Math 105B Lab Report 1

Haley Feng ID: 71813343

Objective

The goal of this project is to write a function in Matlab for interpolation using Lagrange polynomials. By examining the absolute difference between the actual points and the interpolated values computed by Lagrange and plotting the actual function in comparison to the interpolated values, I will achieve a better understand at how using difference values for the Lagrange interpolants provide a trend in receiving a closer estimate to the true values and a more identical approximated plot to the exact function.

Introduction

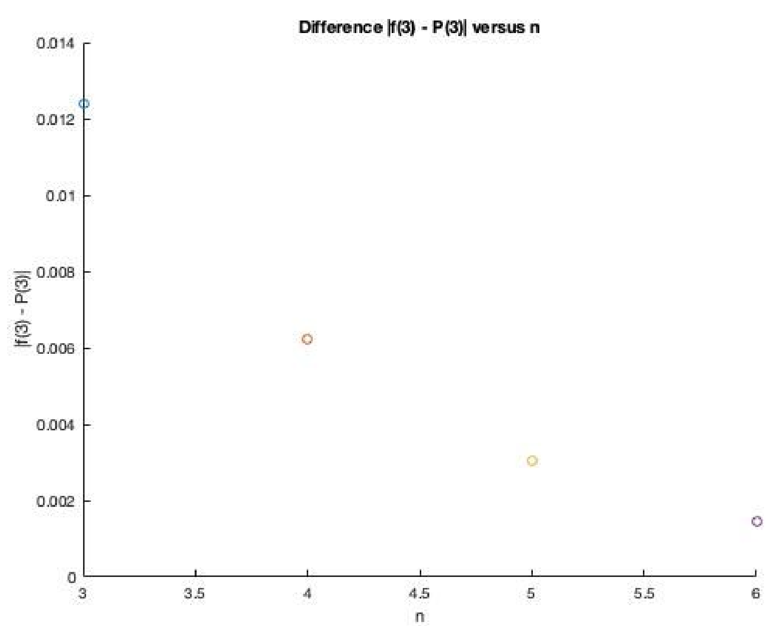
I will first be coding out the Lagrange interpolation polynomial as the function f_lagrange that will take input X, Y and a point x and P(x) as output. Using the function f(x) = 1/x as an example for the model, I will then compute the difference of f(x) and P(x) by evaluating both function at 3 using the degree n = 3 for the convenience of plotting. Furthermore, I will plot my first graph for the absolute differences between f(x) and P(x) versus different Lagrange interpolants i.e. n = 3, 4, 5, and 6. Lastly, I will modify the algorithm method in order to produce my second graph which is a single plot showing the exact function f(x)=1/x and the Lagrange interpolants using n=3, 4, and 6.

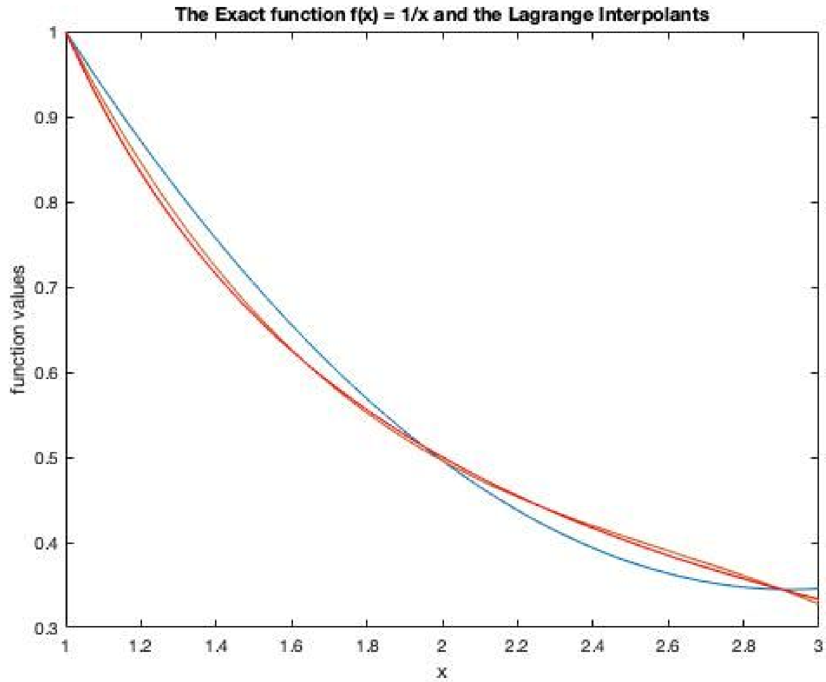
Algorithm Method

Part A: Since the Lagrange interpolation polynomial consist of a summation of f(x) and L(x) from 1 to n, I created a variable named sum and set it to 0 to store the terms that will be added at the end of every loop for j. Using the provided assumption that n = 3, I set a for loop that goes from 1 to 3 for the term j. Since L(x) is a product, I created a variable named product and set it to 1 to store the terms that will be multiplied at the end of every loop for k. Then I set my second and last for loop that goes from 1 to 3 for the term k with an if statement to make sure that the jth term and the kth term is not the same in order to omit getting a zero for the formula. The next few lines consist of ending the if statement and the two for loops and finally storing the product f(x) with its respected L(x). I end the function by getting P(x) as output variable y to the variable sum. Using the code provided from the professor, I run my function f_lagrange in a for loop and proceed to get y(i) for x(i) in the interpolation points set x. To ensure that my function is accurate, I test out my function by inputting x = 1.95 as provided and receive the output of y = 0.5128. In addition, when I plot the interpolated values and the actual function, I received a plot that contains two lines that resemble close to one another. Checking the absolute difference between f(3) - P(3), I got the result of 0.0124 using n = 3.

Part B: In order to plot the absolute difference between f(x) and P(x) versus n, I created a for loop from 3 to 6 and modify my function by adding an extra input n so I can easily change the polynomial in my function to get the difference and solution of P(x) for different n values. For the second graphs, I simply plot the exact function f(x) = 1/x and the Lagrange interpolants with the same modification of my function but this time I created a for loop skipping over 5 for the code to only go through n values 3, 4, and 6. I plot using the code provided by the professor to get the four lines in a single plot.

Graph





Conclusion

As shown above, I obtain the graphs for the difference of f(3) - P(3) versus n and a single plot of the exact function and the three lagrange interpolants by using the function f_lagrange. Taking note that the Lagrange interpolant lines are very identical to the exact function, I can conclude that the estimation using the Lagrange interpolating polynomial provides a very close approximation to the function f(x) using different lagrange interpolants.