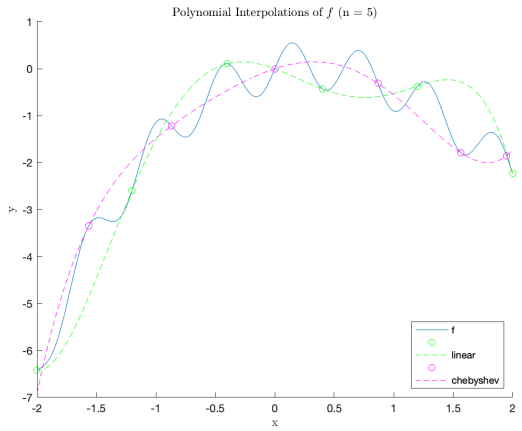
Lab 3

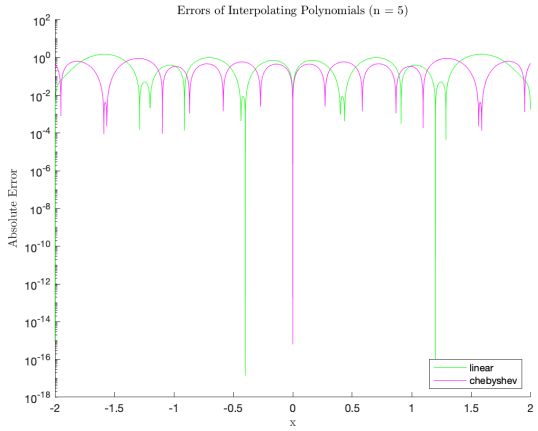
Jessie Li

October 4, 2023

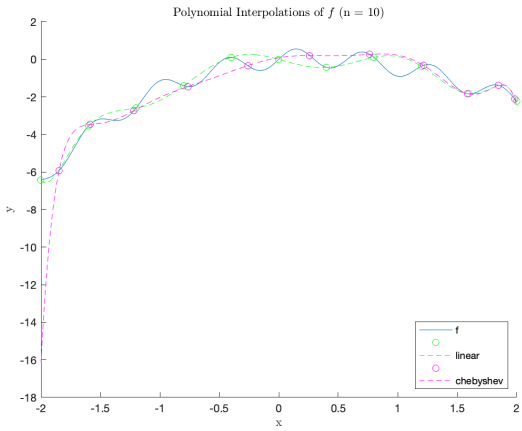
1.

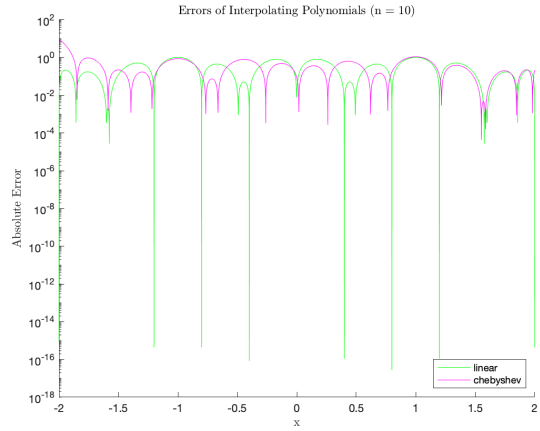
* In case the graphs are too small, the left graph shows the interpolations, and the right graph shows the errors. Function f is blue, linearly-spaced interpolations/errors are green, and Chebyshev interpolations/errors are pink.



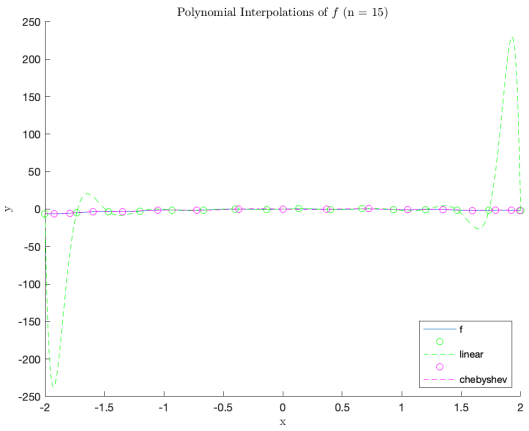


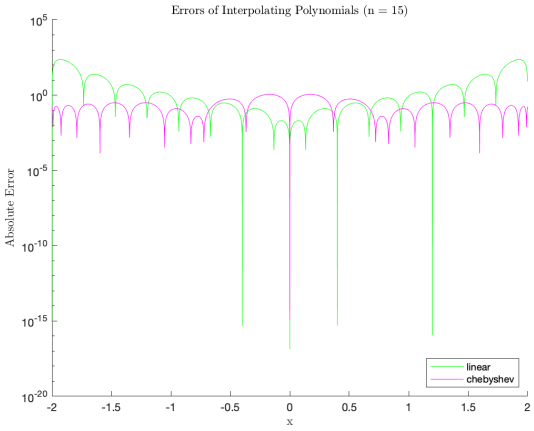
$n = 5$



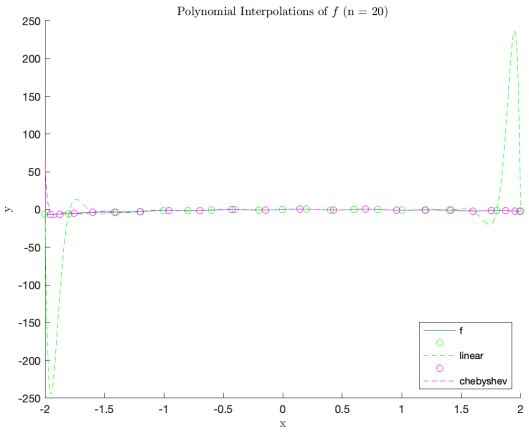


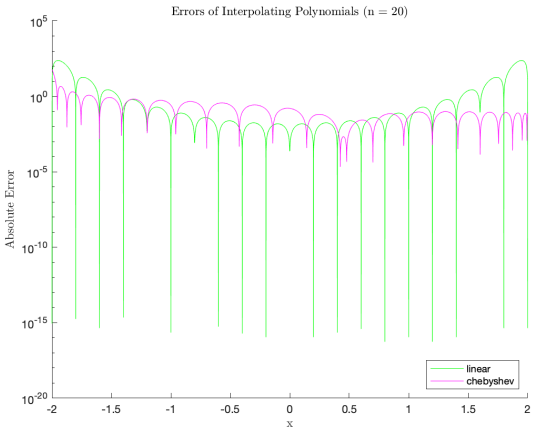
$n = 10$


Polynomial Interpolations of f (n = 15)
f
linear
chebyshev
x
y

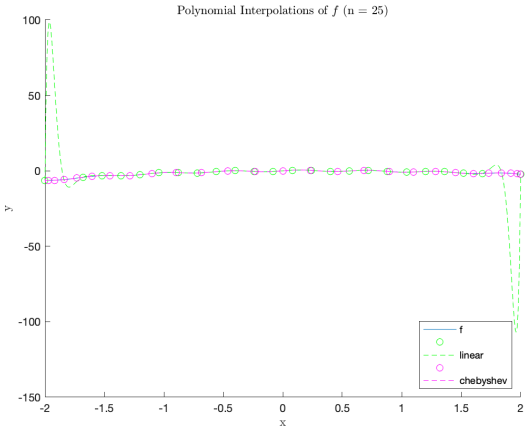

Errors of Interpolating Polynomials (n = 15)
linear
chebyshev
x
Absolute Error

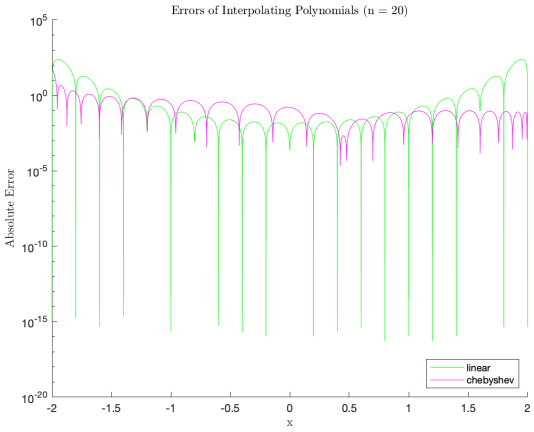
$n = 15$


Polynomial Interpolations of f (n = 20)
f
linear
chebyshev
x
y

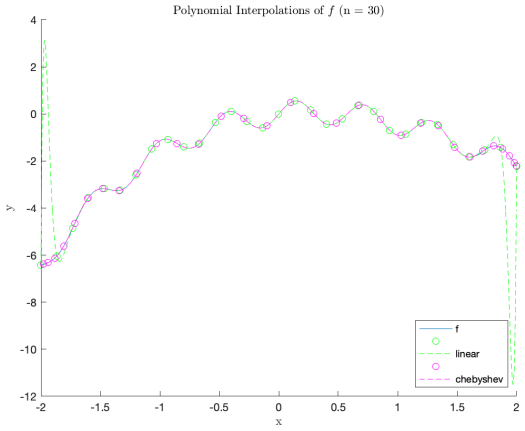

Errors of Interpolating Polynomials (n = 20)
linear
chebyshev
x
Absolute Error

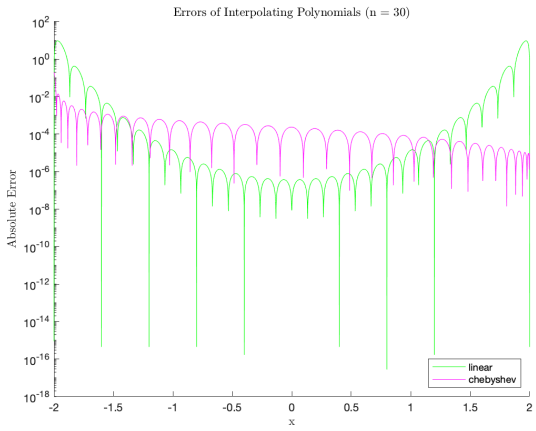
$n = 20$


Polynomial Interpolations of f (n = 25)
f
linear
chebyshev
x
y

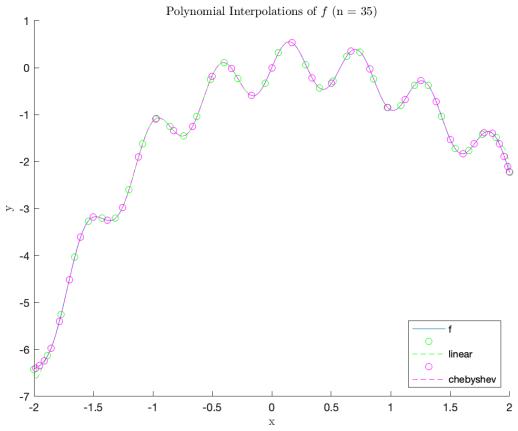

Errors of Interpolating Polynomials (n = 20)
linear
chebyshev
x
Absolute Error

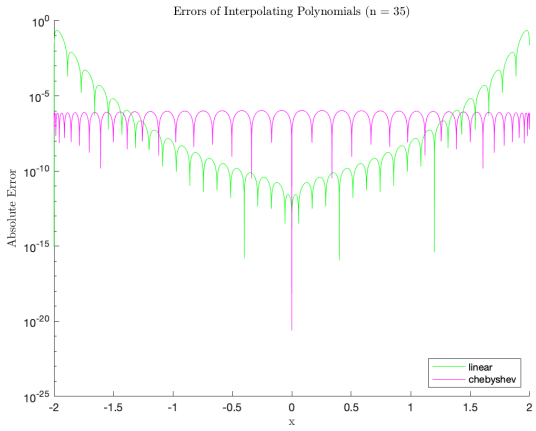
$n = 25$



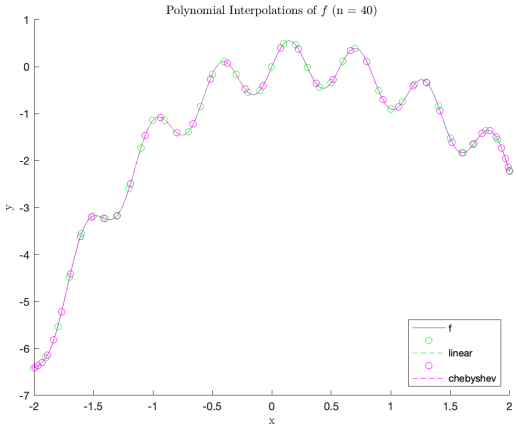


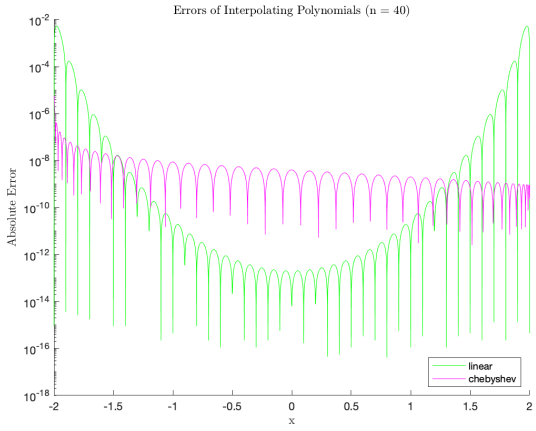
$n = 30$





$n = 35$





$n = 40$

Discuss the merits of each approximation in terms of your results. For example, is the use of Chebyshev optimal points better? What is the behavior of both types of interpolating

polynomials as n increases? Do higher order polynomial approximation to a function is always better than a lower order polynomial?

The difference between using the Lagrange method to interpolate a set of linearly spaced points and the Chebyshev points becomes clearer as n increases:

- The linear and Chebyshev interpolations produce similar error graphs for smaller values of n. Particularly for $n = 5$ and $n = 10$, the error varies in magnitude from 0 at the sampled points to at most 10, and exhibits relatively uniform behavior across the interval. For $n = 10$, the Chebyshev approximation has a maximum error of 9.73 near the left endpoint, higher than the maximum error of the linear approximation, 1.05, but the error graphs are otherwise similar.
- By $n = 15$, the interpolations and error graphs start to diverge. Interpolating at the Chebyshev points still leads to a nearly uniform error throughout the interval on the order of 10^1. Out of the plotted n, the largest occurs at the left interval bound for $n = 20$, reaching about 62.5. Relative to Chebyshev, interpolating at the evenly spaced points leads to a better fit in the middle and a worse fit at the edges. The largest error of the linear approximation, also occurring at the endpoints, increases from 1.05 at $n = 10$ to 231 at $n = 15$ to 238 at $n = 20$.
- Beyond $n = 25$, the Chebyshev interpolation seemingly continues to improve. The maximum error over the interval decreases overall, but not very consistently. For example, the maximum Chebyshev error at $n = 25$ is 0.0162, increases to 0.196 at $n = 30$, decreases to 0.00000 by $n = 35$, and just barely increases to 0.00001 at $n = 40$. In contrast, the linear interpolation seems to continue improving toward the middle, but worsen at the endpoints, as indicated by the deepening error curve. However, the size of the error starts to decrease dramatically with the maximum being 105 at $n = 25$ (lower than 238 at $n = 20$), 9.50 at $n = 30$, then 0.00543 at $n = 40$. These values are still higher than the Chebyshev errors for large n.

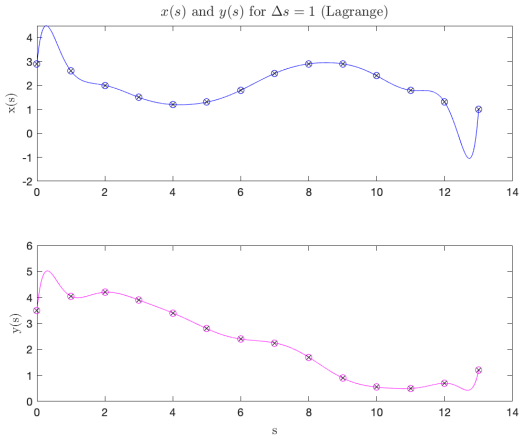
To summarize, for smaller n, the Chebyshev and linear interpolations are comparable. For intermediate values of n, the Chebyshev points seem to give a better fit overall, but the linearly spaced points are better for approximating the middle of the interval. At large values of n, the Chebyshev interpolation generally led to lower errors than the linearly spaced interpolation.

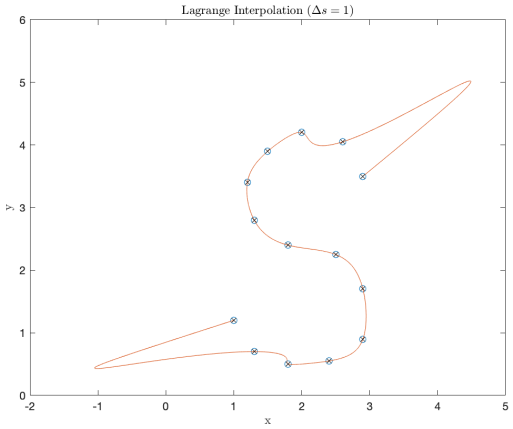
Using a higher order polynomial to approximate a function is not always better than a lower order polynomial. We can see that the maximum error of both approximations increases dramatically from $n = 5$ to $n = 15$ (about 1.52 to 238 for linear, 0.908 to 62.5 for Chebyshev). At extremely large values of n, the errors decrease toward zero, but this could be mostly due to the point density increasing.

2.

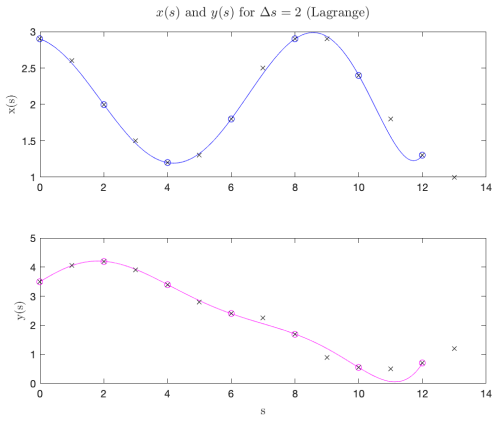
Report the value of Δs you use as well as the maximum and minimum values of $x(s)$ and $y(s)$.

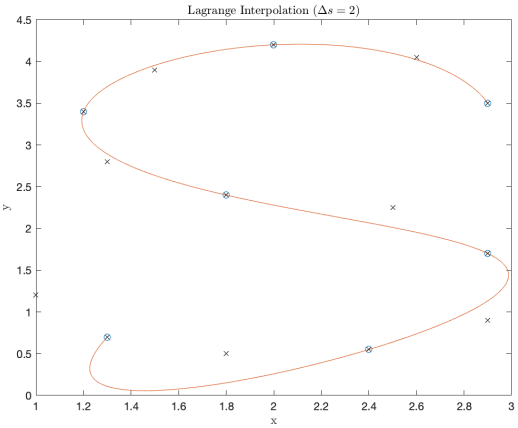
* Leftmost endpoint $(s, x, y) = (0, 2.9, 3.5)$ was included in all samples below
* Interpolated points are marked with 'o'
* All points from the original data set are marked with 'x'
* In case the axes are not legible, the left figure graphs $x(s)$ on top and $y(s)$ on the bottom, and the right figure is the interpolated shape



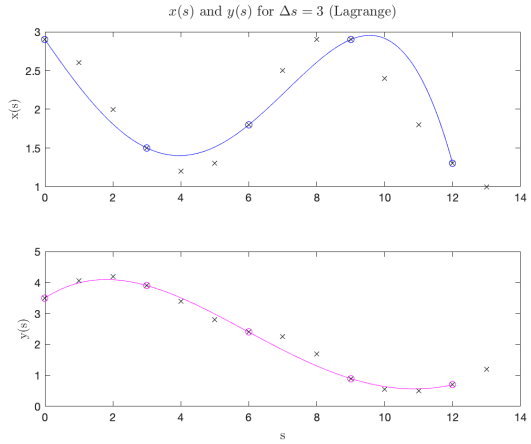


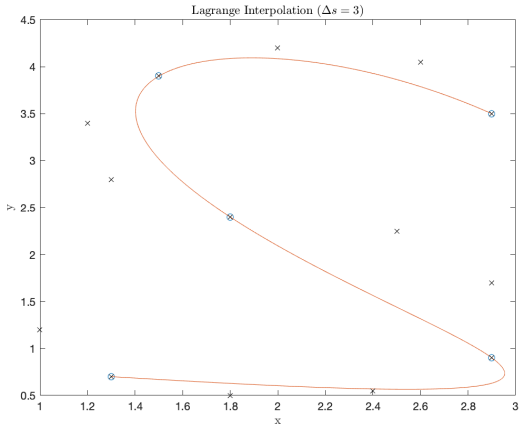
Δs	n	min $x(s)$	max $x(s)$	min $y(s)$	max $y(s)$
1	13	−1.05	4.49	0.439	5.02



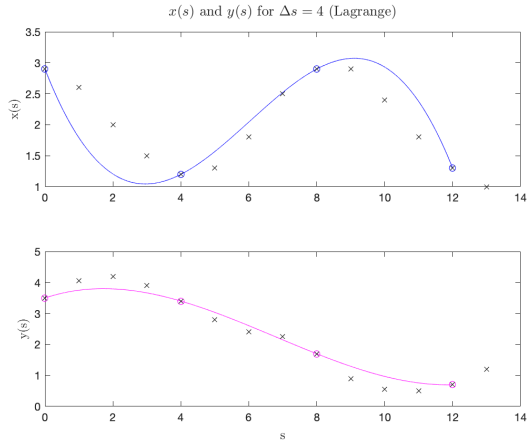


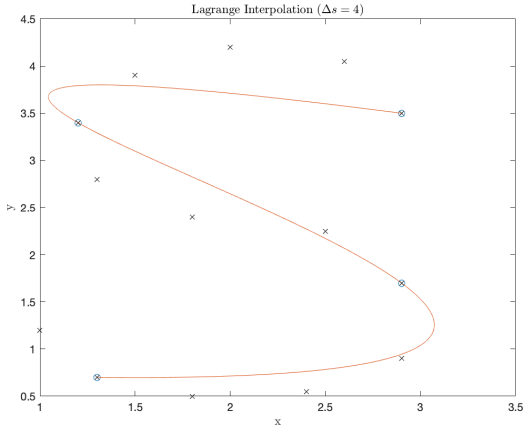
Δs	n	min $x(s)$	max $x(s)$	min $y(s)$	max $y(s)$
2	6	1.19	2.99	0.0565	4.21



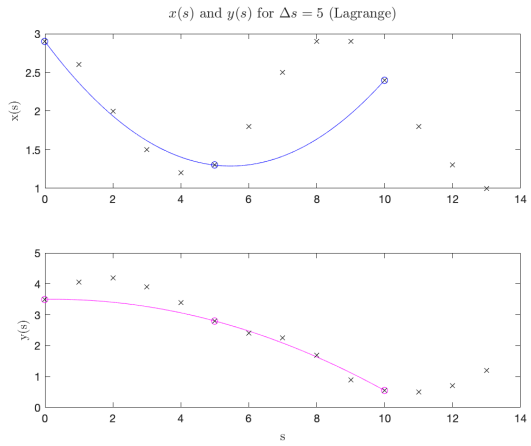


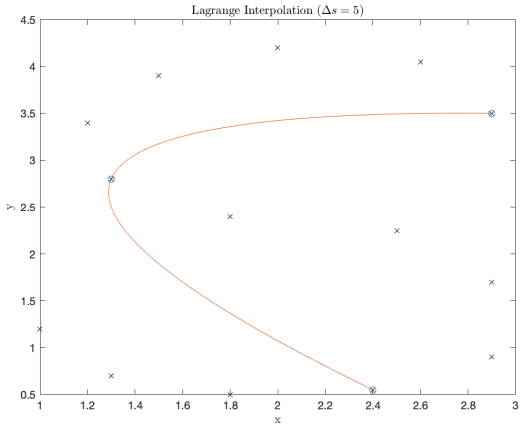
Δs	n	min $x(s)$	max $x(s)$	min $y(s)$	max $y(s)$
3	4	1.30	2.95	0.565	4.09





Δs	n	min $x(s)$	max $x(s)$	min $y(s)$	max $y(s)$
4	3	1.04	3.07	0.698	3.80





Δs	n	min $x(s)$	max $x(s)$	min $y(s)$	max $y(s)$
5	2	1.29	2.90	0.550	3.50

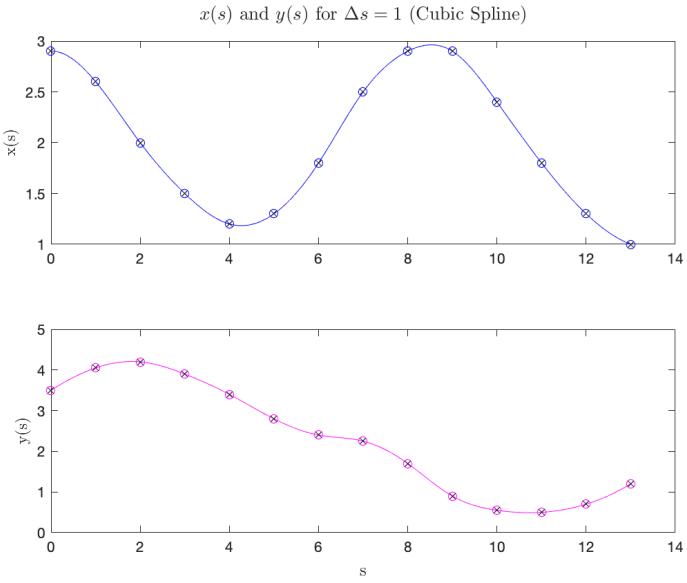
Visually, $\Delta s = 2$ seems to give the most natural-looking interpolation over the interval defined by the data points in a sample with the smallest and largest values of s. While an interpolation with $\Delta s = 1$ passes through all 14 data points, there are large oscillations at the edges of the interpolated shape, whereas for $\Delta s = 2$, the behavior at the endpoints is much smoother. For $\Delta s > 2$, the fit of the interpolation seems to worsen; although the true function is unknown, the interpolated shape seems to get increasingly further away from the shape outlined by the data points. Aside from decreasing the order of the interpolated polynomials, this could partly be due to what points were used for the interpolation. For instance, I used three, uniformly-spaced points towards the left of the full interval for $\Delta s = 5$. Had I chosen three toward the center, or to the right, or increased the spacing between the points, the interpolations would probably look different. Regardless, the $\Delta s = 2$ interpolation seems to follow the path of the points best.

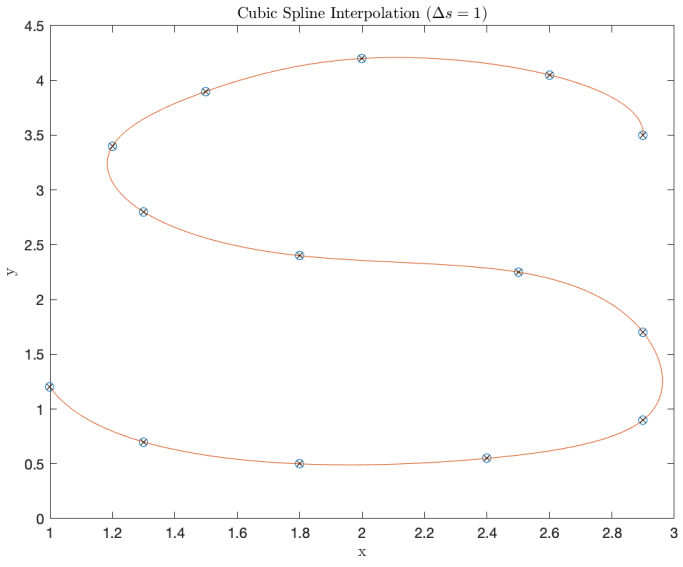
3.

Repeat problem (2) with a natural cubic spline. In addition to showing similar plots and reporting the corresponding values, also list all 4 coefficients for each of the cubics which comprise the interpolants for both x(s) and y(s). How does your letter compare with that produced in problem (2)? Explain any differences.

* Coefficients [a, b, c, d] of the cubic spline correspond with the polynomial:

$$f(x) = a(x - x_i)^3 + b(x - x_i)^2 + c(x - x_i) + d$$





Δs	min $x(s)$	max $x(s)$	min $y(s)$	max $y(s)$
1	1.00	2.96	0.49	4.21

	x coefficients				y coefficients			
	a	**b**	**c**	**d**	**a**	**b**	**c**	**d**
0	0.082	-0.395	0.014	2.900	-0.024	-0.127	0.702	3.500
1	0.082	-0.150	-0.532	2.600	-0.024	-0.200	0.374	4.050
2	-0.009	0.095	-0.586	2.000	0.071	-0.273	-0.098	4.200
3	0.055	0.068	-0.423	1.500	-0.010	-0.060	-0.431	3.900
4	-0.010	0.232	-0.123	1.200	0.067	-0.088	-0.579	3.400
5	-0.016	0.203	0.313	1.300	0.041	0.113	-0.554	2.800
6	-0.126	0.155	0.671	1.800	-0.182	0.236	-0.205	2.400
7	0.021	-0.224	0.603	2.500	0.036	-0.309	-0.277	2.250
8	-0.058	-0.161	0.218	2.900	0.188	-0.201	-0.787	1.700
9	0.110	-0.334	-0.276	2.900	-0.086	0.362	-0.626	0.900
10	0.018	-0.004	-0.614	2.400	0.007	0.104	-0.160	0.550
11	0.016	0.051	-0.567	1.800	0.009	0.124	0.067	0.500
12	0.016	0.100	-0.416	1.300	0.009	0.150	0.341	0.700

For cubic splines, more data points provide a better interpolation because the method approximates segments (with a cubic) and requires the resulting interpolation to be continuous with continuous first and second derivatives, so I used all 14 data points (Δs = 1, 13 segments).

Compared to the Lagrange interpolations generated in Question 2, the cubic spline seems to result in a smoother approximation overall; the cubic spline “naturally” follows the trajectory of the points, without wild oscillations. Even the most “natural” Lagrange interpolation with Δs = 2 seems to have sharper turns than the cubic spline.