Laboratory 3

Evelyn Wilson

October 8, 2025

1 Introduction

In this lab, we explore the concepts of Neville's Method, uniformly spaced points vs Chebyshev optimal points, parametric interpolation, and cubic spline interpolation.

2 Question 1

In the first question, we perform polynomial interpolation on a function with a varying number of points to interpolate on to decide whether a higher order polynomial approximation is better than a lower order polynomial. We also compare the performance of uniformly spaced points and Chebyshev points.

2.1 Using uniformly spaced points

For the Neville's method, we iterate through n+1 points, where n is the order of the polynomial interpolation and perform the following calculation: for $i = 1, 2, ...n$, for $j = 1, 2, ...,$ i, set

$$Q_{i,j} = \frac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}$$

For this problem, we create an array of n points over the interval -2, 2, uniformly distributed. We start with n values (an array of the interpolation points evaluated in f(x)) and converge, reducing one at a time to one value. We get one of these values for each time we run the Neville's method, which we do for each point we would like to graph. We create an array of 1000 points over the interval to evaluate using the Neville's Method function and evaluate those for each value of N, 5 through 15. We plot these results in Figure 1. Then, we plot the highest order polynomial for clarity in Figure 2. Note the great divergence from the original function at the ends. We will attempt to alleviate this with Chebyshev optimal points in Section 2.2.

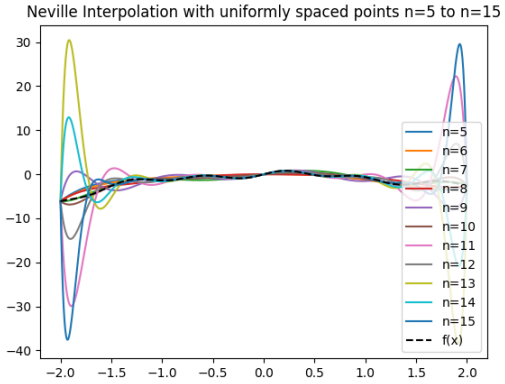

Figure 1: Neville's Method using uniformly spaced points for n = 5 to n = 15

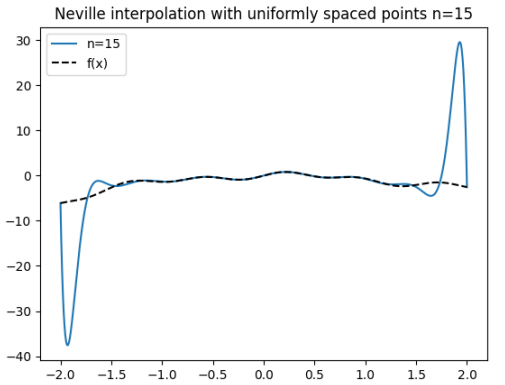

Figure 2: Neville's Method using uniformly spaced points for n = 15

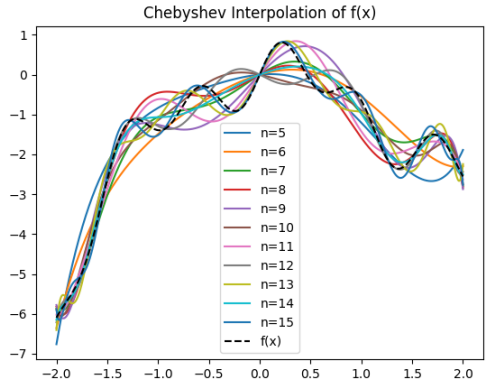

Figure 3: Neville's Method using Chebyshev optimal points for n = 5 to n = 15

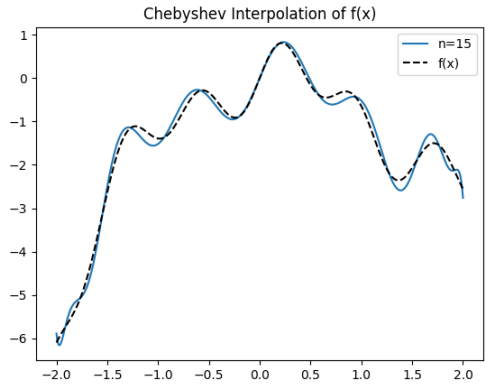

Figure 4: Neville's Method using Chebyshev optimal points for n = 15

2.2 Using Chebyshev optimal points

To find the Chebyshev optimal points, we use the following formula:

$$x_k = cos(\frac{2k+1}{2n}\pi)$$

where $k = 0, ..., n-1$. We then perform the same process of iteration as in Section 2.1 using these points as the interpolation points. We evaluate the function at these points and use these values as the basis of our Neville's Method. The Chebyshev optimal points are more concentrated around the edges, so we can see in Figure 3 and Figure 4that the edges are not as divergent as when we use the uniformly spaced points.

2.3 Evaluation

Now, we evaluate the relative performance of Chebyshev optimal points vs uniformly spaced points and whether a greater order of polynomial interpolation always means better results.

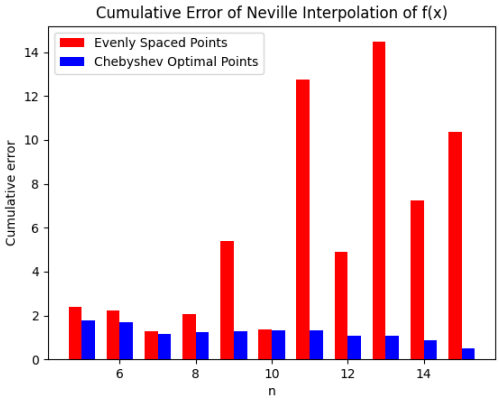

Figure 5: Error comparison for uniformly spaced points vs Chebyshev optimal points for n = 5 to n = 15

As we can see in Figure 5, Neville's method using uniformly spaced points suffers from vastly more error than that using Chebyshev optimal points. We note that the error is calculated as a Riemann sum, so the error is cumulative. For lower orders, the uniformly spaced points perform only marginally better than Chebyshev optimal points, but once we get into higher orders of magnitude, the errors become higher. The opposite is true for Chebyshev optimal points, which actually decrease in error the higher the order is. We can explain this phenomenon easily; the evenly spaced points are not concentrated enough near the end-points and thus more error is accumulating in these areas even as more points are being added. When more points are added to Neville's method using Chebyshev optimal points, they are more likely to be in the end-point region and therefore cause less and less error.

To answer our classmate's question, it is not always better to use a higher order of polynomial interpolation, especially if you are not distributing those points optimally. If we are distributing the points optimally, it is beneficial to evaluate using more points.

3 Question 2

For the second question, we repeat the methodology for the first question using parametric interpolation on a set of points rather than a function. This means we use the provided points as our interpolation points and we perform interpolation for the x- and y-coordinates separately against a third parameter s. We can interpret s as the index

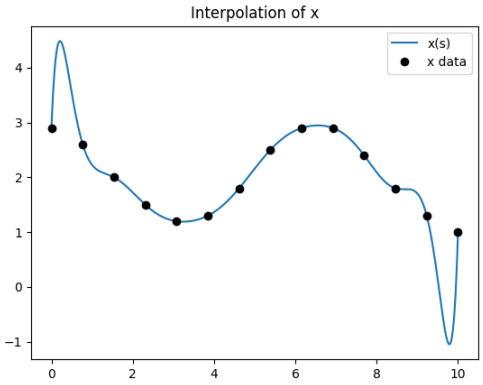

Figure 6: Neville's Method parametric interpolation of x(s)

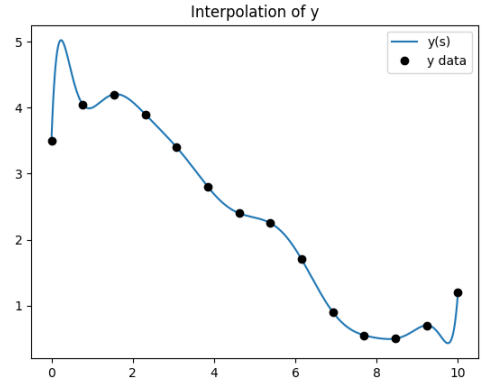

Figure 7: Neville's Method parametric interpolation of y(s)

of the x and y arrays because they are of the same size. We create N evenly spaced values for s across an interval of my choosing (0 to 1), where N is the number of points given and evaluated the interpolation with those. First, we plot x(s) in Figure 6, then we plot y(s) in Figure 7, then we combine the two and plot y(x) in Figure 8. Our Δs is uniform across the interval and the interval for s is [0, 1], so $\Delta s = 0.76923077$.

4 Question 3

To find the cubic spline interpolation of the given set of points, we must first set up a system of matrices. We set up the matrix A as given in the cubic spline handout. Then,we set up an array h to be the differences between values of s (in this case it is uniform). We set the matrix a as the set of points we wish to interpolate. We then set up a matrix b in terms of h, a, and c (which we are

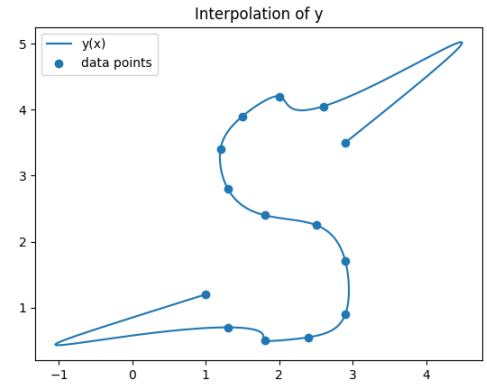

Figure 8: Neville's Method parametric interpolation of y(x)

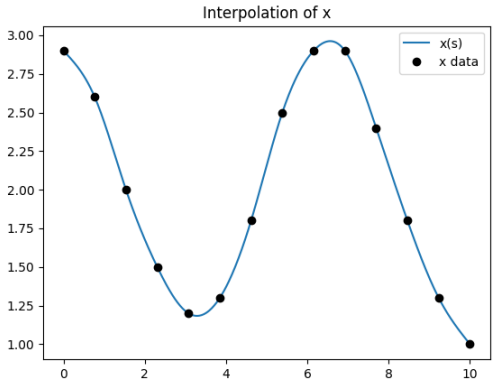

Figure 9: Cubic spline parametric interpolation of x(s)

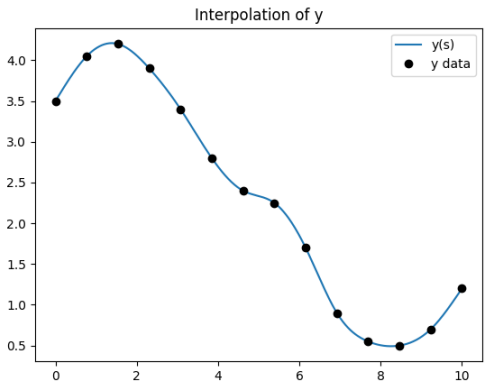

Figure 10: Cubic spline parametric interpolation of y(s)

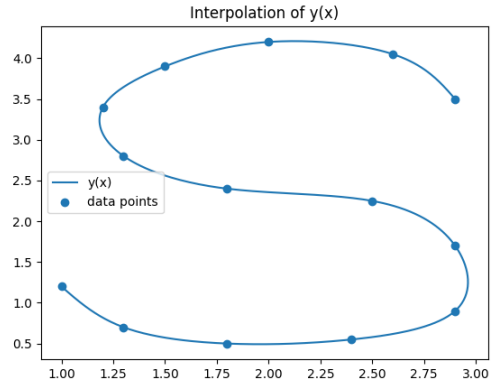

Figure 11: Cubic spline parametric interpolation of y(x)

solving for). We solve for c by using the equation $A * b = c$.
Then, we solve for the remaining coefficients b and d, using a, c, and h. To solve for d:

$$2c_j + 6d_j h_j = 2c_{j+1}$$

$$6d_j h_j = 2(c_{j+1} - c_j)$$

$$d_j = \frac{2(c_{j+1} - c_j)}{6h_j} = \frac{c_{j+1} - c_j}{3h_j}$$

Keep in mind, we are solving for these coefficients for each value of N, which represents an interval over which to evaluate.
Next, we find b using a, c, and h and the expression for d:

$$a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 = a_{j+1}$$

$$b_j = \frac{a_{j+1} - a_j}{h_j} - c_j h_j - d_j h_j^2$$

where:

$$d_j h_j^2 = \frac{c_{j+1} - c_j}{3h_j} h_j^2 = \frac{h_j}{3}(c_{j+1} - c_j)$$

therefore:

$$b_j = \frac{a_{j+1} - a_j}{h_j} - c_j h_j - \frac{h_j}{3}(c_{j+1} - c_j)$$

$$b_j = \frac{a_{j+1} - a_j}{h_j} - \frac{h_j}{3}(c_{j+1} - 2c_j)$$

The coefficients for x are listed in Table 1 and the coefficients for y are listed in Table 2.

j	a_j	b_j	c_j	d_j
0	2.900000	-0.279080	0.000000	-0.187450
1	2.600000	-0.611840	-0.432590	0.278170
2	2.000000	-0.783560	0.209350	-0.046430
3	1.500000	-0.543910	0.102200	0.127240
4	1.200000	-0.160800	0.395840	-0.023140
5	1.300000	0.407110	0.342440	-0.034680
6	1.800000	0.872370	0.262410	-0.277540
7	2.500000	0.783400	-0.378060	0.046330
8	2.900000	0.284010	-0.271150	-0.127480
9	2.900000	-0.359440	-0.565340	0.243910
10	2.400000	-0.796230	-0.002480	0.030650
11	1.800000	-0.745630	0.068260	0.072880
12	1.300000	-0.511250	0.236430	-0.102460

Table 1: Coefficients for x(s)

j	a_j	b_j	c_j	d_j
0	3.500000	0.816460	0.000000	-0.171480
1	4.050000	0.512070	-0.395710	-0.021420
2	4.200000	-0.134750	-0.445150	0.147320
3	3.900000	-0.558080	-0.105190	-0.018590
4	3.400000	-0.752920	-0.148100	0.146760
5	2.800000	-0.720240	0.190580	0.090660
6	2.400000	-0.266110	0.399790	-0.399550
7	2.250000	-0.360310	-0.522250	0.079500
8	1.700000	-1.022650	-0.338790	0.411100
9	0.900000	-0.814100	0.609910	-0.186000
10	0.550000	-0.205960	0.180670	0.003360
11	0.500000	0.077950	0.188420	0.062710
12	0.700000	0.479160	0.333140	-0.144360

Table 2: Coefficients for y(s)

The parametric cubic spline interpolation method proves to be much better than the parametric Neville's method interpolation. In question 2, we observe a lot of oscillation near the edges of the interval, as we observed in question 1 when not using Chebyshev optimal points. The reason the cubic spline interpolation method does so well is that it splits up the interval into smaller chunks and aligns more with local behavior than global behavior while still remaining continuous.