

Numerical Interpolation with Polynomials Approximation and Curve Fitting, Focus the MATLAB

Carlos Figueroa¹, Raul Riera², German Campoy²

¹Industrial Engineering Department. ² DIFUS University of Sonora, Mexico

Abstract: In this work we show classical and known forms to carry out numerical interpolation and curve fitting. Each method is briefly explained and examples from Lagrange, Newton, Hermite, osculating polynomial, and Padé approximation are presented. In the least-squares line, data linearization method of exponential and power function exercises are solved. Our idea is to show the advantages of using MATLAB in the study of numerical analyses and to verify the minimal effort required in using this program to save time in making mathematical demonstrations and developments essential in the obtaining of each method.

Keywords: Polynomial approximation, interpolation, regression.

I. Introduction

To approximate functions, another function that approximates the original is determined. A function is built from a group of known values or is adjusted to a functional dependency; classical techniques are the interpolation with polynomials and least-squares line.

Numerical interpolation consists in estimating values from known data that are considered accurate, while the fit curve fitting considers that data are the result of an experiment and can have a type of error. In both cases the idea is to build a polynomial function that matches the known data so that we have an analytical expression that allows for manipulations.

The polynomial interpolation is justified by the following theorem of Weierstrass given $\varepsilon > 0$ there is a $P(x)$ polynomial, such that

$$|f(x) - P(x)| < \varepsilon \quad (1)$$

The continuous function in the closed interval can be approximated through a polynomial in each point of the interval with an error lower than ε . Important examples are the Taylor, Lagrange, Chebyshev, and Hermite polynomials.

The difference between least squares adjustment and data interpolation is that in the latter, data is assumed accurate; however, we know errors are common, in the minimum least squares adjustment a function such that

$$F(a_1, a_2, \dots, a_n) = \sum_{i=1}^n [f(x_i) - y_i]^2 \quad (2)$$

is sought.

It is a good approximation when a minimum exists. Classical methods are linear regression, polynomial, multilinear, power series, nonlinear; we are focused on the linear and nonlinear regression cases, except when this can be reduced to a linear regression problem, such as exponential and power functions. This approach facilitates the derivation of the necessary calculations in each method.

In our current work, we analyze a group of exercises proposed in textbooks related to numerical analysis and programming in science and engineering courses. Likewise, the mathematical elements from where the applied formulas are derived are presented. For example, one book that accurately covers this subject is from Vázquez et al. [1]. Similarly, [2] is of common use and these comprise the main sources where the proposed exercises were taken and forms of which are here developed. Both works are at university level and are outstanding due to their clarity of presentation while respecting the depth in the treatment of the numerical analysis.

In literature about numerical analysis using MATLAB, the work of Bai Yuand Zhang Meng [3] can be mentioned; this work justifies our proposal and philosophy to contribute with didactic innovations. A breakthrough work is that of Eisa San [4] which presents a method for computing the length of curves using polynomial interpolation with a MATLAB algorithm. Another related work is that of Cavoretto [5] where he proposed an algorithm for modeling data points with MATLAB. Finally, a book with approximation theory is that of Robert Plato [6], which includes polynomials from Chebyshev and Fourier and splines functions. This work is more theoretical.

We are focused on solving the issues raised in Vázquez et al [1]; some examples are available on the Internet such as [7,10,11] and in the book by Mathews/Fink [2] the intent of which is to develop our own solution and generate a discussion that leads to better results with MATLAB regarding interpolated and regression functions. Such an approach can bring about the best didactic on this subject.

II. Lagrange Interpolation

For a group of points $(x_i, f(x_i))$, a polynomial that matches with these points is sought; therefore, having a $P(x)$ such that

$$(x_i, f(x_i)) = (x_i, P(x_i)) \quad (3)$$

Lagrange proposed the following model based on linear interpolation

$$P(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \quad (4)$$

The following is met

$$\begin{cases} P(x_0) = f(x_0) \\ P(x_1) = f(x_1) \end{cases} \quad (5)$$

The polynomial can be built as follows:

$$P(x) = L_0(x)f(x_0) + L_1(x)f(x_1) \quad (6)$$

The coefficients of this function are called Lagrange Coefficient for the x_0, x_1 nodes

$$\begin{cases} L_0(x_0) = 1 & L_1(x_0) = 0 \\ L_0(x_1) = 0 & L_1(x_1) = 1 \end{cases} \quad (7)$$

In general for $n + 1$ points if $x_0 < x_1 < x_2 < \dots < x_n$ we have

$$P(x) = \sum_{k=0}^n L_k(x)f(x_k) \quad (8)$$

which contains the Lagrange polynomials of n grade that can be written as

$$L_k(x) = \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i} \quad (9)$$

this is sufficient for the first exercise: Obtain the Lagrange polynomial of the following points [7]

x	-6	-4	-3	2
y	-50	-16	-5	-10

The interpolating polynomial of these data is

$$P_2(x) = 4 - 3x - 2x^2 \quad (10)$$

Figure 1 shows its graph

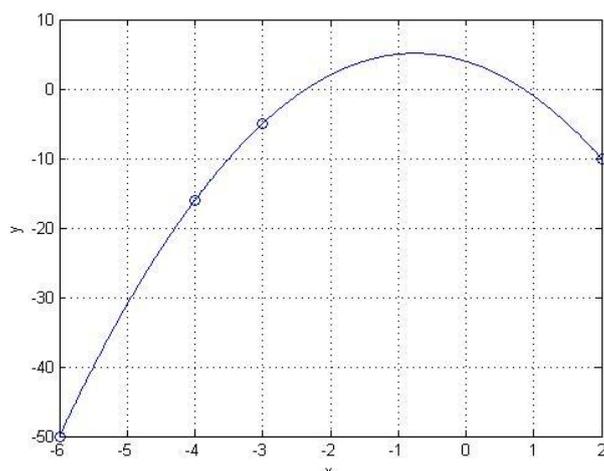


Figure 1. Lagrange Polynomial

The book of Luis Vázquez et al [1] contains, in the exercise section, some interesting problems. For example No. 7, page 115, that attempts to find a Lagrange interpolating polynomial for a function given by the following table

x	-2	-1	0	1	2
f(x)	3	1	-1	0	7

Using MATLAB we get that the polynomial results in

$$P(x) = \frac{1}{2}x^3 + \frac{3}{2}x^2 - x - 1 \quad (11)$$

Figure 2 shows its behavior

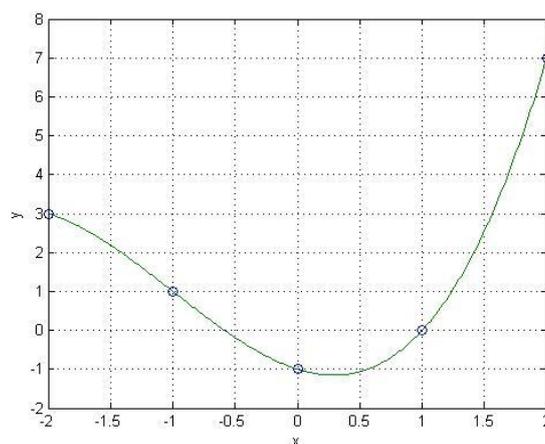


Figure 2. Lagrange Polynomial

III. Newton Interpolation

Sometimes it is useful to build several approximate polynomials and choose the correct one. Lagrange is not convenient because each one is independent. Those of Newton are estimated using the recursive system. In fact,

$$\begin{aligned}
 P_1(x) &= a_0 + a_1(x - x_0) \\
 P_2(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \\
 P_2(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \\
 & a_3(x - x_0)(x - x_1)(x - x_2)
 \end{aligned} \quad (12)$$

in such a way that

$$P_n(x) = P_{n-1}(x) + a_n(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-1}) \quad (13)$$

This is known as Newton's Divided Difference Polynomial Interpolation. There is a relationship between the Lagrange polynomial and Newton polynomial, that is, it is possible to directly obtain the Lagrange polynomial from Newton's formula from the concept of divided difference. This development can be found in [8]. Equation (4) is generated from (13).

Nakamura [9] has a different view of the study of interpolation; he begins with linear interpolation, then power series polynomials and, from there, arrives at Lagrange; if we add the approach [8], the entire demonstrative picture can be covered.

Example of the exercises proposed in [10]. Calculate the interpolation polynomial for the given table using Newton's system and formula:

x	0	2	4	6
y	0.25	0.6	0.9	1

From this calculation we get

$$y = 0.25 + 0.1625x + 0.012x^2 - 3.125 \times 10^{-3} x^3 \quad (14)$$

With the following graph, in Figure 3

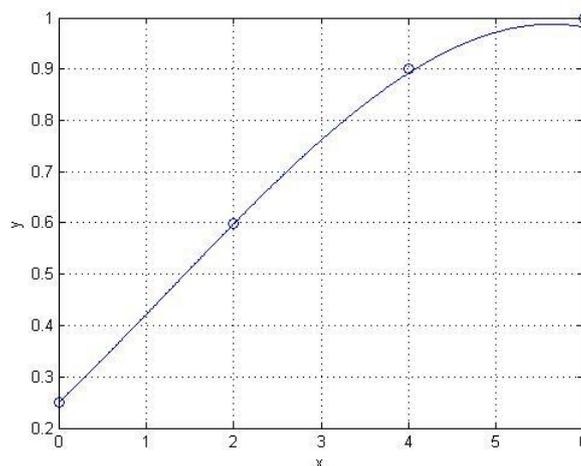


Figure 3. Newton Interpolating Polynomial

Hermite Polynomial

Taylor polynomial allows approximating a point, the function and its n first derivatives. Lagrange polynomials approximate $n + 1$ points and none of its derivatives. Now the case that is approximated to the function and to its first derivative that corresponds to the Hermite polynomials

$$H_{2n+1}(x) = \sum_{k=0}^n f(x_k) \bar{H}_k(x) + \sum_{k=0}^n f'(x_k) \hat{H}_k(x) \quad (15)$$

$$\bar{H}_k(x) = [1 - 2(x - x_k)L'_k(x_k)]L_k^2(x) \quad (16)$$

$$\hat{H}_k(x) = (x - x_k)L_k^2(x)$$

One exercise that we can discuss is the following: Find the grade three polynomial that passes through the following points [11]

0	1/3	1/2	1
4/9	8/9	5/9	6/9

The grade three Hermite polynomial is

$$P_3(x) = \frac{4}{9} + \frac{47}{9}x - 15x^2 - 10x^3 \quad (17)$$

With the following graph, in Figure 4

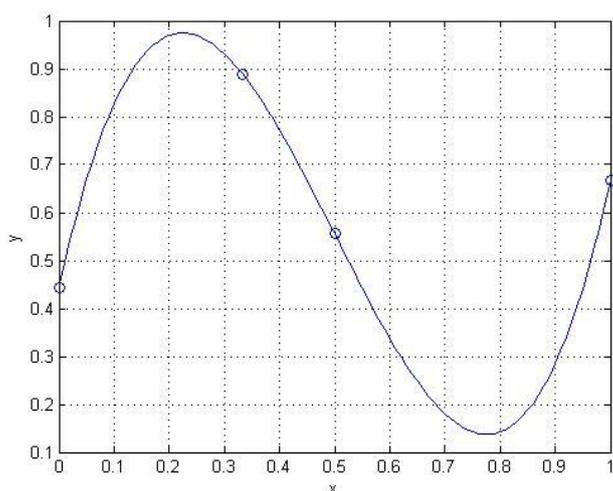


Figure 4. Hermite Polynomial

Osculating Polynomial

A general case is to consider a polynomial that approximates to the function, and some of its derivatives, such that

$$\frac{d^k P(x_i)}{dx^k} = \frac{d^k f(x_i)}{dx^k} \quad (18)$$

A solved exercise in Vázquez et al [1] is of our interest, if we complete it using graphic analysis.

x_i	0	1	2
y_i	1	0	9
y'_i	0		24

which leads to determine that

$$P_4(x) = (1 - x^2)^2 \quad (19)$$

with the following graph, in Figure 5

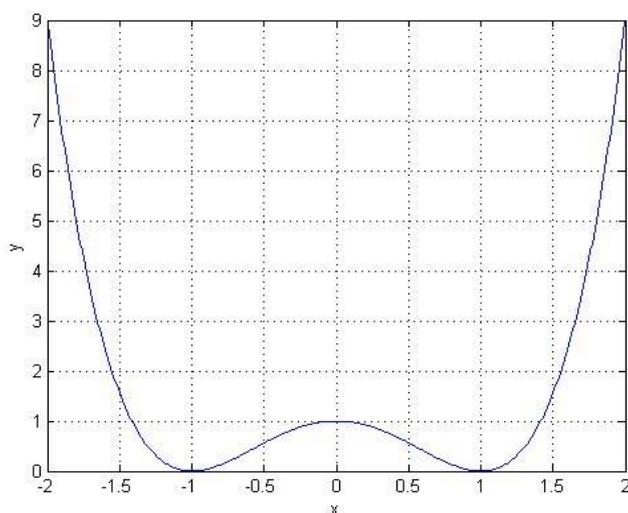


Figure5. Osculating Polynomial

Another possibility is the rational approximation of a function, in a polynomial ratio, and in at least a small portion of the domain. It can be demonstrated that such ratio is $R_{N,M}(x) = \frac{P_N(x)}{Q_M(x)}$ for $a \leq x \leq b$

(20)

Finally, a Padé approximation is given generated by the following exercise proposed in the book of Mathews/Fink. [2] Compare the Padé approximation such that

$$\sin(x) \approx \frac{166320x - 22260x^2 + 551x^5}{15[11088 + 364x^2 + 5x^4]} \quad (21)$$

Figure 6 contains both functions (discontinuous approximation)

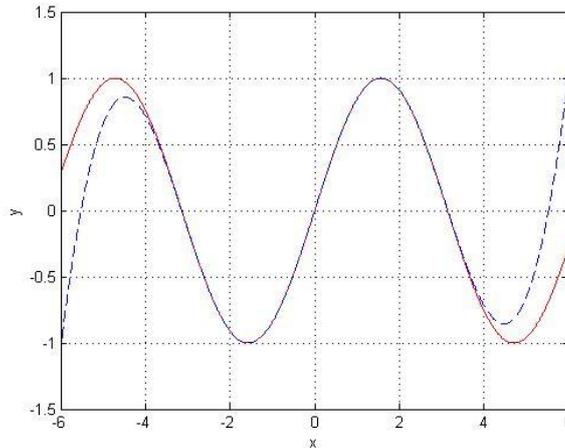


Figure 6. Padé Approximation

IV. Curve Fitting with Least-Squares Line

The interpolation assumes exact values, however, it is common to have an experimental measure with errors. In this case, what is sought is a function that does not pass through the points rather one that minimizes the error made. For example, an estimation of the error has the following form

$$|y_i - f(x_i)| \quad (22)$$

however, the one that is used most is the mean squared error, equation (2). We present the equations of the linear regression, and examples of exponential polynomial regression and power function.

V. Linear Regression

The simplest case, in this case function f is a straight line, therefore

$$f(a,b,x) = a + bx \quad (23)$$

and the function to minimize is

$$F(a,b) = \sum_{i=1}^n (ax_i + b - y_i)^2 \quad (24)$$

then, it is necessary to solve

$$\frac{\partial F(a,b)}{\partial a} = 2a \sum_{i=1}^n x_i^2 + 2b \sum_{i=1}^n x_i - 2 \sum_{i=1}^n x_i y_i = 0 \quad (25)$$

$$\frac{\partial F(a,b)}{\partial b} = 2a \sum_{i=1}^n x_i + 2nb - 2 \sum_{i=1}^n y_i = 0 \quad (26)$$

we arrive at

$$a = \frac{n \sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)}{n \left(\sum_{i=1}^n x_i^2 \right) - \left(\sum_{i=1}^n x_i \right)^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (27)$$

$$b = \bar{y} - a\bar{x} \quad (28)$$

These equations have their matrix form. Likewise, we have

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (29)$$

Polynomial Regression

For the general case of polynomial regression of k grade, the function that is minimized is

$$F(a_k, a_{k-1}, \dots, a_0) = \sum_{i=1}^n \left(a_k x_i^k + a_{k-1} x_i^{k-1} + \dots + a_0 - y_i \right)^2 \quad (30)$$

taking the partial derivatives and equaling to zero

$$\begin{pmatrix} \sum_{i=1}^n x_i^{2k} & \dots & \sum_{i=1}^n x_i^{2k-1} & \dots & \sum_{i=1}^n x_i^k \\ \sum_{i=1}^n x_i^{2k-1} & \dots & \sum_{i=1}^n x_i^{2k-2} & \dots & \sum_{i=1}^n x_i^{k-1} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{i=1}^n x_i^k & \dots & \sum_{i=1}^n x_i^{k-1} & \dots & n \end{pmatrix} \begin{pmatrix} a_k \\ a_{k-1} \\ \cdot \\ \cdot \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i^k y_i \\ \sum_{i=1}^n x_i^{k-1} y_i \\ \cdot \\ \cdot \\ \sum_{i=1}^n y_i \end{pmatrix} \quad (31)$$

The matrix form is observed. Next, we have problem 18, page 118, from the book of Vázquez et al [1].

Polynomial regression of the data to a parabola:

x	-2	-1	-1	0	2	2	3	4	4	5
y	8.2	2.5	2.6	0.2	-0.5	0.10	2.6	8.9	8.6	15.8

Resulting in

$$y = 1.063x^2 - 2.0562x - 0.3143 \quad (32)$$

with the following graph, in Figure 7

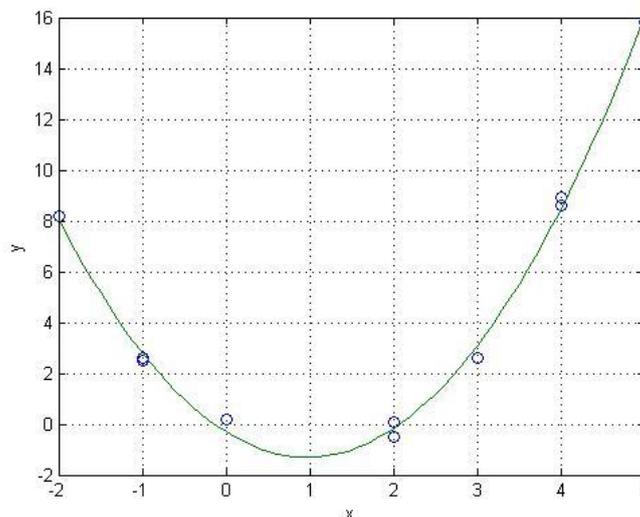


Figure 7. Polynomial Regression

Exponential Regression

In this case the function

$$f(a,b,x) = be^{ax} \quad b > 0 \quad (33)$$

To make a data linearization the following is necessary

$$\ln f(x) = \ln b + ax \quad (34)$$

Since it is a linear regression, it will have the form of equations (25) and (26)

$$a = \frac{n \sum_{i=1}^n x_i \ln(y_i) - \left(\sum_{i=1}^n x_i\right) \left(\sum_{i=1}^n \ln(y_i)\right)}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2} \quad (35)$$

For parameter a , equations (25) and (26) are considered

$$b = \exp\left(\frac{1}{n} \sum_{i=1}^n \ln(y_i) - \frac{a}{n} \sum_{i=1}^n x_i\right) \quad (36)$$

Problem 21 from Vazquez et al [1]

Make an adjustment of exponential decline law of the following data

x	1	2	3	4	5	6	7	8	9	10
y	9.1	6.1	5	5	4.1	2.1	2.8	1.8	1.3	2.2

whose resulting equation is

$$y = 9.4204e^{-0.1886x} \quad (37)$$

with the following graph in Figure 8

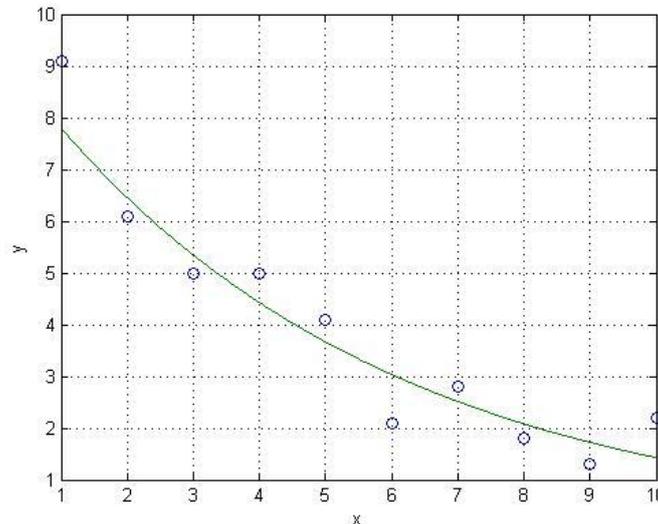


Figure8. Exponential Regression

For a group of data it is possible to predict which function provides a good adjustment. For an exponential function, the representation of the data with the x axis, linear, and the y axis, logarithmic, we see that such approximation is not accurate, not conforming to a straight line. Figure 9 shows this non-conformity

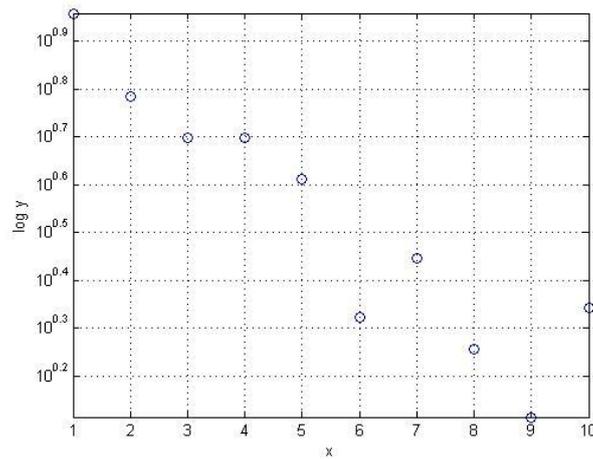


Figure9. Verification of data to a straight line

Power regression

Function $f(a, b, x)$ has the form

$$f(x) = bx^a \quad (38)$$

When the function is linearized, we have

$$\ln f(x) = \ln b + a \ln x \quad (39)$$

In this case, coefficients to verify are

$$a = \frac{n \sum_{i=1}^n \ln(y_i) \ln(x_i) - \left(\sum_{i=1}^n \ln(y_i) \right) \left(\sum_{i=1}^n \ln(x_i) \right)}{n \left(\sum_{i=1}^n \ln(x_i)^2 \right) - \left(\sum_{i=1}^n \ln(x_i) \right)^2} \quad (40)$$

$$b = \exp \left(\frac{1}{n} \sum_{i=1}^n \ln(y_i) - \frac{a}{n} \sum_{i=1}^n \ln(x_i) \right) \quad (41)$$

Next, we present the approximation of the data of problem 21, a power

$$y = 10.738x^{-0.773} \quad (42)$$

which has the behavior given in Figure 10.

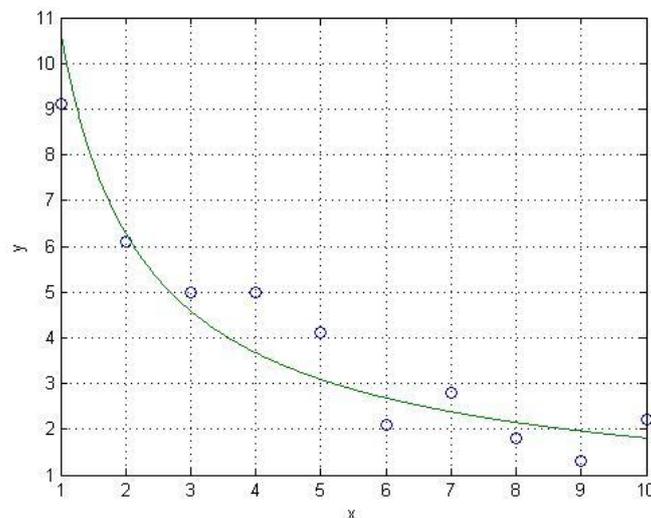


Figure10. Power Regression

Given the differences, it is necessary to test to see if the x and y axis are logarithmic, we have the result shown in Figure 11.

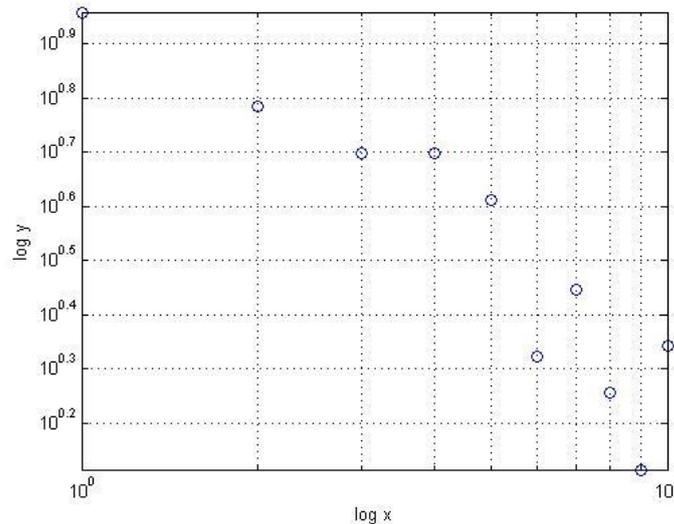


Figure11. Verification of data to a straight line

We think it is not a good approximation; with an n degree polynomial we get

$$y = 2.084x^4 + 2.089x^3 + 2.095x^2 + 2.101x + 2.106 \quad (44)$$

Since it approximates to the following form of Figure 12

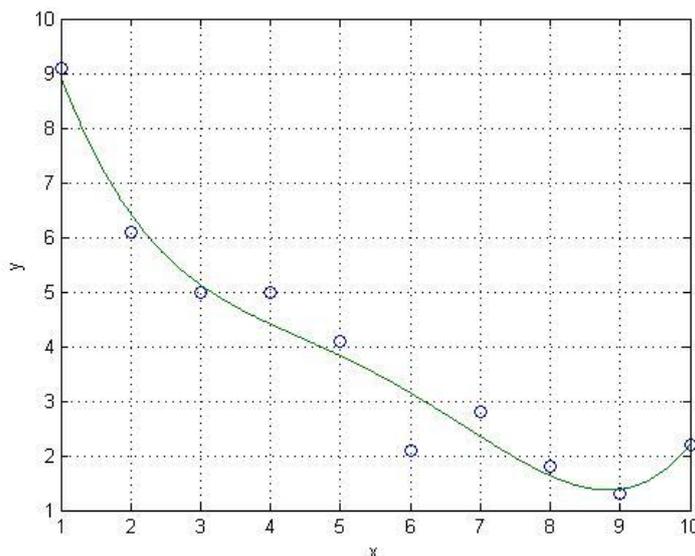


Figure 12. Polynomial Regression

VI. Discussion of Results

On the basis of the analysis carried out, we can establish that references [1] and [2] are complementary; both works have details that can help understand the subject better. We focused more on [1]. In Lagrange polynomials, MATLAB obtains the graph in Figure 2 directly. Likewise, in the Hermite problem of Figure 4 it is necessary to use the method of divided differences for its solution; another case is the example of the osculating polynomial, solved in [1]. They do not generate the graph of Figure 5. Padè approximation of Figure 6 is in a different interval to the one asked for in the book of Mathew-Fink [2] where it has a good result. Equations 10, 11,14,and 17 are consistent with the sources. In the topic of least-squares lines with data linearization method , we have Figure 7 in polynomial regression where we observe an excellent adjustment, while in problem 21 concerning the exponential decline from Vázquez et al [1] does not exhibit a good approximation. We believe that in Figures 8 to 12, a priority, for our work the result of Figure 12 is better. Finally, we acknowledge that we did not discuss and estimate the approximation errors. It is crucial to point out

that the deductions of the equations have to be duly complemented; otherwise, we only provide the main result of the method overlooking its mathematical justification.

VII. Conclusions

The study of numerical analysis in a science or engineering course requires new didactic material. On the Internet exists a myriad of audiovisual lessons of great didactic value. Our contribution would like to be part of the trend toward the generation of new approaches, but with the characteristic of putting emphasis on the use of MATLAB, and have time to carry out mathematical developments that demonstrate the formula of each method, such as the ones presented here. We think that the combination of plotting and calculating with software and mathematically justifying everything, is an efficient way to learn numerical analysis in science and engineering for promoting a deeper understanding of the subject of interpolation.

Referencies

- [1]. Luis Vázquez, Salvador Jiménez, Carlos Aguirre y Pedro J. Pascual. Métodos Numéricos para física e ingeniería. Edit. McGraw Hill pp 85-118. Spain (2009)
- [2]. John H. Mathews and Kurtis D. Fink. Métodos numéricos con MATLAB. Edit. Prentice Hall pp 203-263. Spain (2000).
- [3]. Bai Yu and Zhang Meng. Visual teaching of numerical analysis based on MATLAB. The 1st International conference of Information science and Engineering pp 3552-3555 (2009)
- [4]. Eisa San et al. Numerical curve length calculation using polynomial interpolation. Journal of Mathematical Imaging and Vision 49 2 pp 377-383 (2014)
- [5]. Roberto Cavoretto. A numerical algorithm for multidimensional modeling of scattering data points. Computational and Applied Mathematics. (2013)
- [6]. Robert Plato Concise Numerical mathematics. Edit. American Mathematical Society. USA (2000)
- [7]. ejercicios de interpolación y aproximación. numat.net/ejerc/interp/
- [8]. Steven C. Chapra and Raymond P. Canale. Métodos numéricos para ingenieros Edit McGraw Hill pp 458-459 Mexico (2011)
- [9]. Shoichiro Nakamura. Análisis numérico y visualización gráfica con MATLAB. Edit. Prentice Hall, pp 146-154 Mexico (1997)
- [10]. www.ugr.es/~mibanez/ejemplos/interpolacion.pdf
- [11]. Jaime Álvarez Maldonado