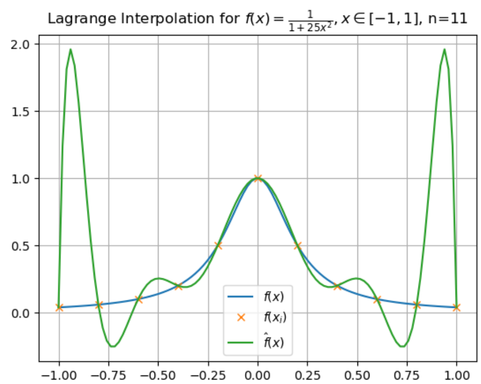
1. Using Python, a Lagrange Polynomial is computed to interpolate the function $f(x) = \frac{1}{1+25x^2}$ on the domain $x \in [-1, 1]$ for $n = 11, 21$ evenly spaced points respectively. Plots comparing the original function and the interpolated functions are shown below in Figure 1. Here we an clearly see the Runge phenomenon causing large fluctuations in the interpolant near the end points. Increasing the number of points/intervals does not fix the fluctuations, instead compressing them closer to the endpoints and increasing their magnitude.



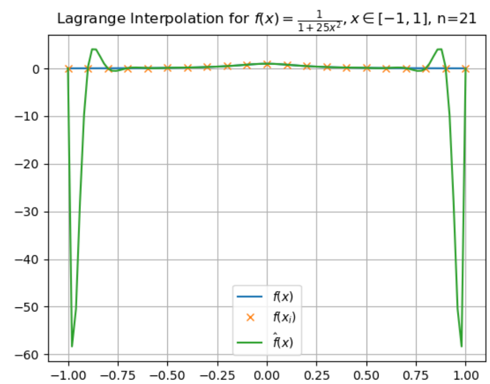


Figure 1: Comparing $f(x)$ with the Lagrange polynomial interpolated $\hat{f}(x)$ for $n = 11, 21$.

2. We now compute a natural cubic spline to interpolate Runge's function: $f(x) = \frac{1}{1+25x^2}$ on the domain $x \in [-1, 1]$ for $n = 11, 21$ evenly spaced points respectively. Plots comparing the original function and the interpolated functions are shown below in Figure 2. We see that Runge's phenomenon is no longer present, even for $n = 11$ equidistant points. The 0th, 1st, and 2nd order continuity conditions for the spline introduce a global coupling into the interpolation, and we see that spline fit is smooth and without fluctuations. Analyzing the change in maximum error between the spline and the original function $f(x)$ as we increase the number of points/intervals in Figure 3, we see that the error decays exponentially as the number of points increases, hence the size of the intervals decreases.

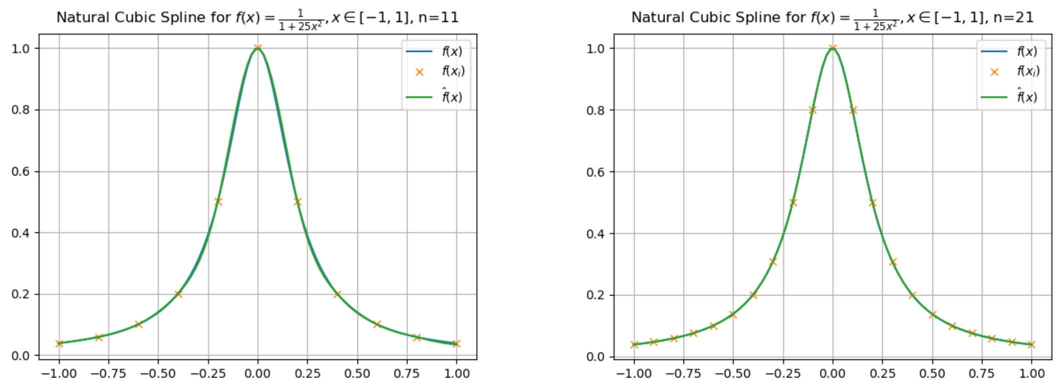


Figure 2: Comparing $f(x)$ with the natural cubic spline interpolated $\hat{f}(x)$ for $n = 11, 21$.

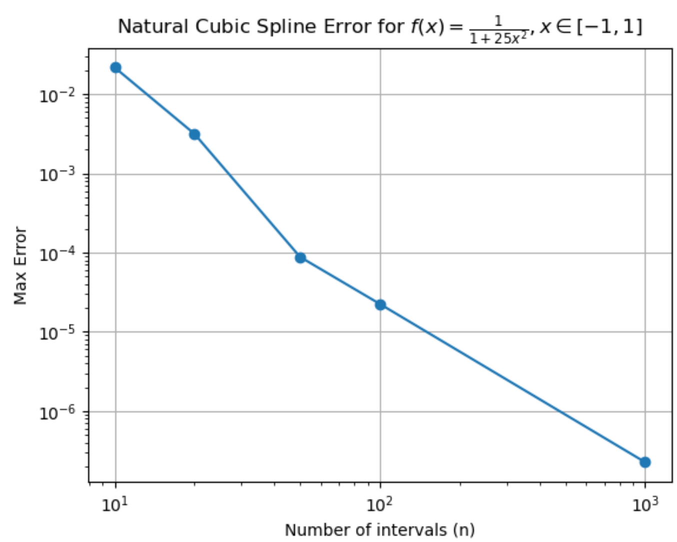


Figure 3: Convergence of maximum error as the number of intervals is increased, hence interval size is decreased.