



PHYS 514 - COMPUTATIONAL PHYSICS

Solution Set 4

Author:

Volkan Aydıngül (Id: 0075359)

Date: November 13, 2020

Contents

1	Problem XI	2
1.1	General Looking of the Data	2
1.2	Linear Fit on the Data	2
1.3	Nonlinear Fit on the Data	3
1.4	Error Estimation	5
2	Problem XII	8
2.1	Exact Polynomial Fit with Varying Number of Equidistant Nodes	8
2.2	Linear and Cubic Spline Interpolation	9
2.3	Exact Polynomial Fit with Chebyshev Nodes	12
2.4	Polynomial Fit when Number of Nodes is not Equal to Degree of Polynomial	14

1 Problem XI

1.1 General Looking of the Data

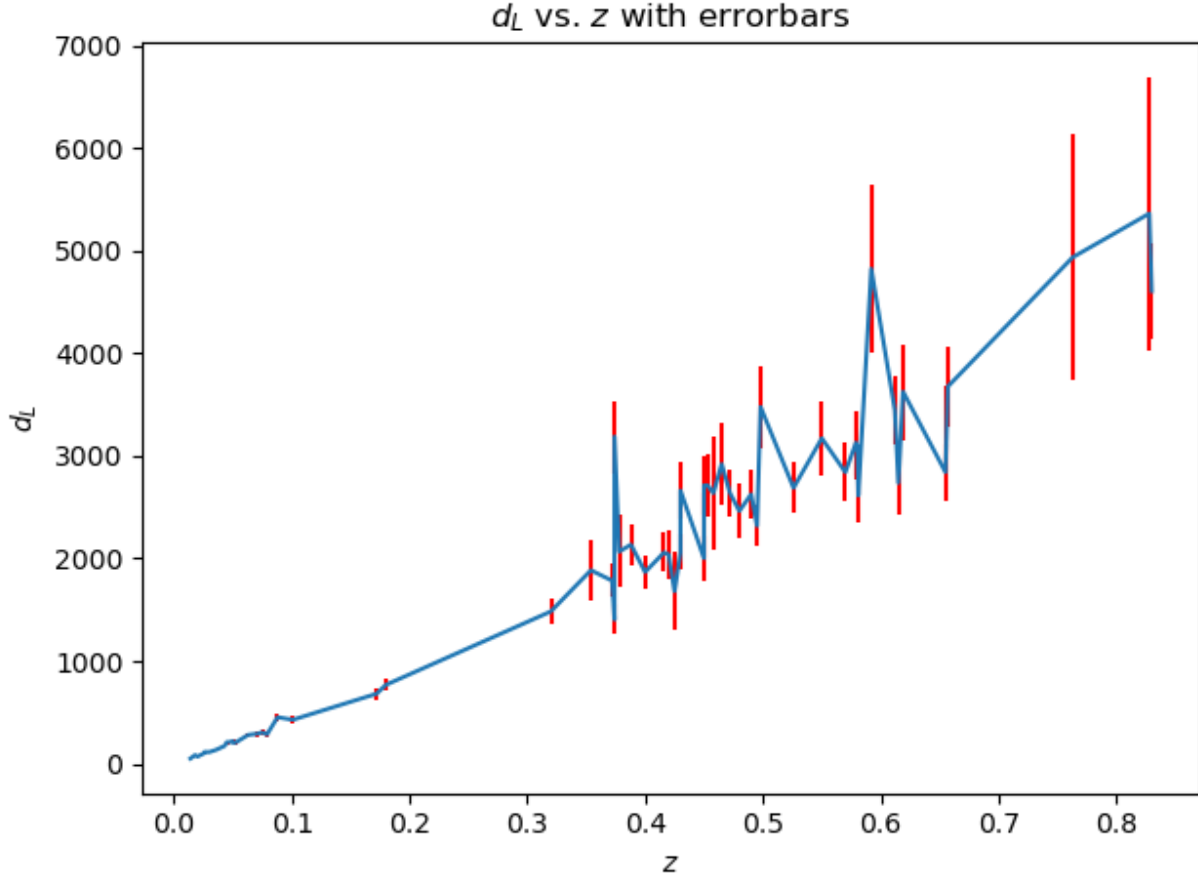


Figure 1: Errorbar Plot of the Data

In Figure 1, the general trend of the data can be observed and the errors in the measurement can be investigated.

1.2 Linear Fit on the Data

In Figure 2, the linear fitting process executed on the data can be observed. The fitting scheme is following:

$$d_L = az + b$$

where z represents the independent variable (redshift parameter), d_L represents the dependent variable (luminosity distance of the source). The purpose was to obtain a and b for the condition of $z < 0.1$.

In the problem set, also, the cosmological term called *Hubble length* is asked. It is known that:

$$d_L = d_H z$$

where d_H is the *Hubble length*.

According to the above notion, one can easily expect that, as a result of linear fitting, a should result in approximately d_H and b should be result in approximately 0. From Figure 2, it can be displayed that:

$$d_H = a = 4508.135$$

and b is relatively close to 0.

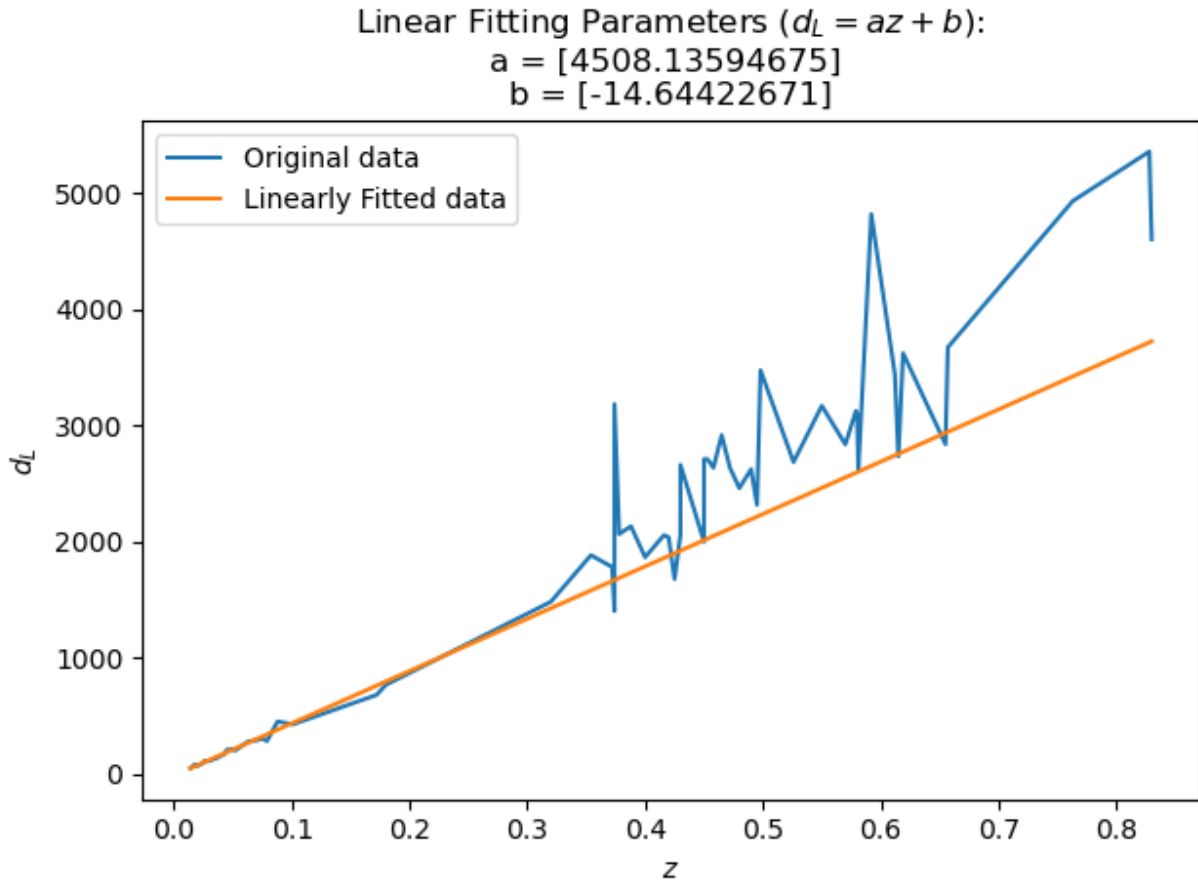


Figure 2: Linear Fit on the Data

1.3 Nonlinear Fit on the Data

In many cases, it is obvious that the linear fit does not result in fairly satisfying results. In this case, nonlinear fitting is the option that one should choose. In this context, the objective

function is following:

$$d_L = (1 + z)d_H \int_0^z \frac{1}{\sqrt{\Omega_M(1+x)^3 + (1 - \Omega_M)}} dx$$

which can be also represented as:

$$f(z, d_H, \Omega_M) = d_L = (1 + z)d_H I(z, \Omega_M)$$

where z are independent variable and d_L is the dependent variable. Also, it should be noted that Ω_M is normalized form of matter density in the universe. To solve nonlinear fitting problem, there exists several methods. One method can be the transformation of the problem into nonlinear root finding problem. The procedure will be explained in following statements.

First, an objective function should be defined, which can be observed below:

$$J = \sum_{k=1}^n \left((f_k(z, d_H, \Omega_M) - \hat{d}_{L_k})^2 \right)$$

As in the case of all optimization problem, the purpose is to minimize above equation to get optimal parameters. However, the important point is that the minimization should be with respect to d_H and Ω_M , since the main objective is to find them. Therefore, the derivative with respect to these variables should be computed, which can be observed below:

$$\frac{\partial J}{\partial d_H} = \sum 2((1 + z)d_H I(z, \Omega_M) - d_L)(1 + z)I(z, \Omega_M) = 0$$

$$\frac{\partial J}{\partial \Omega_M} = \sum 2((1 + z)d_H I(z, \Omega_M) - d_L)(1 + z)\frac{\partial I(z, \Omega_M)}{\partial \Omega_M} = 0$$

It should be noted that $\frac{\partial I(z, \Omega_M)}{\partial \Omega_M}$ can be easily found via applying *Leibniz's Integration Rule*, which constructs the following form:

$$\frac{\partial I(z, \Omega_M)}{\partial \Omega_M} = \int_0^z \frac{\partial \frac{1}{\sqrt{\Omega_M(1+x)^3 + (1 - \Omega_M)}}}{\partial \Omega_M} dx$$

Finally, above nonlinear equation set can be solved via using nonlinear root finding methods. In Figure 3, the results can be displayed.

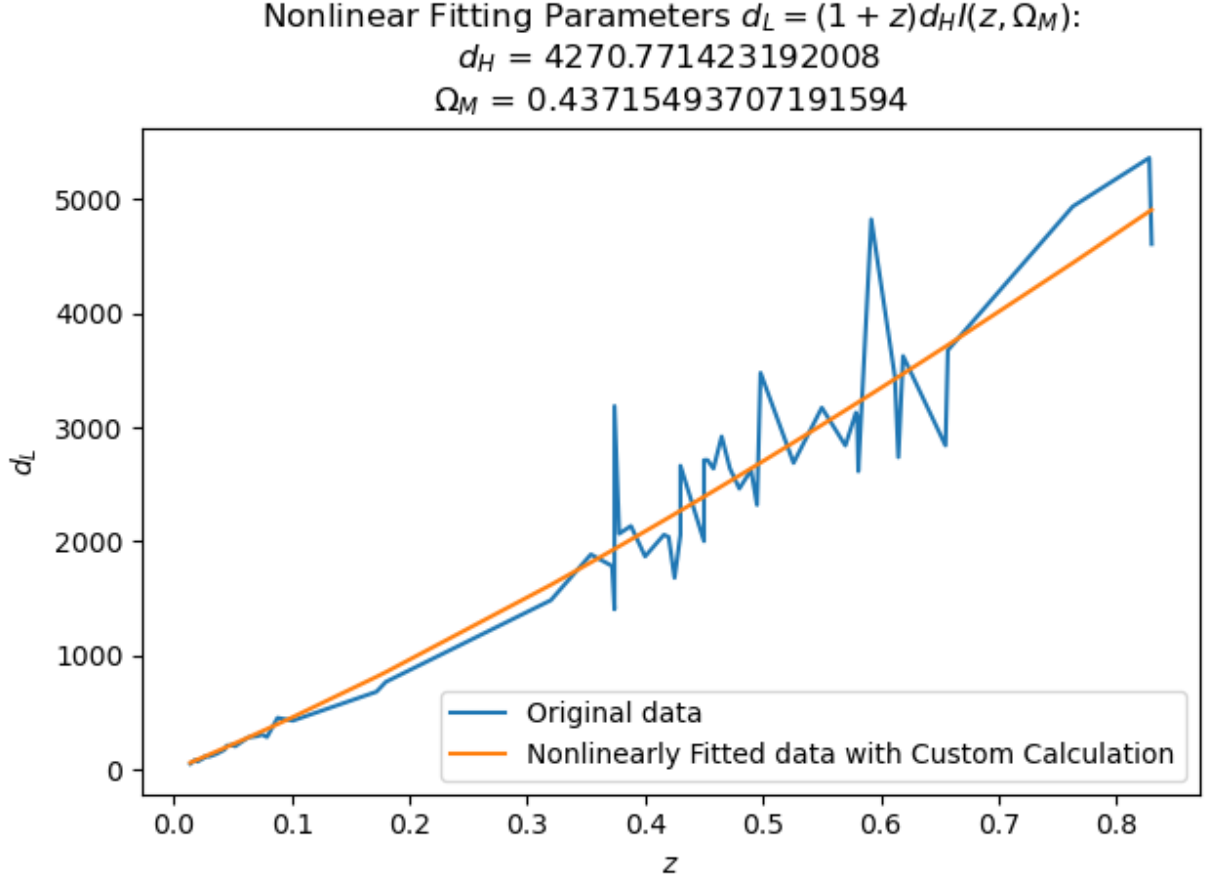


Figure 3: Nonlinear Fit on the Data with Custom Method

As a first guess of d_H , the result found from linear fit is used, and, for the first guess of Ω_M , the value of 0.5 is used. As a result, the below outcomes are obtained.

$$d_H = 4270.771$$

$$\Omega_M = 0.4371$$

1.4 Error Estimation

For the error estimation, the separate methods are followed. Firstly, usage of the *Gaussian* distributions of the d_L and its measurement errors are incorporated as explained in the *Problem Set 4*. Secondly, the whole nonlinear curve fitting process is managed by the *SciPy* built-in function, and its error estimations are observed.

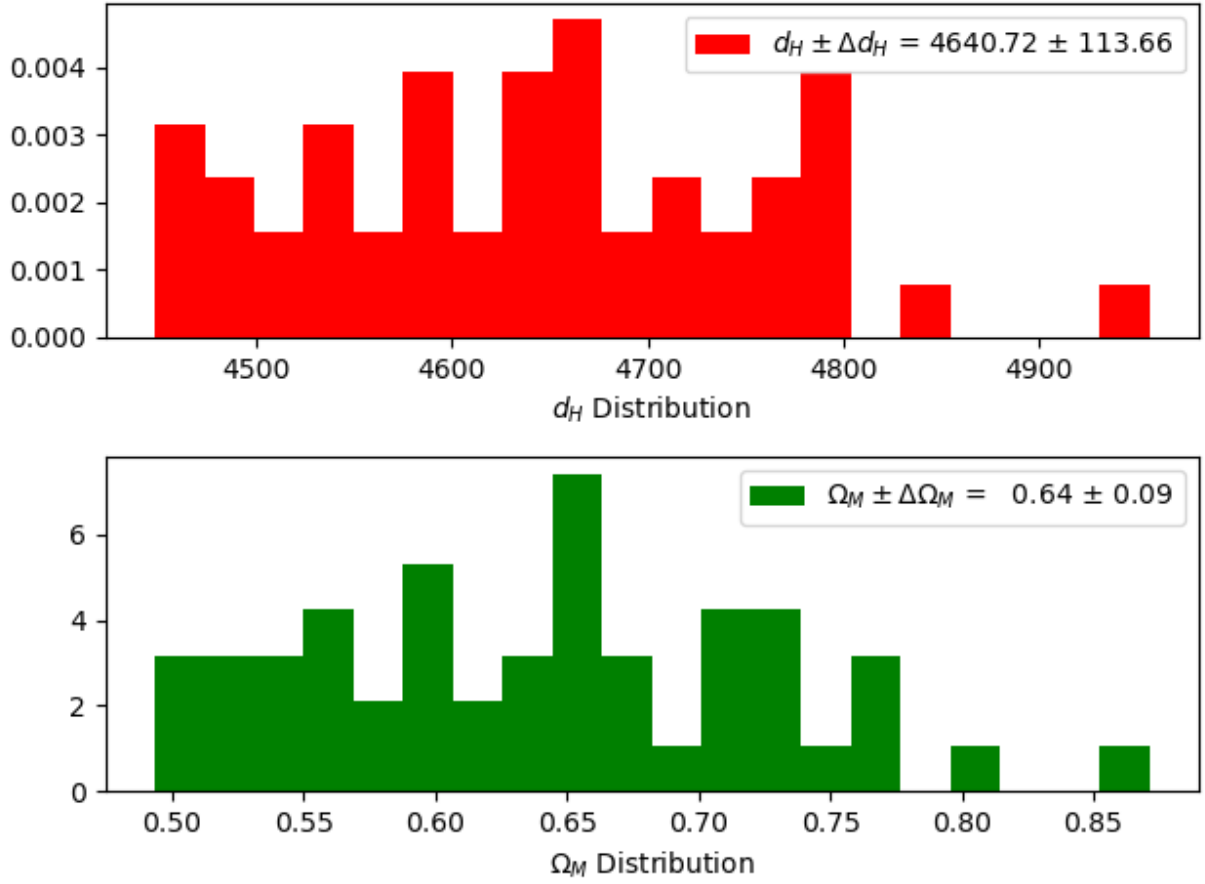


Figure 4: Error Estimation using Gaussian Distributions

Observing Figure 4, and Figure 6, it can be concluded that the custom error estimation does not result in with a good accruacy as it can be encountered in the SciPy function case. However, it should be also noted that the custom error estimation gives a smaller uncertainty bound, which can be interpreted favourably. On the other hand, from the Figure 3, and Figure 5, it should be concluded that the both models have a quiet good approximation performance.

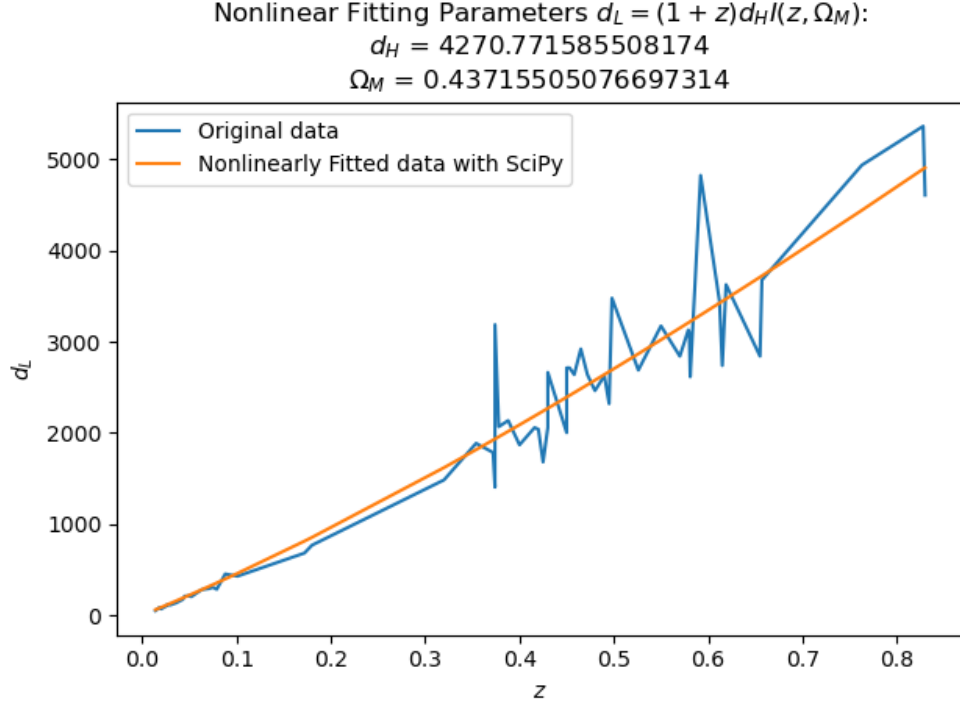


Figure 5: Nonlinear Fit on the Data with SciPy Built-in Function

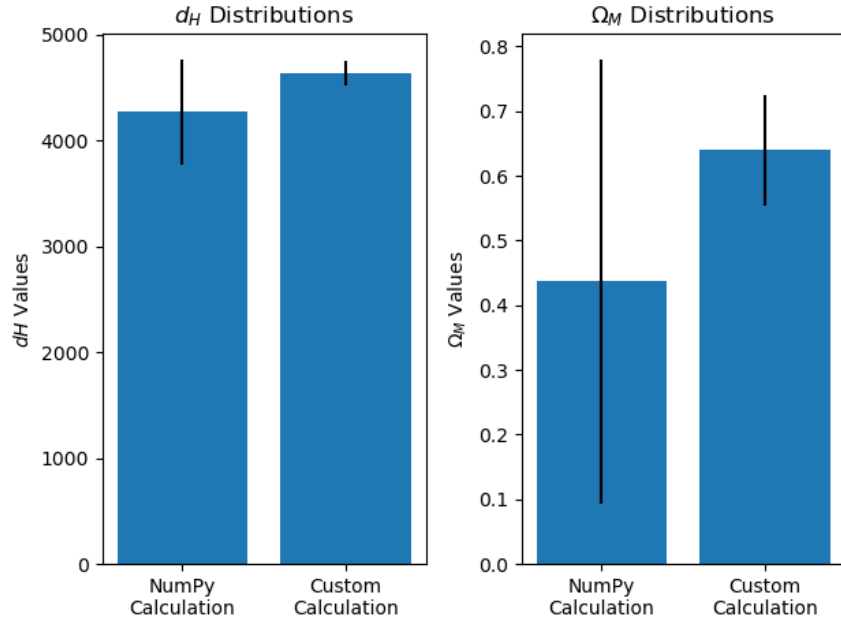


Figure 6: Error Estimations using SciPy Built-in Method and its Comparison with Custom Method

2 Problem XII

In this problem, the subject function is the following:

$$f(x) = \frac{1}{1 + 25x^2}$$

Above function was being tried to approximate via using various methods, and the performance of the approximation was calculated via following metric called *Frobenius norm*.

$$E(n) = \left[\frac{1}{2} \int_{-1}^1 |R(x) - R_n(x)|^2 \right]$$

2.1 Exact Polynomial Fit with Varying Number of Equidistant Nodes

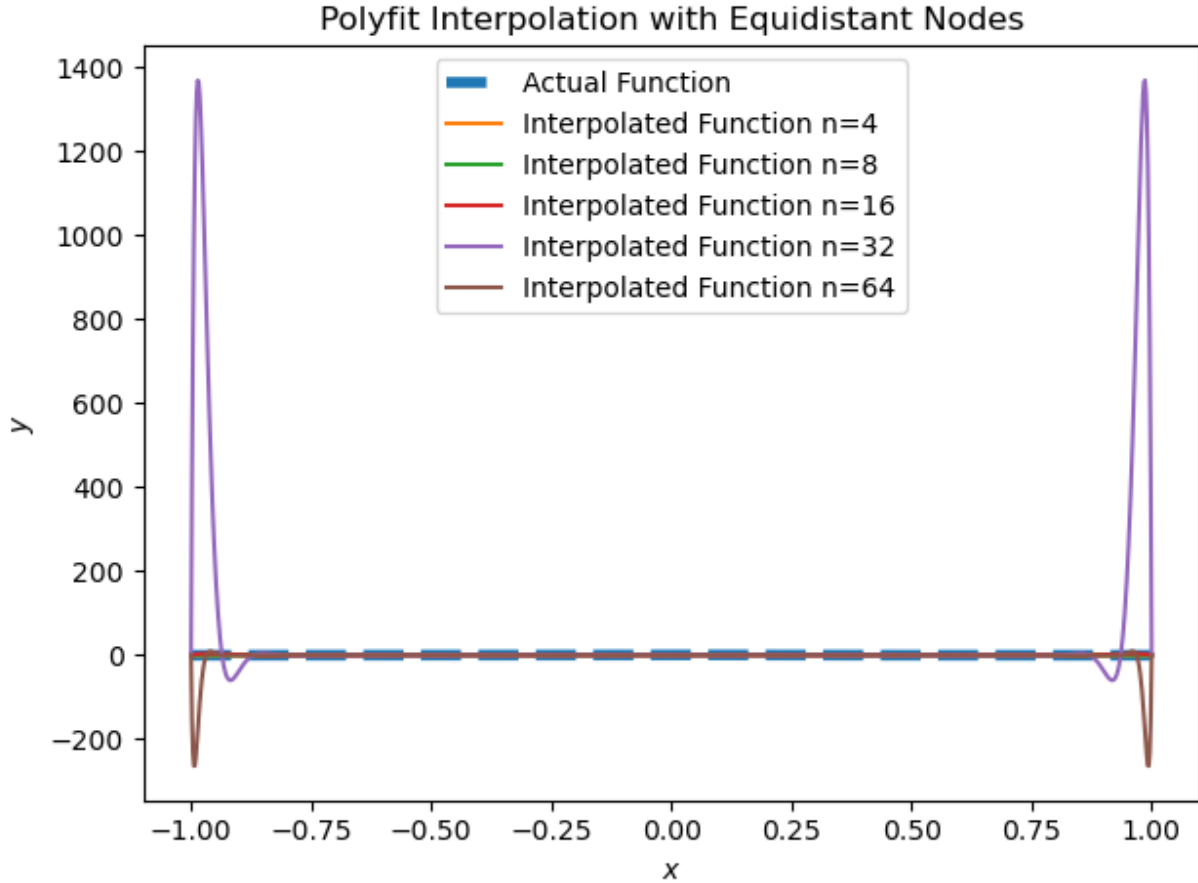


Figure 7: Exact Polynomial Fit with Varying Number of Equidistant Nodes

From Figure 7 and Figure 8 it can be observed that the increase in the number of nodes does not result in the increase of the quality. It can be explained by the *Runge's phenomenon*,

which basically tells that the increase in the number of equidistant nodes does not necessarily result in the increase of accuracy. From Figure 8, it can be seen that the Frobenius norm tends to decrease, however, Figure 7 clearly shows that it is not an logical approximation.

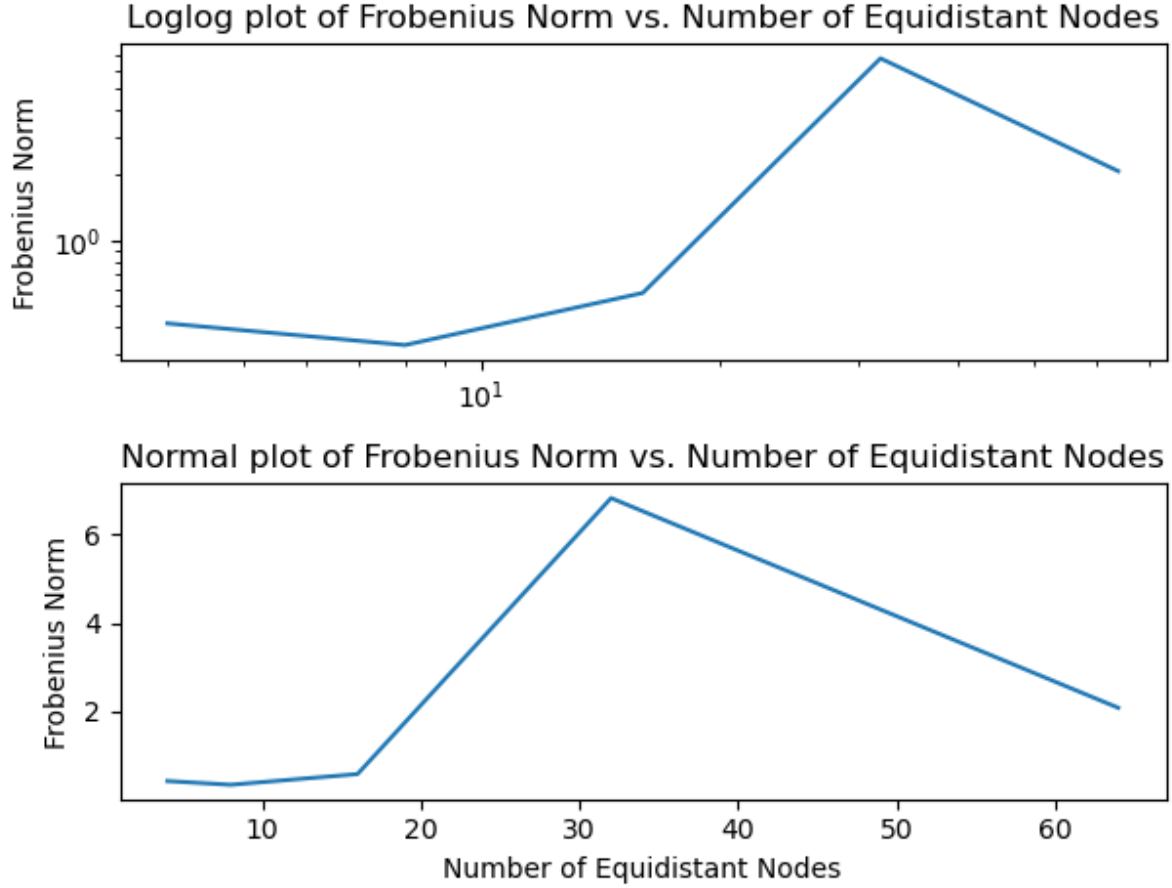


Figure 8: Frobenius Norm of Exact Polynomial Fit with Varying Number of Equidistant Nodes

2.2 Linear and Cubic Spline Interpolation

In this section, the spline interpolation will be discussed. From Figure 9 and Figure 10, even linear spline interpolation does result in good approximation. The basic result behind that might be the fact that it is most basic thing one can come up with to approximate a function, like connecting the dots. However, the downside is the fact that this approximation yields a huge number of sharp points due to the piecewise line connections. These sharp points may result in undesirable derivative expression at there.

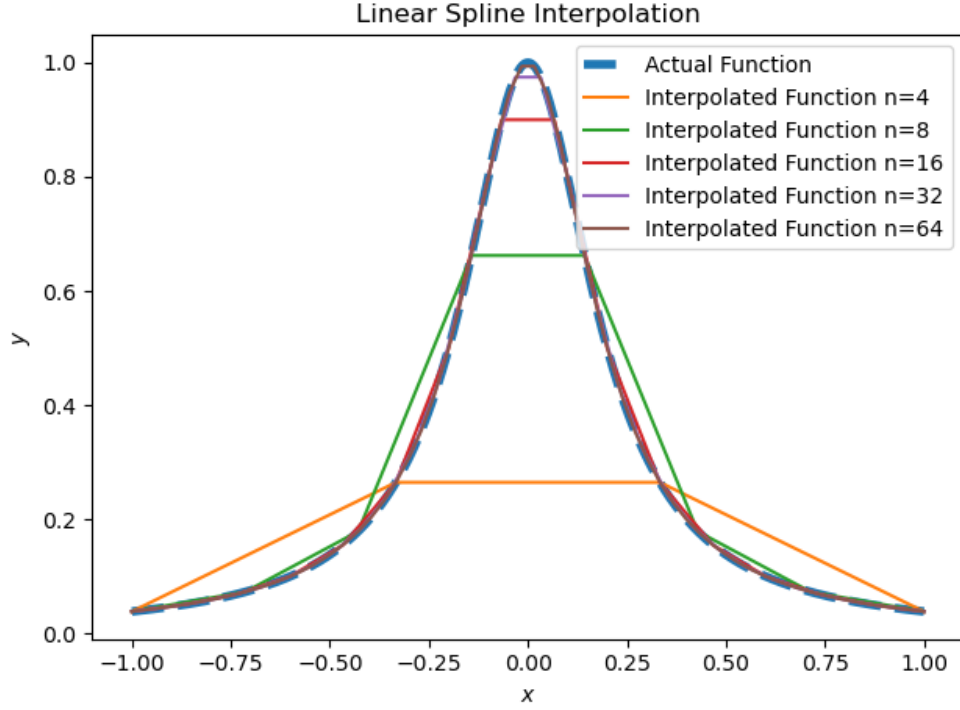


Figure 9: Linear Spline Interpolation on the Data

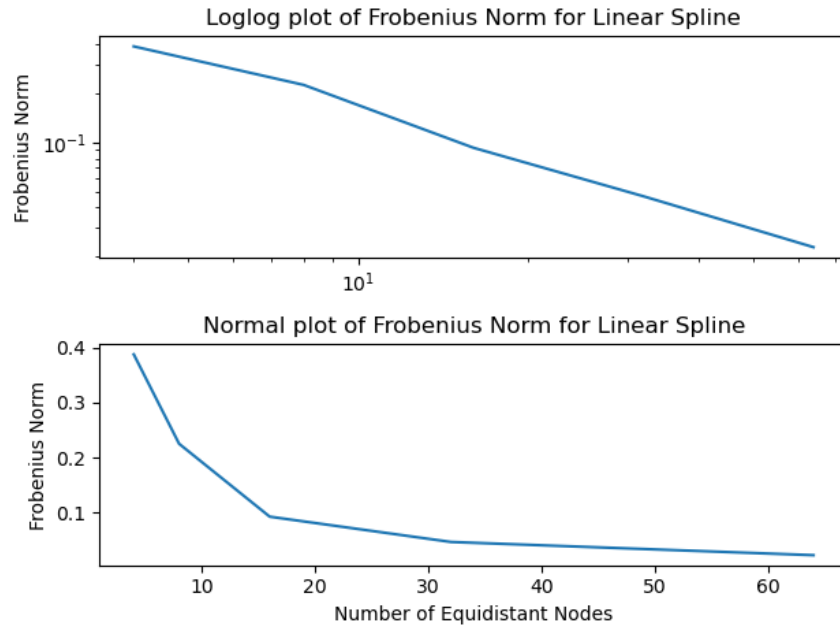


Figure 10: Frobenius Norm of the Linear Spline Interpolation on the Data

In Figure 11 and Figure 12, the cubic spline interpolation results can be investigated. In linear spline interpolation, the sharpness of the result was discussed. In this sense, the

cubic spline interpolation can be thought as a solution to that problem. The cubic spline interpolation successfully manages to connect nodes without losing the continuity.

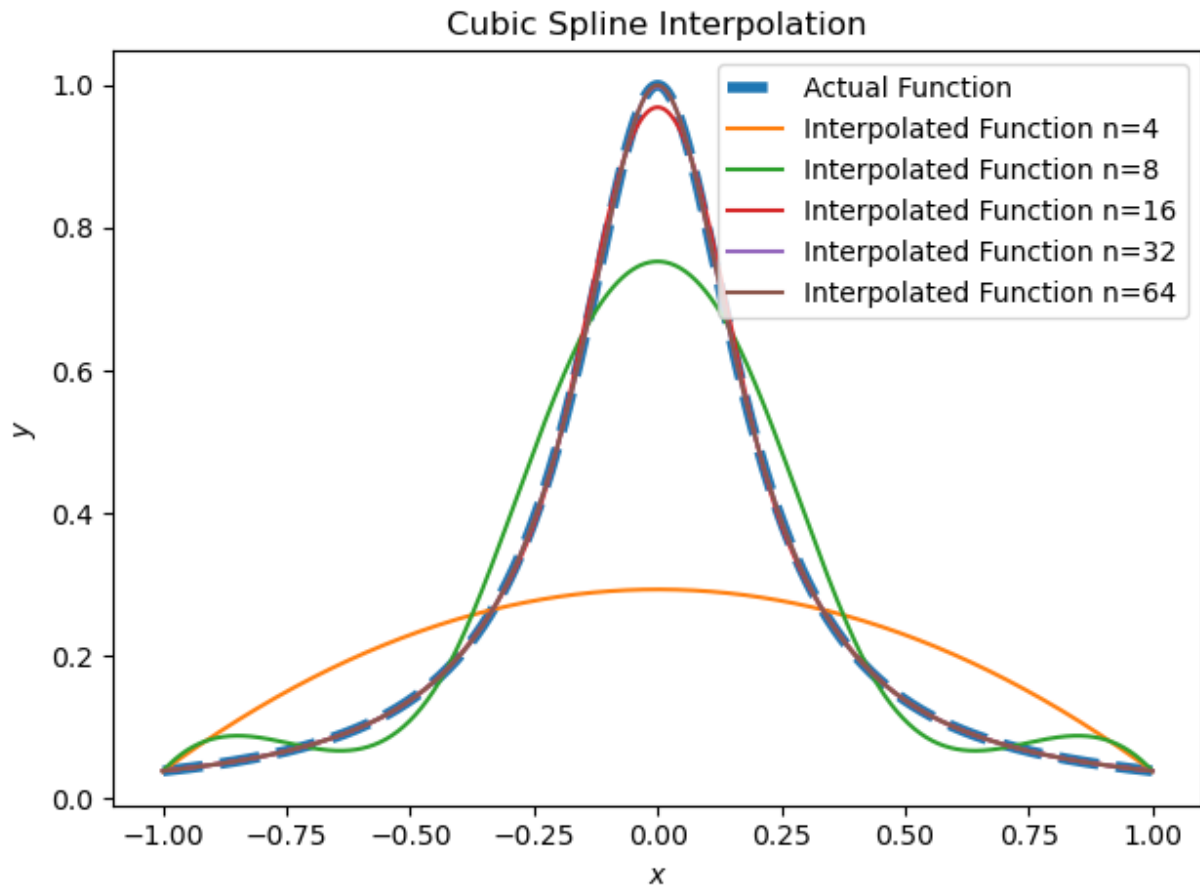


Figure 11: Cubic Spline Interpolation on the Data

Therefore, in the the context of differentiability, the more desired results are obtained.

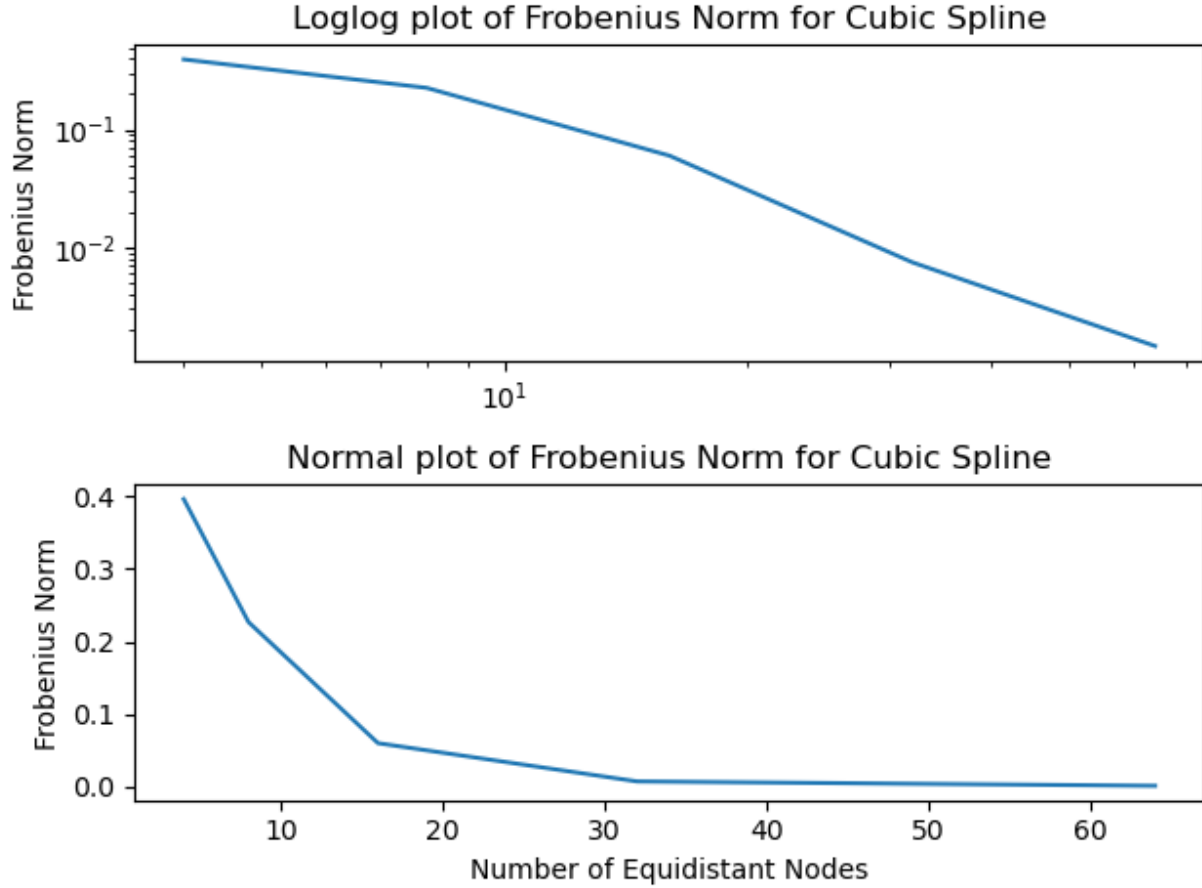


Figure 12: Frobenius Norm of the Cubic Spline Interpolation on the Data

2.3 Exact Polynomial Fit with Chebyshev Nodes

In Section 2.1, it was discussed that the increase in the number of Equidistant nodes does not result in the performance increase. However, there is a way to fix it. The usage of the not-equidistant nodes can somehow yield more efficient results. The one convention as a not-equidistant node distribution is the usage of the *Chebyshev nodes*. Chebyshev nodes can be defined as a projection of equidistant points on a unit circle. From Figure 13, one can observe that the usage of the *Chebyshev nodes* immediately brings more satisfying approximations.

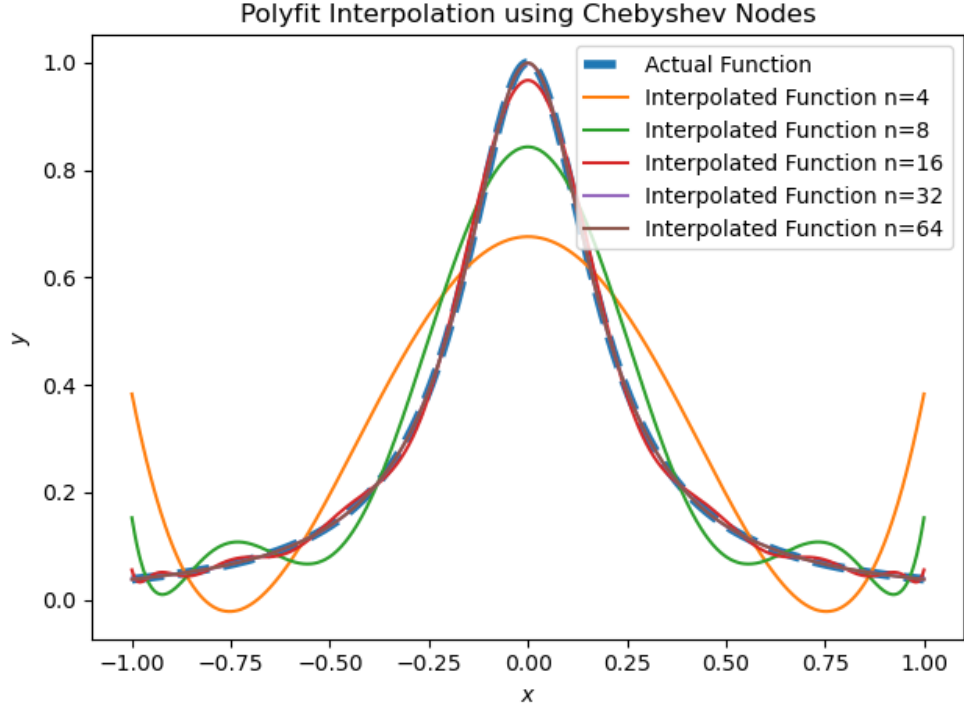


Figure 13: Exact Polynomial Fit with Chebyshev Nodes

Moreover, having observed Figure 14, the decrease in the Frobenius norm can be easily detected.

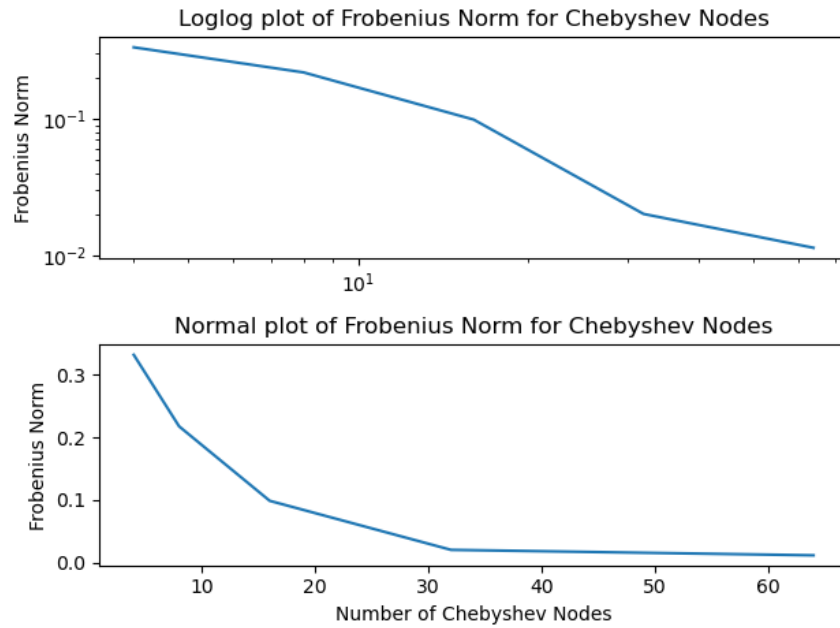


Figure 14: Frobenius Norm of the Exact Polynomial Fit with Chebyshev Nodes

2.4 Polynomial Fit when Number of Nodes is not Equal to Degree of Polynomial

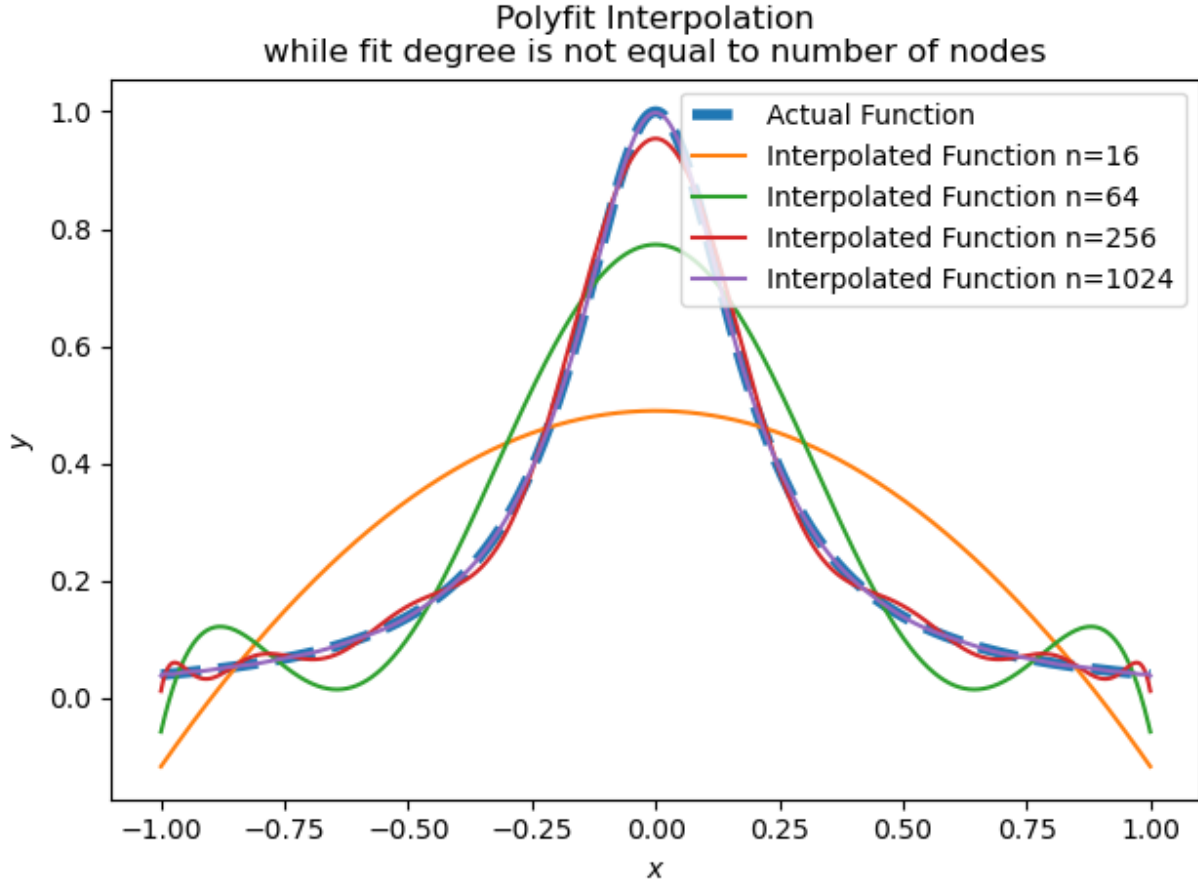


Figure 15: Polynomial Fit when Number of Nodes is not Equal to Degree of Polynomial

The final observation is the relation between the number of nodes and degree of fit. So far, for polynomial fitting, the number of nodes were equal to the degree of fitting. However, one should also observe the effect of the nonequality between them. In the all curves located on the Figure 15, the following relation was followed:

$$\text{Degree of Fitting} = \sqrt{n} - 1$$

According to the Figure 15, one can easily deduct that although the number of nodes is climactic, the fitted lines manages to approximate well. This finding also can be verified by investigating the Figure 16, which clearly shows that although the number of nodes increases, the Frobenius norms insist on decreasing.

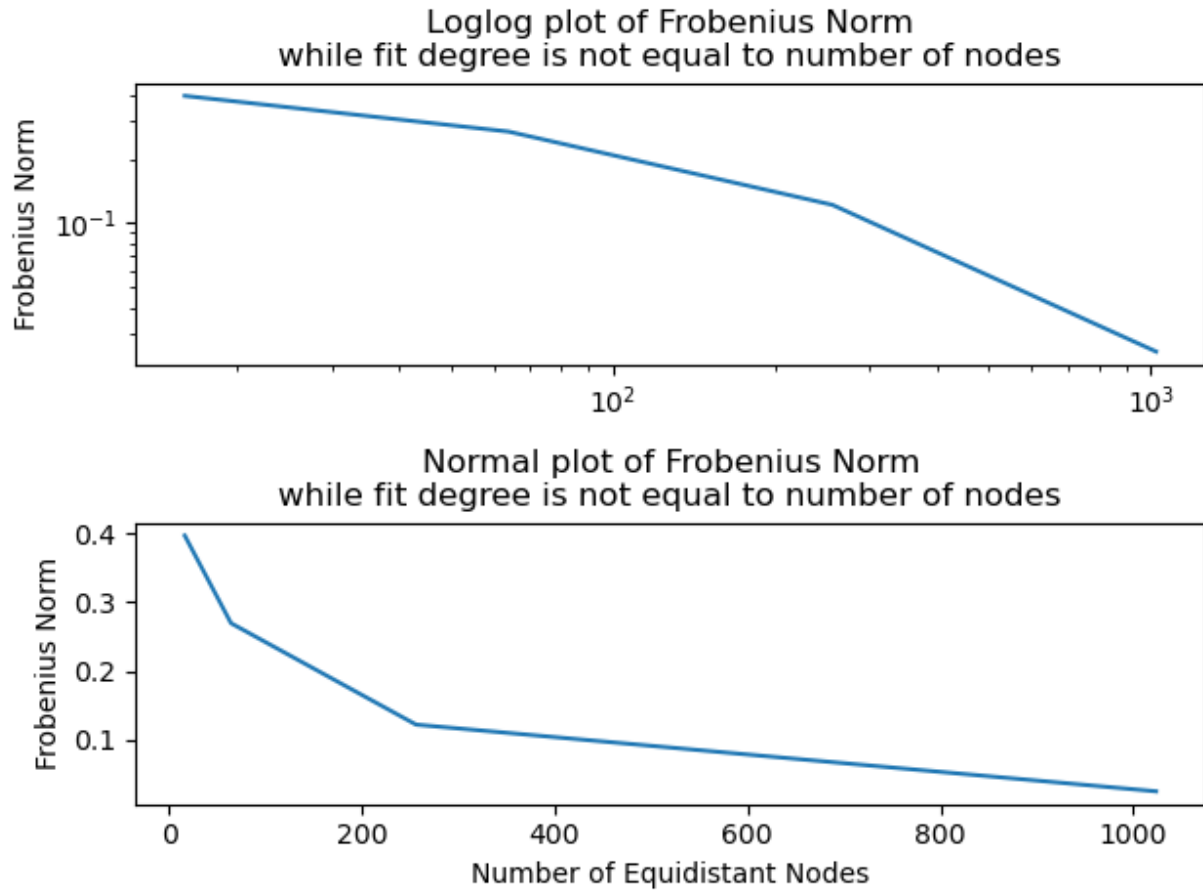


Figure 16: Frobenius Norm of the Polynomial Fit when Number of Nodes is not Equal to Degree of Polynomial