LAB # 10

03/18/2025

1 Introduction

This lab focuses on building and testing out code used to approximate L^2. The code that will be built utilizes the quadrature algorithm that is built into *Scipy*.

2 Pre-Lab

Before lab, we are asked to first consider *Legendre Polynomials*. We are given that these polynomials can be defined by the following three-term recursion.

$$\phi_0 = 1$$

$$\phi_1 = x$$

$$\phi_{n+1} = \frac{1}{n+1}((2n+1)\phi_n(x) - n\phi_{n-1}(x))$$

In the above recursion, n is the order of the polynomial and x is the value at which the polynomial is to be evaluated. For pre-lab, we are to implement this recursion in a sub-routine called *eval_legendre* which takes in the order n and the value x at which we are to evaluate the polynomial. This subroutine is to use the above recursion to produce a vector $\tilde{\mathbf{p}}$ of length $n+1$ which contains the evaluation of the polynomial at x for each order $0 \leq N \leq n$. The code produced can be found in the GitHub repository in the *Lab_10* directory.

3 Lab-Day: Building the L^2 Approximations

The main focus of today's lab is to produce code that evaluates L^2 approximations of functions. This will first be achieved using the provided code for lab as well as the pre-lab code and later developed using variations of both.

3.1 Creating the L^2 Approximation

From class, we know that the polynomial of degree n, $p_n(x)$, that approximates a function $f(x)$ with respect to a weight function $w(x) \geq 0$ on the interval I is given by the following.

$$p_n(x) = \sum_{j=0}^{n} a_j \phi_j(x), \text{ where } a_j = \frac{\langle \phi_j, f \rangle}{\langle \phi_j, \phi_j \rangle} = \frac{\int_I \phi_j(x) f(x) w(x) dx}{\int_I \phi_j^2(x) w(x) dx}$$

In the above expressions, $\phi_j(x)$ are a set of polynomials orthogonal on I with respect to $w(x)$. This will be used in the following exercises.

3.2 Exercises

3.2.1

For the first exercise, we are to use the *scipy.integrate* function *quad* and our pre-lab code to create a one line code that evaluates a_j. This is done by writing subroutines to evaluate both the numerator and the denominator in the formula for a_j. The code written for this part can be found on the GitHub repository.

3.2.2

For this exercise, we are to combine the code from the previous exercise with the (incomplete) provided code for lab. The provided code asks for us to call our function written for pre-lab in order to evaluate the Legendre Polynomials needed and call our functions from the last question in the lambda functions of the `eval_legendre_expansion` function. The completed code can be found on the GitHub repository.

The completed code produced the following plots. The first figure shows the plots of the function $f(x) = e^x$ over the interval $[-1, 1]$ as well as the Legendre Expansion approximation of the same function. The second figure displays the log error of the approximation.

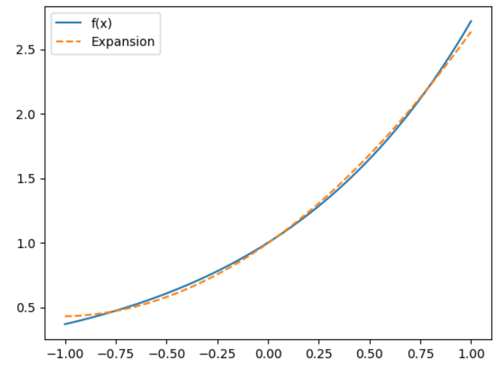

Figure 1: Plot of the function $f(x) = e^x$ and the Legendre Expansion approximation over the interval $[-1, 1]$.

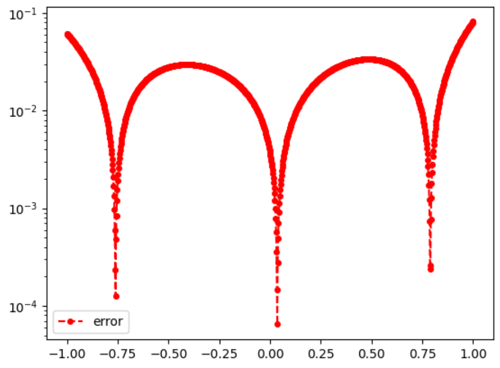

Figure 2: Log Error of Legendre Expansion Approximation

3.2.3

This exercise asks us to know use the same techniques to approximate the function $f(x) = \frac{1}{(1+x^2)}$. The plots produced can be found below.

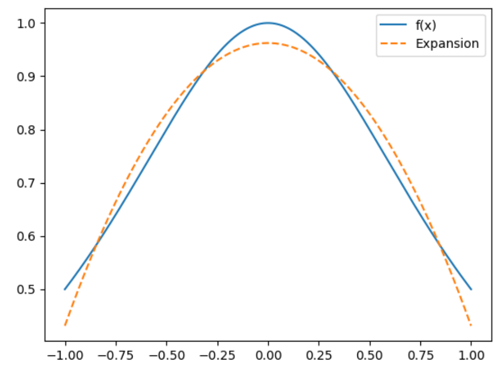

Figure 3: Plot of the function $f(x) = \frac{1}{(1+x^2)}$ and the Legendre Expansion approximation over the interval $[-1, 1]$.

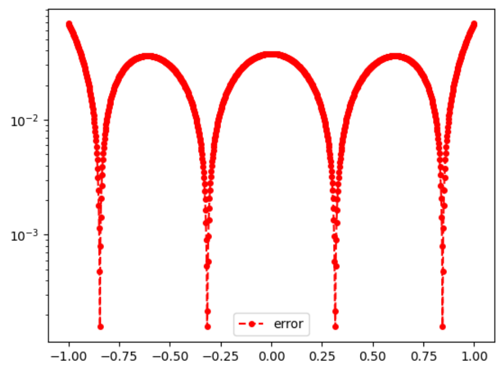

Figure 4: Log Error of Legendre Expansion Approximation

3.3 Additional Exercises

The additional exercises ask us to now approximate the same function as §3.2.3 using new code that approximates L^2 using Chebyshev polynomials. The new three-term recursion is given below.

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_{n+1} = 2xT_n(x) - T_{n-1}(x)$$

The new associated weight function is given by $w(x) = \frac{1}{\sqrt{1-x^2}}$. As before, the code required additional helper functions. The code that was produced can be found in the GitHub repository.

3.3.1

The code produced the following plots where the first is the plot of the function and its approximation and the second is the log error of the approximation.

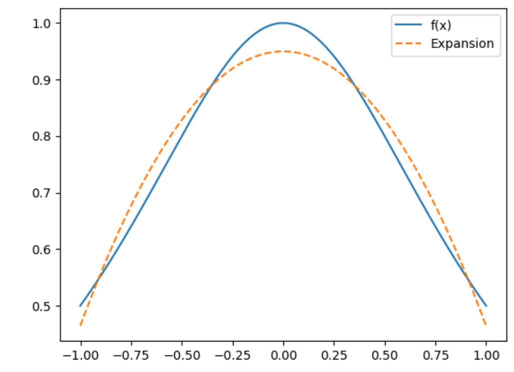

Figure 5: Plot of the function $f(x) = \frac{1}{(1+x^2)}$ and the Chebyshev Expansion approximation over the interval $[-1, 1]$.

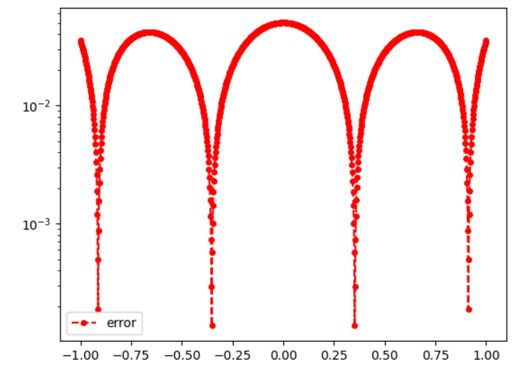

Figure 6: Log Error of Chebyshev Polynomial Expansion Approximation

4 Deliverables

All code, LaTeXcode, and LaTeXrenderings can be found on the GitHub repository under `Lab_10`. The rendered PDF has also been submitted to Canvas.