Jon Pilling

HW4

CS3200

1. Polynomial Interpolation using Vandermonde
   1. See Matlab
   2. See Matlab
   3. L = 2 always give the best fit for our plots. When l was equal to 1, we got a horizontal line through the noisy data. When L was equal to Nd-1 we got a good fit through the middle of the data, but the ends of our polynomial were heading towards extrema max and min. The first graph with only 3 data points and l equaling 2 was the closest fit polynomial I got.

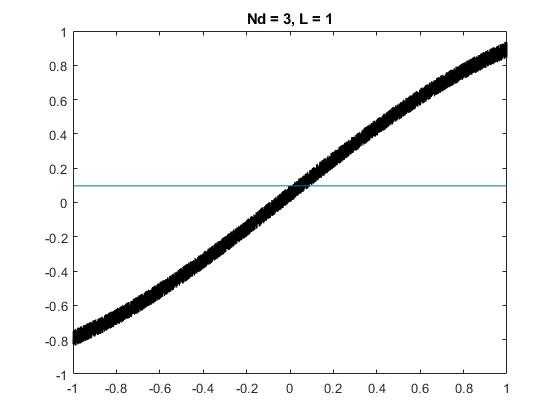


Figure . Nd = 3, l = 1

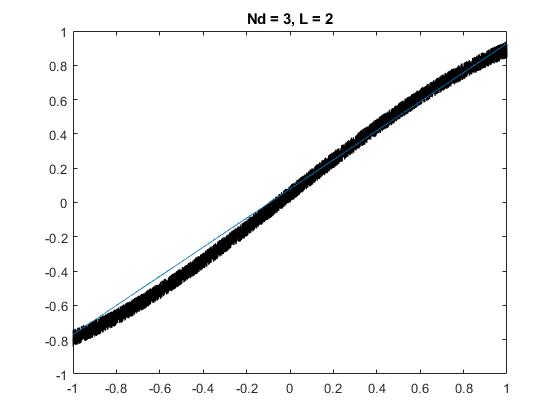


Figure . Nd = 3, L = 2

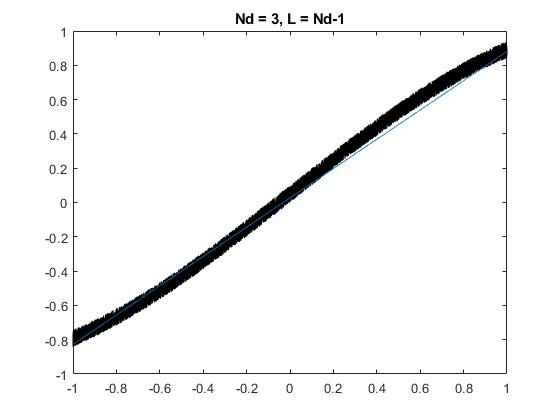


Figure . Nd = 3, L = Nd-1

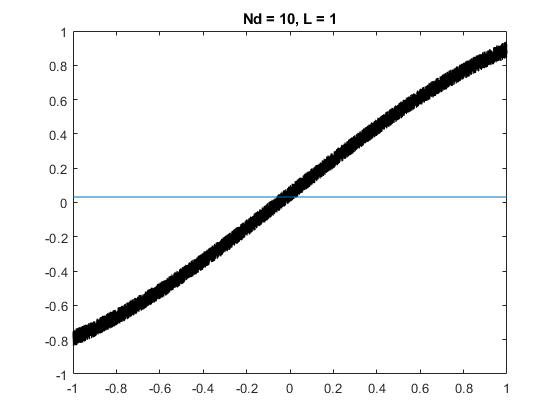


Figure . Nd = 10, l = 1

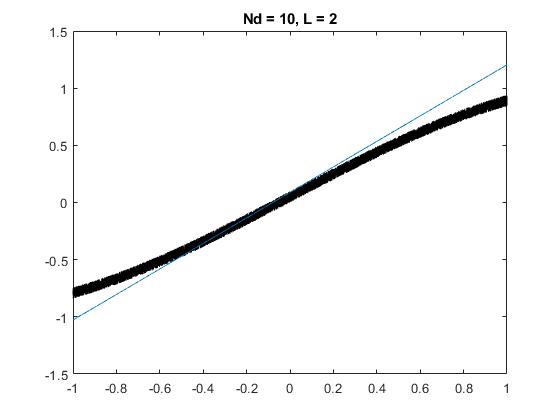


Figure . Nd = 10, l = 2

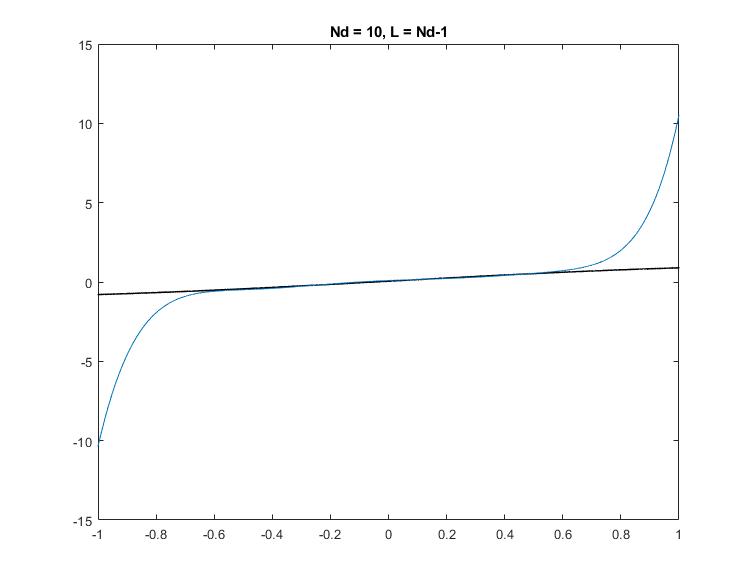


Figure . Nd = 10, l = Nd-1

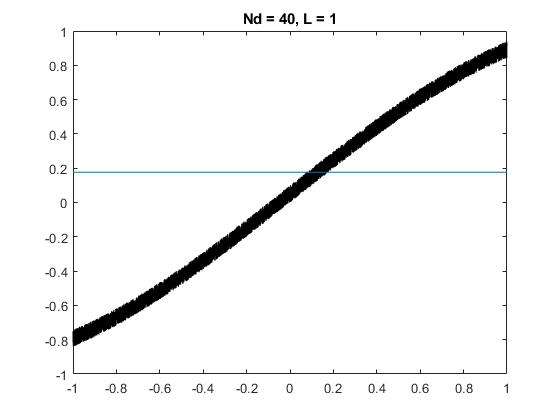


Figure . Nd = 40, l = 1

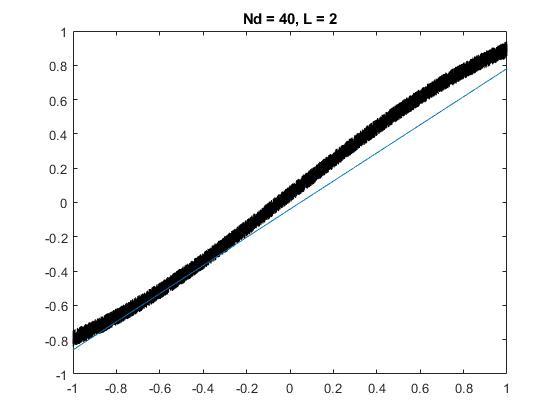


Figure . Nd = 40, l = 2

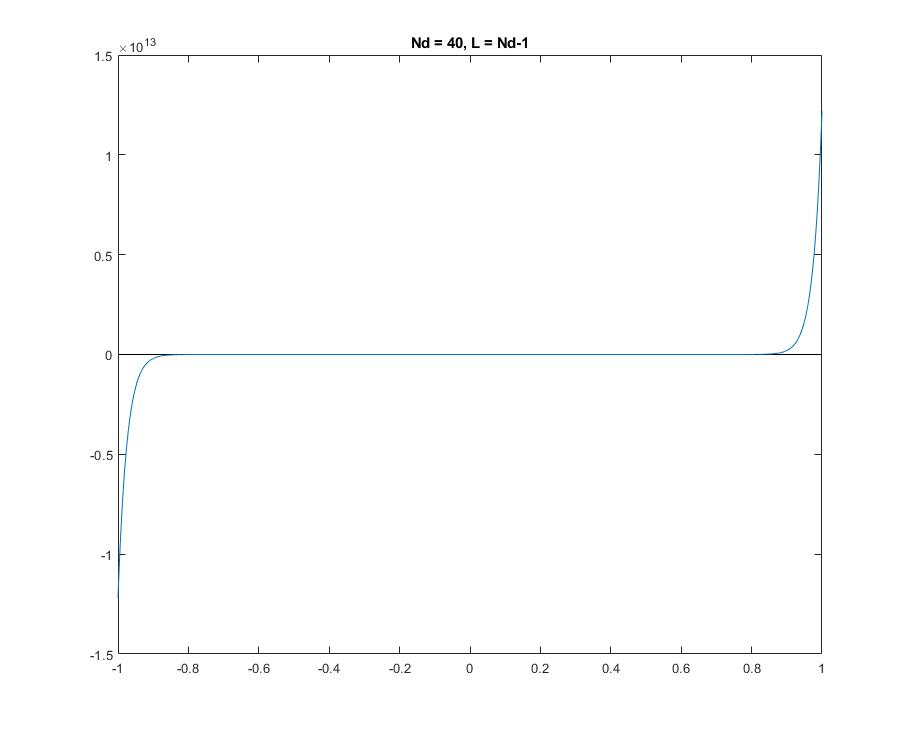


Figure . Nd = 40, l = Nd-1

* 1. Because we want polynomial to fit the data and not the noise. We shouldn’t be using this noise in our relative error calculation. So we should just measure relative error against sin(x).