step size.

**Left\_Riemann.m**

This function takes in 4 parameters. These parameters are f, a, b, and n. Parameter f is the function to find the area under. Parameter a is the lower limit of integration. Parameter b is the upper limit of integration. Finally, parameter n is the interval used in the evaluation of the Riemann sum.

**Right\_Riemann.m**

This function takes in 4 parameters. These parameters are f, a, b, and n. Parameter f is the function to find the area under. Parameter a is the lower limit of integration. Parameter b is the upper limit of integration. Finally, parameter n is the interval used in the evaluation of the Riemann sum.

**Middle\_Riemann.m**

This function takes in 4 parameters. These parameters are f, a, b, and n. Parameter f is the function to find the area under. Parameter a is the lower limit of integration. Parameter b is the upper limit of integration. Finally, parameter n is the interval used in the evaluation of the Riemann sum.

**Trapezoidal\_Method.m**

This function takes in 4 parameters. These parameters are f, a, b, and n. Parameter f is the function to find the area under. Parameter a is the lower limit of integration. Parameter b is the upper limit of integration. Finally, parameter n is the interval used in the evaluation of the trapezoidal sum.

**Program Outputs**

**Lab10.m**

Prints out the left Riemann Sum, right Riemann Sum, middle Riemann Sum, and the trapezoidal sum using fprintf. Plots the given function using fplot over the interval [a,b].

**Left\_Riemann.m**

Passes back the calculated Left Riemann sum to the calling function in the variable ‘leftSum’.

**Right\_Riemann.m**

Passes back the calculated Right Riemann sum to the calling function in the variable ‘rightSum’.

**Middle\_Riemann.m**

Passes back the calculated Left Riemann sum to the calling function in the variable ‘middleSum’.

**Trapezoidal\_Method.m**

Passes back the calculated Left Riemann sum to the calling function in the variable ‘trapezoidalSum’.

**Program Description**

**Lab10.m**

This file asks for user input for the variables a,b, and n. It also asks for the user to input a function, but we have been told to just input the function outselves. The function is: [f=@(x)2.\*x.^5-5.\*x.^3-10.\*x+9](mailto:f=@(x)2.*x.%5e5-5.*x.%5e3-10.*x+9);. The variable a is the lower limit of integration, b is the upper limit of integration and n is the interval/step size.

**Left\_Riemann.m**

This program uses a function, lower limits, upper limits, and interval to calculate the left Riemann Sum.

**Right\_Riemann.m**

This program uses a function, lower limits, upper limits, and interval to calculate the right Riemann Sum.

**Middle\_Riemann.m**

This program uses a function, lower limits, upper limits, and interval to calculate the middle Riemann Sum.

**Trapezoidal\_Method.m**

This program uses a function, lower limits, upper limits, and interval to calculate the Trapezoidal Sum.

**Source Code**

**Lab10.m**

f=@(x)2.\*x.^5-5.\*x.^3-10.\*x+9;

a=input('Input lower limit: ');

b=input('Input upper limit: ');

n=input('Input the interval: ');

left\_Riemann=Left\_Riemann(f,a,b,n);

right\_Riemann = Right\_Riemann(f,a,b,n);

middle\_Riemann = Middle\_Riemann(f,a,b,n);

trapezoidal\_Method = Trapezoidal\_Method(f,a,b,n);

fprintf('Left Riemann Sum: %f\n',left\_Riemann);

fprintf('Right Riemann Sum: %f\n',right\_Riemann);

fprintf('Middle Riemann Sum: %f\n',middle\_Riemann);

fprintf('Trapezoidal Method: %f\n',trapezoidal\_Method);

fplot(f,[a,b]);

**Left\_Riemann.m**

function [ leftSum ] = Left\_Riemann( f,a,b,n )

dx=(b-a)/n;

leftSum=0;

for i = a:dx:b-dx

leftSum = leftSum + f(i)\*dx;

end

end

**Right\_Riemann.m**

function [ rightSum ] = Right\_Riemann( f,a,b,n )

dx=(b-a)/n;

rightSum=0;

for i = a+dx:dx:b

rightSum = rightSum + f(i)\*dx;

en0064

end

**Middle\_Riemann.m**

function [ middleSum ] = Middle\_Riemann( f,a,b,n )

dx=(b-a)/n;

middleSum=0;

for i = a+dx\*.5:dx:b-dx\*.5

middleSum = middleSum + f(i)\*dx;

end

end

**Trapezoidal\_Method.m**

function [ trapezoidalSum ] = Trapezoidal\_Method(f,a,b,n)

dx=(b-a)/n;

trapezoidalSum=f(a)+f(b);

for i = a+dx:dx:b-dx

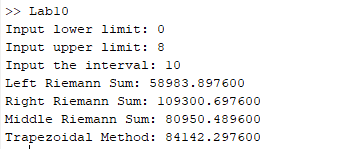
trapezoidalSum = trapezoidalSum + 2\*f(i);

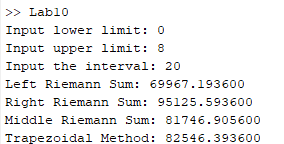
end

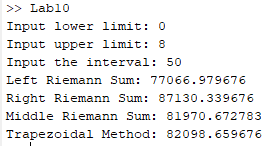
trapezoidalSum= trapezoidalSum \* (dx/2);

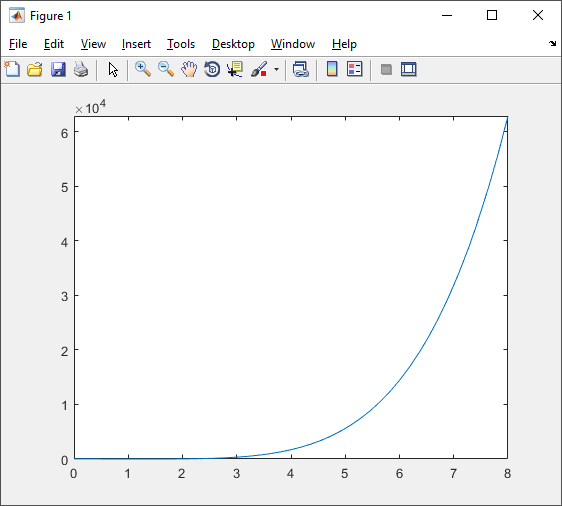
end

**Code Execution Results**









Using the rsums(f,a,b) function, we found that the value shown to us was the same as our calculated value for the middle Riemann sum. It did not show or match any other sum that was printed out.

When n is equal to 10, we saw that the left Riemann sum was far below the rest. The right Riemann sum was significantly larger than the rest of the calculated values. The middle and trapezoidal sums were quite close to each other in comparison to the other two sums, however, they were still not equal.

The approximation gets much larger as we increase the n value. This can be due to the calculation being much more accurate to what the actual area under the curve is. When we have a larger n variable, there are more rectangles, and this makes it much more accurate. 10 rectangles provide the least accuracy, while 50 will show a much more accurate value.

**Conclusions**

This lab was a little confusing to start out with because we had to come up with the concept of how to find a Riemann sum. After logical thinking we identified what our iterator should be and were able to write the code that would calculate each different sum. This was an interesting lab because we discovered how to calculate Riemann sums via code. We have expanded our skills in for loops and have learned how we can use MATLAB to create/calculate functions.