HOMEWORK # 07

03/07/2025

1 Question 1

This problem focuses on finding the polynomial $p(x) = c_n + c_{n-1}x + c_{n-2}x^2 + \dots + c_1x^{n-1}$ which interpolates the data given by $(x_j, y_j) = (x_j, f(x_j))$, where $j = 1, \dots, n$. I the first part we will derive the system of equations represented by $\mathbb{V}\tilde{\mathbf{c}} = \tilde{\mathbf{y}}$ to determine the coefficients $\tilde{\mathbf{c}} = [c_1, \dots, c_n]^T$ where $\tilde{\mathbf{y}} = [y_1, \dots, y_n]^T$. In the second part, we will then solve for the equations in order to find the polynomial that interpolates a given function $f(x)$ for a given set of points. In addition to finding the polynomial, we will also plot the data points and analyze what happens.

1.1

As mentioned above, in this part we are focusing on finding the matrix $\mathbb{V}$ that represents the system $\mathbb{V}\tilde{\mathbf{c}} = \tilde{\mathbf{y}}$. Let us assume that (x_j, y_j) for $j = 1, \dots, n$ are given. Since we know $p(x)$ from above interpolates the given points, we have n equations as shown below.

$$\begin{aligned} c_n + c_{n-1}x_1 + \dots + c_1x_1^{n-1} &= y_1 \\ c_n + c_{n-1}x_2 + \dots + c_1x_2^{n-1} &= y_2 \\ &\vdots \\ c_n + c_{n-1}x_n + \dots + c_1x_n^{n-1} &= y_n \end{aligned}$$

If we let $\tilde{\mathbf{c}}$ denote the vector of the coefficients in the system and $\tilde{\mathbf{y}}$ represent the y-component of the given points $y_1, \dots, y_n$, we get the following.

$$\begin{aligned} \tilde{\mathbf{c}} &= \left[c_1, c_2, \dots, c_n\right]^T \\ \tilde{\mathbf{y}} &= \left[y_1, y_2, \dots, y_n\right]^T \end{aligned}$$

If we let $\mathbb{V}$ represent the Vandermonde matrix of the above system, we have the following.

$$\mathbb{V} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix}$$

Using this, we can represent the system as $\mathbb{V}\tilde{\mathbf{c}} = \tilde{\mathbf{y}}$.

1.2

As mentioned above, for this problem, we are using Part (a) to find the polynomial that interpolates the following function.

$$f(x) = \frac{1}{1 + (10x)^2}$$

We are to find the polynomial that interpolates the above function, $f(x)$, in the points $x_i = -1 + (i-1)h$ where $h = \frac{2}{N-1}$ for $i = 1, \dots, N$. Afterwards, we are to plot the data and analyze what happens with a finer grid of 1001 points, and when we increase the value of N. To accomplish the above task, python was used to write the necessary code. Helper functions were written based on the code provided for *Lab_07*. The helper functions include one to create the necessary Vandermonde matrix, $\mathbb{V}$, which used the *numpy* function *vander()* and another to find the coefficients and solve for the polynomial and evaluate which uses the *numpy* functions *linalg.solve()* and *polyval()*. The code produced can be found in the Github repository under *Homework_07*.

1.2.1 Plotting and analyzing

Using the code described above, two plots were created. The first plot shown below in 1 shows the original function, $f(x)$, and the interpolated polynomials with values of $N = 2, 3, 4, 5, 6, 8$ while the second plot 2 shows the original function with the interpolated polynomials with values of $N = 12, 14, 16, 18, 20$. As N increases, the interpolated polynomials more closely resemble the original function $f(x) = \frac{1}{1+(10x)^2}$. It is important to note, however, that the *best* looking interpolated polynomials depends on the region of the interval $[-1, 1]$ you are considering. For the lower values of N, the odd values of N tend to match the original function more closely when x is closer to 0 while the even values more closely match when x is further from 0. For the odd values, the more extreme error closer to the endpoints of the interval reflect Runge's phenomena which is a product of the function used.

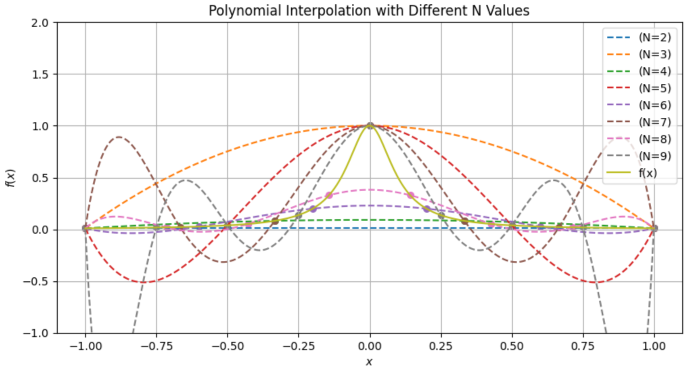

Figure 1: Plot of interpolated polynomials with $N = 2, 3, 4, 5, 6, 7, 8, 9$ and the function f(x).

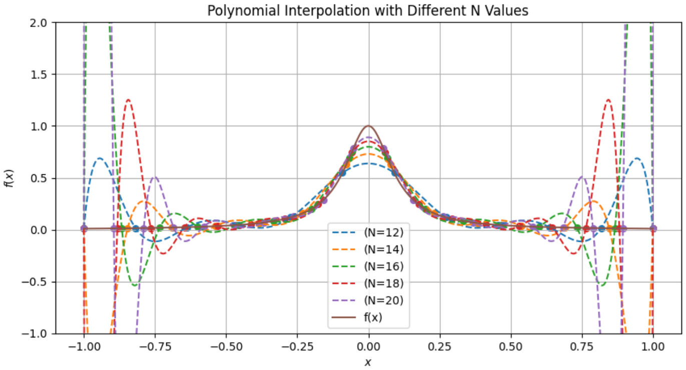

Figure 2: Plot of interpolated polynomials with $N = 12, 14, 16, 18, 20$ and the function f(x).

As shown above, as N continues to increase, the interpolated polynomials continue to more closely match the function $f(x)$.

2 Question 2

As we learned from both class and lab, solving interpolation problems using the monomoial basis as done in Part (a) is *"notoriously ill-conditioned"*. To combat this, a better way to interpolate includes either of the Barycentric Lagrange Interpolation formulas shown below.

$$p(x) = \Phi_n(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j} f(x_j)$$

$$p(x) = \frac{\sum_{j=0}^{n} \frac{w_j}{x-x_j} f(x_j)}{\sum_{j=0}^{n} \frac{w_j}{x-x_j}}, \text{ for } x \neq x_j$$

In the above formulas, we have $\Phi_n(x) = \prod_{i=0}^{n}(x - x_j)$ and $w_j = \frac{1}{\prod_{i=0, i\neq j}(x_j - x_i)}$. In the following section, we will use the second formula to again interpolate $f(x)$ from Question 01 Part (b) and plot and analyze the interpolation.

2.1 Building the interpolation

As done for Question 1, the implementation of the Barycentric Lagrange Interpolation method was written in python with extensive use of *Numpy*. For this implementation, I chose to use the second formula listed above and can be found in the Github repository in the *Lab_07* directory along with the rest of the code.

2.2 Interpolation output

As stated in the Homework Assignment, this method of interpolation cannot dilute the effects of Runge's Phenomenon for nodes near the endpoints. What this method does, however, is produce more stable approximations for large values of N when evaluating at small x. This can be seen in the two plots below. The first plot is of the Monomial expansion method used in Question 1 while the second is of the Barycentric method used in this question.

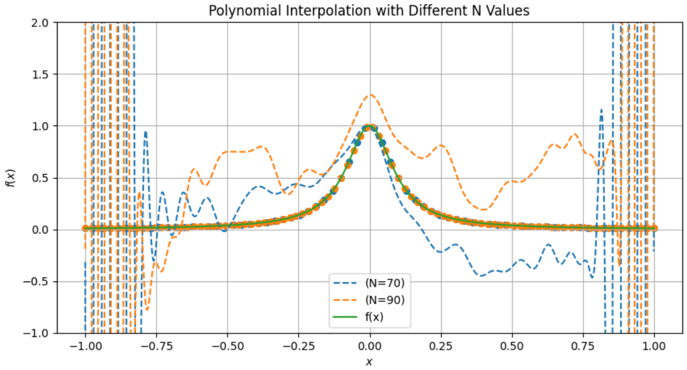

Figure 3: Plot of interpolated polynomials with $N = 70, 90$ and the function f(x).

As shown in 3 above, when N is substantially large, the interpolations fail to behave properly for sufficiently low values of x. That is, the interpolations act erratically for values of x close to 0. This, however, is not the case for the Barycentric Lagrange interpolations as shown below in 4. As shown above, for substantially large values of N, the Barycentric Lagrange interpolations behave properly for sufficiently small values of x near 0.

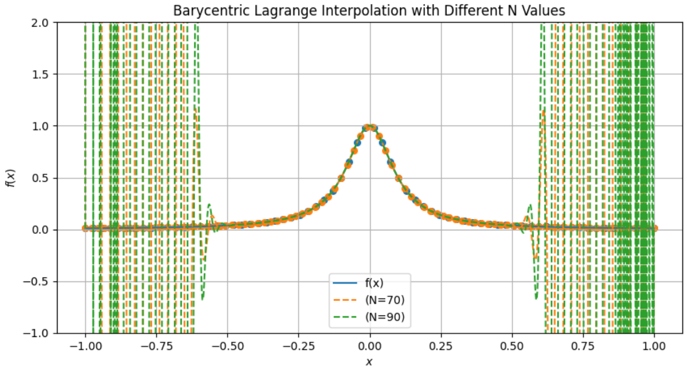

Figure 4: Plot of interpolated barycentric Lagrange polynomials with $N = 70, 90$ and the function f(x).

3 Question 3

For this question, we will again interpolate $f(x)$ from Question 01 Part (b) using the Barycentric Lagrange Interpolation formula used in Question 02. The difference, however, is that we will now interpolate $f(x)$ in the Chebyshev points given below.

$$x_j = \cos\left(\frac{(2j-1)\pi}{2N}\right), \text{ where } i = 1, \dots, N$$

.

3.1 Implementing the Chebyshev points

To implement the Chebyshev points such that I could use them in place of the equispaced points, I wrote a helper function that used the above formula. The code can be found in the Github repository under *Lab_07*.

3.2 Interpolation output

The use of the Chebyshev points in lieu of the equispaced points for both the Monomial interpolation and the Barycentric Lagrange interpolation yielded far more stable results for low and moderately high values of N as shown below in figures 10, 6, 7, and 8.

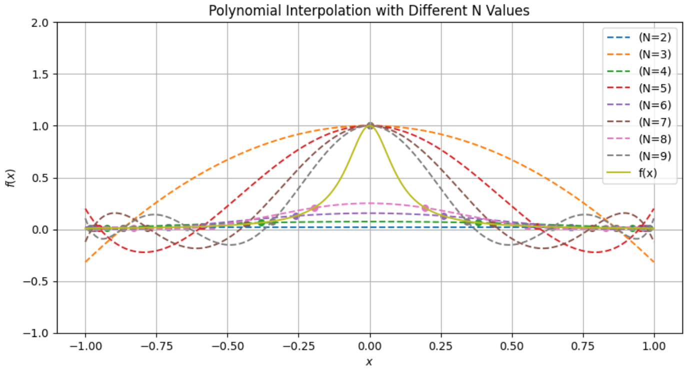

Figure 5: Plot of interpolated polynomials with $N = 2, 3, 4, 5, 6, 7, 8, 9$ using Chebyshev points and the function f(x).

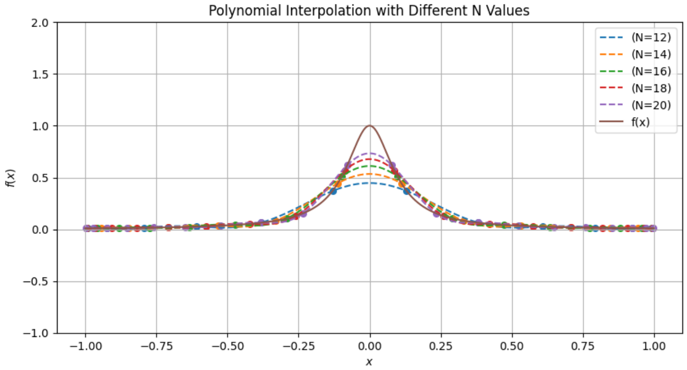

Figure 6: Plot of interpolated polynomials with $N = 12, 14, 16, 18, 20$ using Chebyshev points and the function f(x).

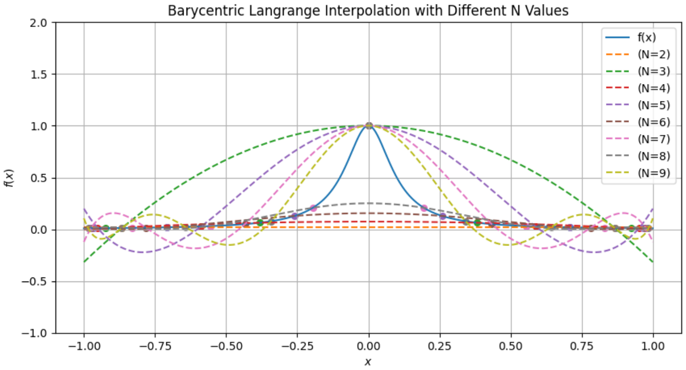

Figure 7: Plot of barycentric Lagrange interpolated polynomials with $N = 2, 3, 4, 5, 6, 7, 8, 9$ using Chebyshev points and the function f(x).

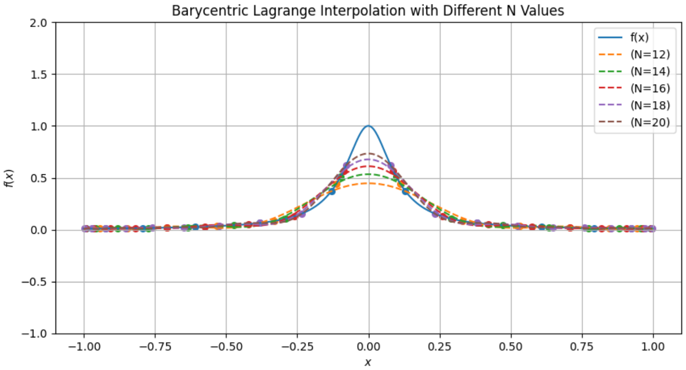

Figure 8: Plot of barycentric Lagrange interpolated polynomials with $N = 12, 14, 16, 18, 20$ using Chebyshev points and the function f(x).

When the value of N was increased as done in Question (3), the use of the Chebyshev points greatly benefited the barycentric Lagrange interpolation as shown in figure 9. The use of the points for the monomial interpolation, however, broke the interpolation similar to that in Question (3) as seen below in figure 10.

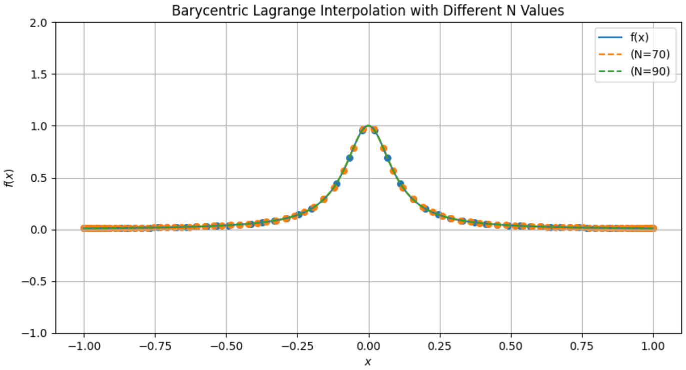

Figure 9: Plot of barycentric Lagrange interpolated polynomials with $N = 70, 90$ using Chebyshev points and the function f(x).

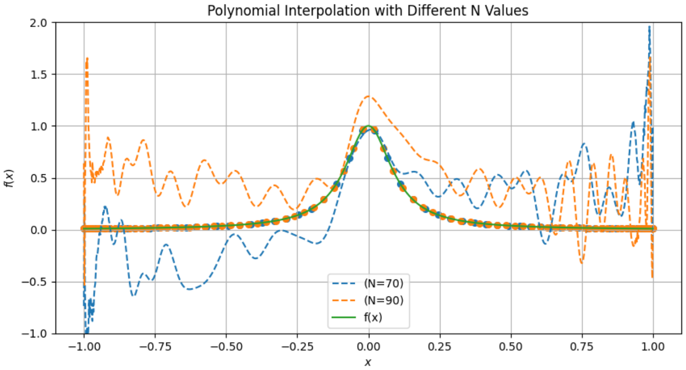

Figure 10: Plot of interpolated polynomials with $N = 70, 90$ using Chebyshev points and the function f(x).