ECSE 343: Project 1 Report

Deis, Anas
260684605
dept. of ECSE
McGill University
Montreal, Canada
anas.deis@mail.mcgill.ca

Habib, Yehia
260730649
dept. of ECSE
McGill University
Montreal, Canada
yehia.habib@mail.mcgill.ca

Malekian, Reza
260535177
dept. of ECSE
McGill University
Montreal, Canada
reza.malekian@mail.mcgill.ca

I. Design

The strategy used in this project was based on the use of monomial and Legendre basis functions. We used QR decomposition to compute the coefficients as it's been shown to be the most accurate. We later scaled both methods to improve the goodness of fit.

First, when it comes to the monomial method, we first derived the Vandermonde matrix that represents the polynomial or regression matrix *M* with a monomial basis. The system to solve is $Ma = y$, where a is the polynomial coefficients vector and y the output vector of the generated input data *x*. The way we derived the polynomial matrix is such that its columns are the powers of the input vector *x* up to degree *n*. Once we apply QR decomposition on *M*, we get the coefficients vector *a* for the best fit in a least-square sense for the output vector *y*. In order to verify the obtained function, we generated a *x_new* vector with 1,000 linearly spaced data points for [0 2π], and applied the coefficients vector *a* obtained previously. Then, we applied the exponents starting from 0 up to *n* degree on the *x_new* vector values by iterating over its column while increasing the degree such that the interpolant polynomial function $f(x) = a_0 + a_1 x + \cdots + a_p x^p$ is respected, i.e. $f_p = f_{p-1} + a * x^p$. Finally, f was compared with the true function using *x_new* as its input data points.

Second, as for the Legendre method, we used the Legendre basis function to interpolate the data. In the *LegendreMatrix(x,n)* function that we implemented, we constructed a horizontal vector starting from 0 up to *n* degree. We then expanded that vector vertically up until the rows became equal to the number of data points and horizontally up until the columns became equal to the number of degrees starting from 0. This allowed us to get a matrix *N* that can be used as the first input for MATLAB *legendreP(n,x)* function. For example, with $n = 3$ and an *x* vector with 5 data points, we get a 5x4 matrix with each row equal to [0 1 2 3]. Then we expanded the *x* vector horizontally so that each column was equal to $[x_0\ x_1\ x_2\ x_3\ x_4]$. The MATLAB function would then apply the appropriate degree to its data points and return the polynomial matrix. We then applied QR decomposition to obtain the coefficients. In order to verify the obtained function, we generated a *x_new* vector with 1,000 linearly spaced data points for [0 2π], and applied the coefficients obtained previously. Then, we applied the powers on the *x_new* such that the interpolant polynomial function $f(x) = a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_p\phi_p(x)$ is respected, i.e. $f_p = f_{p-1} + a * a_p\phi_p(x)$. Finally, f was compared with the true function using *x_new* as its input data points.

Third, to design the scaled version of our interpolant function developed using the monomial and Legendre basis functions, we had to scale our input data. To be more specific, we scaled the input data x_j lying between the interval [0 2π] to values between -1 and 1, using the following formula:

$$\hat{x}_j = \frac{2}{b-a}\left(x_j - \frac{a+b}{2}\right)$$

The constant *a* represents the lower bound of our interval, which happens to equal 0 in the case of our project, and the constant *b* represents the upper bound of our interval, that is, 2π. Aadditionally, when plotting the scaled version of our interpolant functions, we also made sure to scale our *x* plotting values in order to compensate for our new scaled coefficients and get the correct *y* values for our constructed interpolant functions. The motivation behind the implementation of our scaling method was that the regression matrices used in our Monomial and Legendre methods can be ill-conditioned, which can result in erroneous results for the coefficients a_i*'s*.

II. Experimentation

To test the precision of fit, a visual examination is required as the first step. Beyond that first step, we implemented the root mean squared error (RMSE) and the coefficient of determination (R^2) as numerical indicators for goodness of fit. RMSE is the standard deviation of residuals that indicates the fit standard error, i.e., how close is the observed data to the expected one. Lower values of RMSE indicate better fit. R^2 is the square of the correlation between the observed and expected data. R^2 can take on any value between 0 and 1, with a value closer to 1 indicating that a greater proportion of variance is accounted for by the model. Moreover, we made our MATLAB code compact where 4 graphs are generated with each graph representing one data type that is fully configurable by the following four variables: *deg1-4*, *num_data1-4, scale1-4,* and *legendreBasis1-4*. The first two variable are natural numbers whereas the latter two are Boolean. For instance, if we set *degree1* = 25, *num_data1* = 28, *scale1*= true, and *lendegreBasis1*= false, then the output of case 1 (linearly spaced data) would use the monomial basis function, scaling, a degree of 25 and 28 data points.

A. Monomial Unscaled

In this part of our experimentation, we tried to determine the effect of increasing and decreasing the data points count

and degree on the RMSE and R^2 using a monomial basis function without scaling. There were a couple of things that we observed. Firstly, the values for randomly selected data are naturally unpredictable, and have very high margins resulting in unpredictable numbers. Secondly, Chebyshev provides more stable, and somewhat decent, results that are less prone to massive changes due to slight degree or point count variations, making it a good choice when dealing with high number of data points and/or degrees. Legendre roots data and Chebyshev were found to perform the best using the same degree of 15 while providing somewhat decent results with only 40 data points. The RMSE and R^2 of Chebyshev is 1.6709 and 0.203 respectively performing slightly worse than Legendre roots with 1.6691 and 0.2045 respectively. Thirdly, we achieved the best results out of all data types using linearly spaced data points, but only when the number of data points was very high. The R^2 and RMSE achieved for a degree of 15 and 250 data points were 0.2053 and 1.668 respectively.

Legendre roots comes as a close second to linearly spaced data in terms of fit accuracy. It was found to be the most predictable albeit the results of R^2 and RMSE were extra sensitive to variations in parameters. The strength of Legendre lies in the fact that we were able to achieve accuracy levels comparable to a 250 point linearly spaced data but only using 40 points. That is a significant 84% required point reduction. Therefore, taking the number of data points into account, one can say that the best fit is achieved using Legendre roots data.

While experimenting, we aim to achieve the most balanced and stable function with as few data points as possible as seen in Figure 1. At lower numbers of data points, we experienced a case of curve overfitting as shown in Figure 2. On the other hand, we observe a curve underfitting with a lower degree in Figure 3. The numerical indicators RMSE and R^2 are also shown to be negatively impacted in Figures 2 and 3 compared to Figure 1 as expected.

```
RMSE = 1.6691
R2 = 0.2045
```

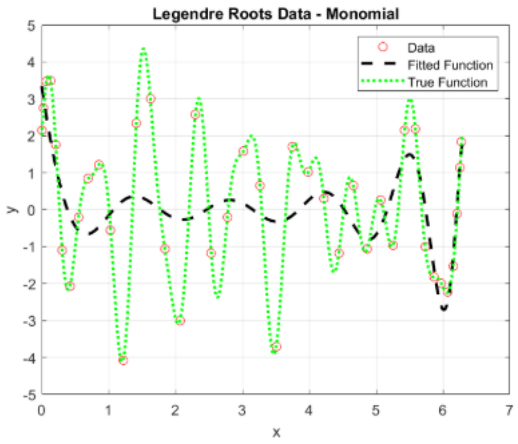

Figure 1. Degree = 15 and no. of data points = 40

```
RMSE = 2.2475
R2 = 0.1505
```

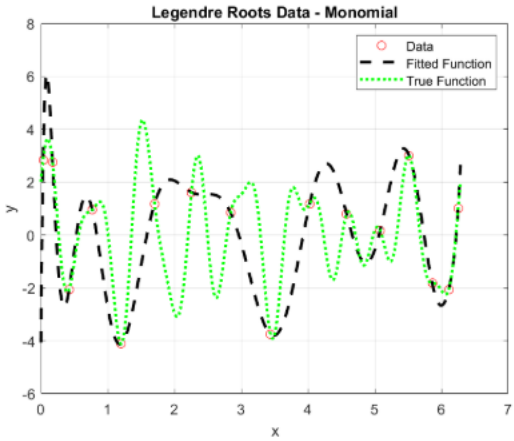

Figure 2. Degree = 15 and no. of data points = 16

```
RMSE = 1.8299
R2 = 0.0462
```

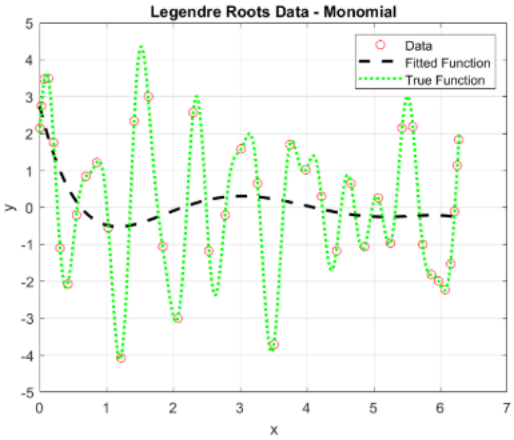

Figure 3. Degree = 5 and no. of data points = 40

B. Monomial Scaled

In part b of monomial, we try to use scaling to achieve better fit. Given that we do not start off by knowing the ideal degree and number of data points, we start by setting the degree to 10 while going through data points in increments of 50 to deduce a general pattern.

We made a few observations. Firstly, randomly generated data provides random values which provide ever-changing unstable results. Consequently, this approach is to be ignored due to unpredictability of random number choices. Secondly, Legendre roots are predictable albeit the goodness of fit is sensitive to change in parameters. Legendre roots with the best fit had a degree of 73 with 144 data points, an RMSE and R^2 of 0.3333 and 0.9683 respectively. Thirdly, Chebyshev provides optimal degree value of 127. By fixing that degree and modifying the data points count to 400, we noticed that best accuracy is maintained between 128 to 250 data points. Given that the experiment requires lowest data point count for best fit, the minimum data point of 128 was chosen. Chebyshev yields a slightly better RMSE and R^2 than Legendre roots, namely 0.3323 and 0.9685 respectively as seen in Figure 6. Also, the curve using Chebyshev was found

to be very predictable and has the smoothest curve. Fourthly, we noted that the linearly spaced data approach provides better fit at high number of data points until a certain threshold point, after which it provides a worse fit. The best suitable degree with the lowest data points possible for the linearly spaced data case was 109 and 406 with an RMSE and an R^2 of 0.3949 and 0.9555 respectively, performing worse than the others.

Finally, we opt to use the Chebyshev technique as it achieved the most accurate fit with the fewest data points. Moreover, we observe an increase in RMSE and decrease in R^2 as demonstrated in Figures 5 and 6 which indicates a worse fit accuracy. We see a curve underfitting when start experimenting with a low degree value as seen in Figure 5. In Figure 6, we see that the fit accuracy when the degree is slightly larger than 127 worsens. As mentioned earlier, 128 data points was the best choice for a degree of 127, which happens to be the lowest number of data points possible for this degree. Therefore, our finalized design parameters are shown in Figure 4.

```
RMSE = 0.3323
R2 = 0.9685
```

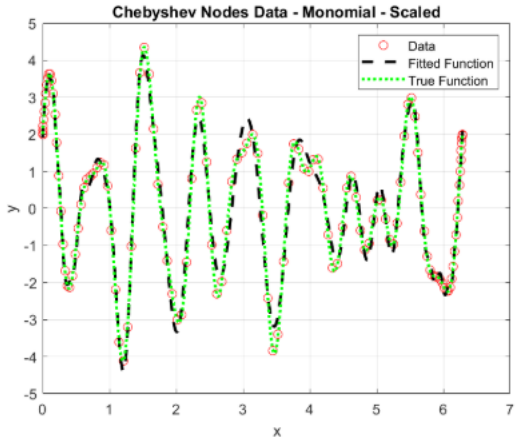

Figure 4. Degree = 127 and no. of data points = 128

```
RMSE = 1.7294
R2 = 0.1456
```

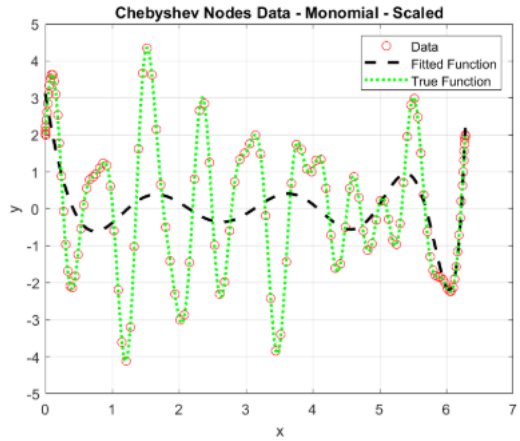

Figure 5. Degree = 10 and no. of data points = 128

```
RMSE = 0.3391
R2 = 0.9672
```

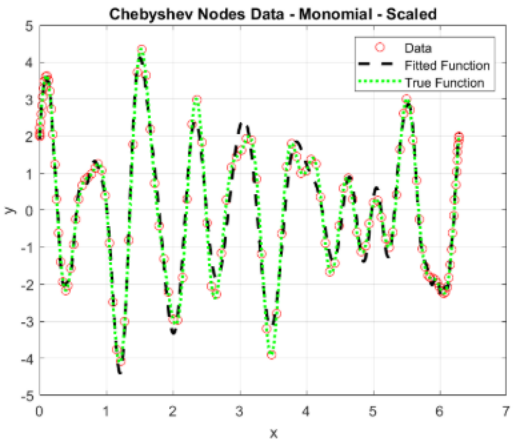

Figure 6. Degree = 130 and no. of data points = 131

C. Legendre Unscaled

We experimented using Legendre basis without scaling and observed that the curves and precision of fit numerical indicators were very unpredictable. For this reason, this part took a lot of time to test as we were more precise in varying the parameters and track changes. Therefore, there's more uncertainty in determining the best fit as the condition number is always very high with very unpredictable results as opposed to monomial.

The randomly sampled data points generate different results on each run, so we did not experiment with it as was the case with the other sections. As for the other three data types, we found the most suitable degree to be 11 for all of them with very similar fit accuracies. The RMSE and R^2 values were 1.7023 and 0.1733 for 83 linearly spaced data points, 1.7082 and 0.1674 for 52 Chebyshev nodes data points, and 1.7041 and 0.1711 for 30 Legendre roots data points. The fit accuracy of Legendre roots data comes as a close second to that of linearly spaced data with a negligible difference between the two. However, we achieve the accuracy of Legendre roots using 64% less data points than linearly spaced data. Therefore, the best fit using the Legendre basis without scaling is achieved while using Legendre roots as seen in Figure 7. During testing, we observed a curve underfitting with a lower degree as seen in Figure 8 and a case of overfitting with lower data points as seen in Figure 9. The RMSE and R^2 numerical indicator are negatively impacted when the curves are overfit or underfit as the fit accuracy is significantly worse for Figure 8 and 9 than Figure 7.

```
RMSE = 1.7041
R2 = 0.1711
```

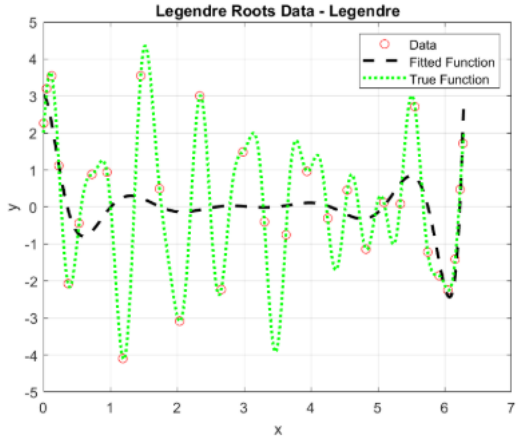

Figure 7. Degree = 11 and no. of data points = 30

```
RMSE = 1.8285
R2 = 0.0475
```

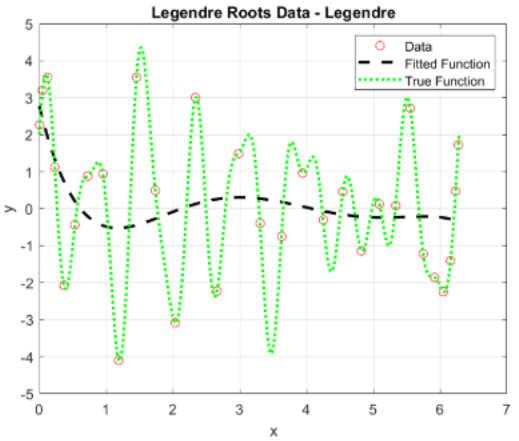

Figure 8. Degree = 5 and no. of data points = 30

```
RMSE = 2.1092
R2 = 0.0520
```

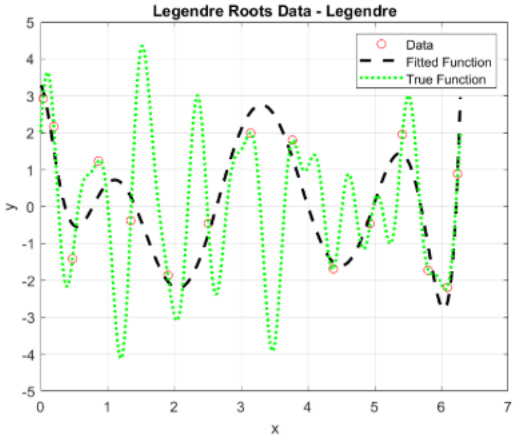

Figure 9. Degree = 11 and no. of data points = 15

D. Legendre Scaled

In part b of using Legendre basis, we tried to use scaling to achieve better fit. We stopped experimenting using randomly sampled data as it yielded random results, as expected. We observed that the R^2 reaches the ideal value of 1 more often than RMSE reaching an ideal value of 0, probably due to rounding errors and other minor calculation factors. Therefore, the negligible difference in RMSE is not to be put in the spotlight. In fact, all the best fits for each of the three data types excluding randomly sampled data have an R^2 of 1. However, we needed to minimize the number of data points even if the fit accuracy is the same. We also observed that the number of data points tended to be the lowest possible for the chosen degree, particularly for Legendre roots and Chebyshev nodes.

Starting with Legendre roots, we suspected the optimal degree to be between 45 and 50 while experimenting. Eventually, we narrowed it down to 49 with only 50 data points which is the lowest possible number for a degree of 49. The RMSE of Legendre roots is 0.0127. Then, we experimented with Chebyshev and achieved great fit accuracy while using a degree between 50 and 55 that is then narrowed down to 51 using 52 data points. The RMSE of Chebyshev nodes is 0.004, slightly better than Legendre roots but the difference can be considered as negligible. Finally, the linearly spaced data performed the best using a degree of 73 and 85 data points. The RMSE of linearly spaced data is 0.0015, better than the other two data types, but still can be considered as negligible.

We opt for Legendre roots as the best fit using the Legendre basis function when scaled as seen in Figure 10. We chose Legendre roots as it provides the lowest number of data points considering all data types have great fit accuracy with a negligible difference between them. A case of curve underfitting with worse fit accuracy can be observed in Figure 11 when we tested using a lower degree.

```
RMSE = 0.0127
R2 = 1.0000
```

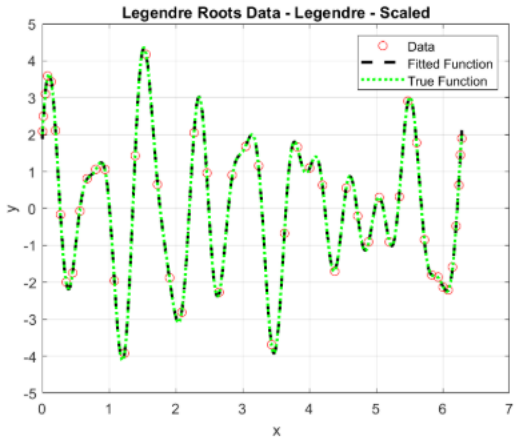

Figure 10. Degree = 49 and no. of data points = 50

```
RMSE = 1.7264
R2 = 0.1486
```

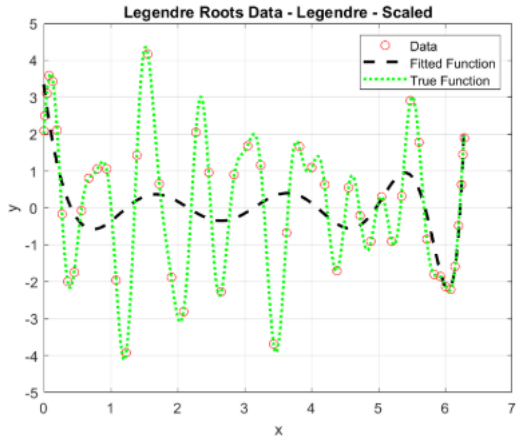

Figure 11. Degree = 10 and no. of data points = 50

E. Best fit

In this section, we compare the best fit accuracy between monomial and Legendre basis functions and discuss the impact of scaling on the condition number of regression matrices. In other words, we pick our overall best fits when scaled and unscaled while considering all parameters.

Firstly, observing the effect of scaling on the condition number of the regression matrices, we see that in all cases and in all scenarios, scaling has a positive impact on the condition number of the regression matrices. More specifically, when we compare the scaled and unscaled versions of a specific fitted polynomial that share the same basis function, degree, number of data points, and input data types, we observe that for the scaled version the condition number of the regression matrices decreases significantly. This behavior serves as a huge advantage as it helps improve the performance and produce more accurate results. For example, when constructing our unscaled monomial using 250 linearly spaced data points and degree of 15, we get a condition number of 1.2321e+29. However, the condition number decreases to 4.9689e+10 when scaling. Similarly, when constructing our unscaled Legendre polynomial using 30 Legendre roots data points and degree of 11, we get a condition number of 1.1301e+25. However, using scaling, the condition number decreases to only 30.6478! It is also important to note that the impact of scaling on the condition number of the regression matrices is more significant when constructing Legendre polynomials as opposed to monomial polynomials.

Secondly, comparing unscaled monomial to Legendre, we see that for unscaled monomial, our best fit was obtained using 40 Legendre roots data points and a polynomial degree of 15. These parameters gave us an RMSE value of 1.6691 and an R^2 value of 0.2045. On the contrary, for unscaled Legendre our best fit was obtained using 30 Legendre roots data points and a polynomial degree of 11. These parameters gave us an RMSE value of 1.7041 and an R^2 value of 0.1711. Therefore, we observe the best fit of unscaled monomial to be slightly more accurate. However, it required a greater number of data points, i.e. 40, in comparison to unscaled Legendre that required only 30. Given the number of data points used in unscaled monomial is 33% greater than that used in unscaled Legendre that resulted in only 2.05% decrease in RMSE value and a relatively small increase in the value of R^2, we conclude that we would go with unscaled Legendre as the best fit overall without scaling as shown in Figure 7. This is mainly due to similar precision and accuracy levels to monomial whilst using only 75% of its data points.

Thirdly, comparing scaled monomial to Legendre, we see that our best fit for scaled monomial was obtained when using 128 Chebyshev data points and a polynomial degree of 127. These parameters gave us an RMSE value of 0.3323 and an R^2 value of 0.9685. On the contrary, for scaled Legendre our best fit was obtained using 50 Legendre roots data points and a polynomial degree of 49. These parameters gave us an RMSE value of 0.0127 and an R^2 value of 1.000. Therefore, we observe that the best fit of scaled Legendre achieved a more accurate fit. Additionally, it required a smaller number of data points, i.e. 50, in comparison to scaled monomial which required 128. Scaled monomial required 2.56 times the number of data points required by Legendre. Moreover, it is also important to note that using the abovementioned parameters, scaled Legendre gave us an almost perfect fit of the true function, rendering it the best and most accurate constructed polynomial overall in the entire project. Hence, we conclude that we would go with scaled Legendre as a final decision as seen in Figure 10.

In short, our final design parameters without scaling are 30 Legendre roots data points and a polynomial degree of 11 using Legendre basis function as shown in Figure 7. As with scaling, we also go with Legendre basis function, but with 50 Legendre roots data points and a polynomial degree of 49 as shown in Figure 10. The scaled version obviously offers the better fit due to a significantly lower condition number and higher fit accuracy and precision.