POLYNOMIAL INTERPOLATION USING MATRIX METHOD IN MATLAB

Siti Hawa Binti Aziz[1]

[1]Politeknik Ungku Omar
shawa@puo.edu.my

ABSTRACT

Data fitting is the problem of constructing such a continuous function. Many times, data given only at discrete points. The process of finding such a polynomial is called interpolation. In this study, researcher determining the polynomial interpolation by using matrix method. The manual calculation using matrix method take a long time and complicated especially for a points more than three. So, a mathematical modelling was built by using MATLAB programming to determine the polynomial interpolation for a given points using Matrix method. The result of the study presented that the manual calculating and the MATLAB mathematical modelling gave the same answer for the points given. This project will help students to understand further more about polynomial interpolation using matrix method as it is not only shows the equation for the polynomial but it also shows the graph for given points.

Keywords: Polynomial, Interpolation, Matrix Method, MATLAB

1. Introduction

The problem of constructing a continuously defined function from a given discrete data is unavoidable whenever one wishes to manipulate the data in a way that requires information not included explicitly in the data. Interpolation, by polynomials or other functions, is a rather old method in applied mathematics. Gasca mention that it is already indicated by the fact that, apparently, the word 'interpolation' itself has been introduced by J. Wallis as early as 1655 (Gasca & Saeur, 2000). Sir Edmund Whittaker, a professor of Numerical Mathematics at the University of Edinburgh from 1913 to 1923, said "the most common form of interpolation occurs when we seek data from a table which does not have the exact values we want" (Meijering, 2002). Many problems concerning the applications of neural networks, such as in pattern recognition and systems control, can be converted into the ones of approximating multivariate functions by the superposition of activation functions of the neural networks, for which an extensive study on approximation by neural networks has been carried out in a huge literature (Li, 2002). In engineering or science, there always has a number of data points, obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable. However, there are several problems which may occur when we want to calculate manually the polynomial interpolation by using matrix method, such as it was complicated and time consumption especially for three points and above. Then, as we want to understand further more about the polynomial interpolation, we should know the graph of the polynomial. So, to overcome the problem, the researcher creates the command window by converting the manual calculation into suitable MATLAB command. The researcher didn't use specific build-in command for polynomial, but design a command depends on manual calculation of Matrix method. It used the basic built-in command such as *loop* and *plot*; and compatible it with another command such as, *inpu*t to insert data, *inv = inv(A)* to find the inverse of the square matrix *A* and many other commands. All models such as linear, quadratic or cubic are using the same coding. The differences are only at the equation used and the colour of the graph. For this study, researcher use the MATLAB 7.8.0 (R2009a) which has a specific command for matrix but researcher do some modification so that it is not only find the polynomial interpolation using Matrix method, but it also shows the graph generated for given points.

2. Literature Review

Curve fitting is used in a variety of fields, especially in physics, mathematics and economics. The method is often used to smooth noisy data and for doing path planning. Many methods and application has been discussed about the curve fitting and a research about the curve fitting has been grown up faster as it is important in our life. For instance, Andersson et al. (2009) used the calculus of variations to derive a formula for finding an optimal curve to fit a set of data points. One research which concerned with a practical method for fitting an ordered set of data in space with a free-form curve, with no specific function or parameterization given for the data has been done (Lane,1995). Problems such as this arise routinely in a variety of disciplines from the Arts to Engineering and Science. The techniques presented in the thesis is for data in 2 plane (2 dimensions), but can be adapted to many dimensions (Lane, 1995). According to Meijering (2002), the problem of interpolation by finite or divided differences had been studied at the beginning of the 20th century by astronomers, mathematicians, statisticians, and actuaries and most of the now well-known variants of Newton's original formulae had been worked out. There are many researches about the polynomial interpolation. Various matrix methods for solving univariate polynomial equations have been proposed in many research (Piers, 2013). Polynomial interpolation has been used to solve many problems in Mathematics. T.Yabe and T.Aoki from Japan have developed a new universal solver for hyperbolic equations by using a cubic-polynomial interpolation (Yabe & Aoki, 1991). According to Vikstrom (2009), there are several benefits of starting the algorithm development process in MATLAB and then ending it in C. MATLAB offers a wide selection of functions, automatic memory handling of variables and other interesting features that allows the engineer to focus on the function of the algorithm instead of the practical implementation. MATLAB also simplifies the algorithm testing process with its ability to easily produce plots and reports. However, the process to translate the MATLAB code to a language which is more suitable for hardware implementation such as C is time consuming and error prone (Vikstrom, 2009). Besides that, some numerical experiences with MPI are described by Saeur (1995) . Hussain (1994) has shown in his master thesis that polynomial can be solved using three types of software which is MATLAB, Maple and Mathematica. An obvious fact, however, is that the MATLAB routines are easier to handle and therefore are much more suitable for the casual user who is mainly interested in interactively experimenting with polynomial interpolation without having to write and compile a program (Gasca & Saeur, 2000). On the other hand, the C++ routines in MPI are much more trimmed for efficiency, but in order to apply them clearly a certain background in programming is needed to use classes in an appropriate way.

3. Polynomial Interpolation Using Matrix Method in Matlab

3.1. Introduction

In this subtopic, it shows the methodology on how to design the mathematical modelling for polynomial interpolation. Starting from the manual calculation to find the value of coefficients such as *a*, *b*, *c*, *d* and *e* for every equation using matrix method, then researcher convert into a coding in MATLAB. Researcher does not use *polyval* command, but design a user-friendly interface to insert any input and create the graph by doing some modification on a command in MATLAB.

3.2. Linear Function

3.2.1. Manual calculation by using inverse matrix

A linear function, of the form $y = mx + c$, is determined by two points. The researcher use Matrix method to find the value of *m* and *c* for polynomial interpolation degree 1. Let say there are two points with coordinates *(2,*-5) and *(-3,8).* We want to create a linear equation by using these two points. We know that linear equation is $y = mx + c$. So, for two points above,

$$-5 = 2m + c \quad \text{...(1)}$$
$$8 = -3m + c \quad \text{...(2)}$$

Then, by using matrix method, convert equation (1) and (2) into matrix form as below and solve the matrix to find the value of *m* and *c*:

$$\begin{pmatrix} 2 & 1 \\ -3 & 1 \end{pmatrix}\begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} -5 \\ 8 \end{pmatrix}$$

After solving the matrix, we get the answer of $m = \frac{-13}{5}$ and $c = \frac{1}{5}$ and the linear equation with two points *(2, -5)* and *(-3,8)* is:

$$y = \frac{-13x}{5} + \frac{1}{5}$$

3.2.2. Linear function by using MATLAB

After researcher did the manual calculation, the researcher writes the command or function in the m-file. After the M-file was debugged or run, the command window will appear as in Figure 1 and the graph generated as in Figure 2.

```
We want to find the value of m and c of y=mx+c if two points are given
Enter coordinate x1: 2
Enter coordinate y1: -5
Enter coordinate x2: -3
Enter coordinate y2: 8
Matrix A is :
     2     1
    -3     1

Matrix y is :
    -5
     8

Value of m is :
   -2.6000

Value of c is :
    0.2000
```

Figure 1. Command window in MATLAB for linear function.

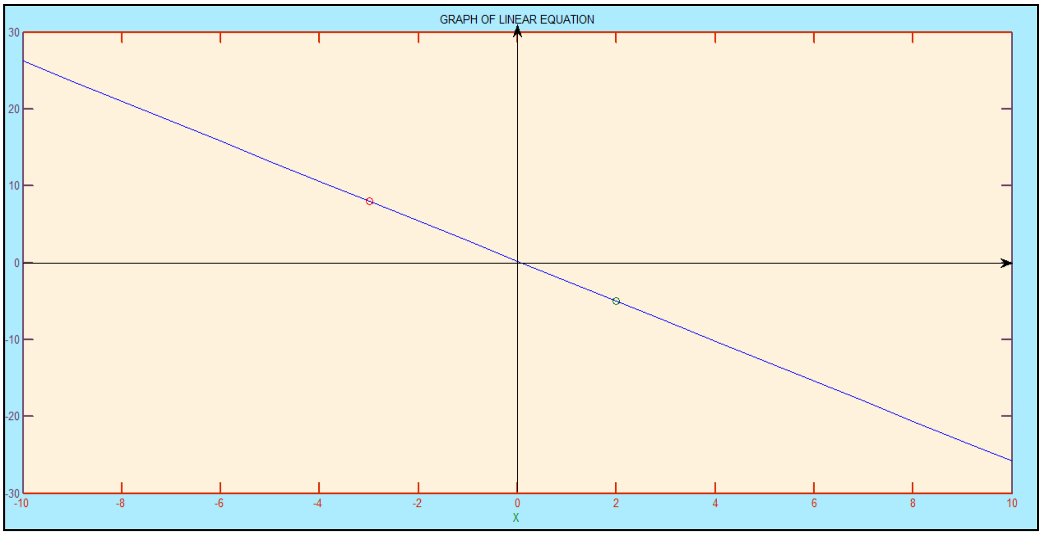

Figure 2. Command window in MATLAB for linear function.

3.3. Quadratic Function

3.3.1. Manual calculation using matrix method

A quadratic function, of the form $y = ax^2 + bx + c$, is determined by three points. Let say there are three points (-2,5), (1,8) and (6,4) and researcher solve the polynomial interpolation by using Matrix method. First, substitute the points into equation $y = ax^2 + bx + c$. These three points give the following equations:

$$5 = a(-2)^2 + b(-2) + c$$
$$5 = 4a - 2b + c \qquad (1)$$
$$8 = a(1)^2 + b(1) + c$$
$$8 = a + b + c \qquad (2)$$
$$4 = a(6)^2 + b(6) + c$$
$$4 = 36a + 6b + c \qquad (3)$$

The value of *a*,*b* and *c* can be determined more easier by finding the inverse of a matrix and solve the matrix. First, rewrite the equation into matrix form as below.

$$\underset{A}{\begin{pmatrix} 4 & -2 & 1 \\ 1 & 1 & 1 \\ 36 & 6 & 1 \end{pmatrix}} \underset{x}{\begin{pmatrix} a \\ b \\ c \end{pmatrix}} = \underset{b}{\begin{pmatrix} 5 \\ 8 \\ 4 \end{pmatrix}}$$

Let assume the above matrix as matrix *A*, *x* and *b*. Find the value of matrix *x*. So,

$$Ax = b$$
$$x = A^{-1}b$$

Then, determine the inverse of matrix *A*, A^{-1}.
Since, x = A^{-1}B, find the inverse of matrix A. By using Elementary Row Operation:

$$[A|I] \Rightarrow [I|A^{-1}]$$
$$\left(\begin{array}{ccc|ccc} 4 & -2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 36 & 6 & 1 & 0 & 0 & 1 \end{array}\right)$$

After solving the matrix, the inverse of matrix A, $A^{-1} = \begin{pmatrix} \frac{1}{24} & \frac{-1}{15} & \frac{1}{40} \\ \frac{-7}{24} & \frac{4}{15} & \frac{1}{40} \\ \frac{1}{4} & \frac{4}{5} & \frac{-1}{20} \end{pmatrix}$.

From $Ax = b$, determine matrix x and we get the answer as below:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \frac{-9}{40} \\ \frac{31}{40} \\ \frac{149}{20} \end{pmatrix} = \begin{pmatrix} -0.225 \\ 0.775 \\ 7.45 \end{pmatrix}$$

Finally, the quadratic equation for three points (-2,5), (1,8) and (6,4) is ;

$$\boldsymbol{y = -0.225x^2 + 0.775x + 7.45}$$

3.3.2. Quadratic function by using MATLAB

After the manual calculation has been done, the researcher converts the calculation into the command in MATLAB. The coding that researcher are using is same as linear function, but the difference is only at the number of inputs, the matrix notation, the equation which is quadratic function and the value of coefficients which are a, b and c. After we run the M-file, the result is shown in the command window in Figure 3 and the graph will generate as in Figure 4.

```
We want to find the value of a, b and c for y=ax^2 + bx + c if three points are given
Enter coordinate x1: -2
Enter coordinate y1: 5
Enter coordinate x2: 1
Enter coordinate y2: 8
Enter coordinate x3: 6
Enter coordinate y3: 4
Matrix A is :
     4    -2     1
     1     1     1
    36     6     1

Matrix y is :
     5
     8
     4

The value of a is :
   -0.2250

The value of b is :
    0.7750

The value of c is :
    7.4500
```

Figure 3. Command window in MATLAB for quadratic function

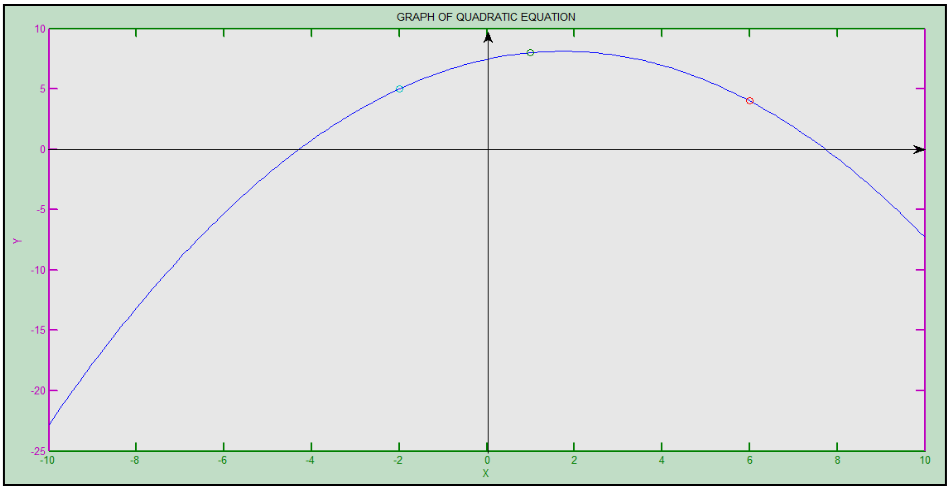

Figure 4. Graph generated in MATLAB for quadratic function

3.4. Cubic Function

3.4.1. Manual calculation by using matrix method

Since the cubic equation have four points, solving using Matrix Method is easier than solving using simultaneous equation. If you have four distinct points in the xy-plane, and no two x-coordinates are equal, then there is a unique cubic equation of the form $y = ax^3 + bx^2 + cx + d$ that passes through the four points. We can use matrix algebra to find the coefficients *a*, *b*, *c*, and *d*, or use the convenient calculator on the left. The matrix method of solving this problem is explained below, and a graphing calculator can be used to quickly compute matrix operations. The value of coefficient can be determine by solving a matrix equation *Ax* = *B*, where *A* is a 4-by-4 matrix, *x* is a 4-by-1 matrix of the determined coefficients, and *B* is a 4-by-1 matrix. The entries of matrix A are determined by the x-coordinates of the pairs (x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4). While, the entries of matrix *B* are the y-coordinates of the pairs (x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4). Thus, the full matrix equation is:

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

The solution to this matrix equation is $x = A^{-1}B$. As long as all of the *x* values are distinct, matrix *A* will be invertible, thus, there will be a unique solution set for *a*, *b*, *c*, and *d*. Let say there are 4 points given *(1,2), (3,5),(0,2)* and *(-2,-3).* Substitute the values of *x* and *y* in the matrix form, we get the matrix below:

$$\underset{A}{\begin{bmatrix} 1^3 & 1^2 & 1 & 1 \\ 3^3 & 3^2 & 3 & 1 \\ 0^3 & 0^2 & 0 & 1 \\ (-2)^3 & (-2)^2 & -2 & 1 \end{bmatrix}} \underset{x}{\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}} = \underset{b}{\begin{bmatrix} 2 \\ 5 \\ 2 \\ -3 \end{bmatrix}}$$

Since, $x = A^{-1}B$, find the inverse of matrix *A*. By using Elementary Row Operation. After solving the matrix, we found that the value of A^{-1}, inverse of matrix *A* is

$$A^{-1} = \begin{pmatrix} \frac{-1}{6} & \frac{1}{30} & \frac{1}{6} & \frac{-1}{30} \\ \frac{1}{6} & \frac{1}{30} & \frac{-1}{3} & \frac{2}{15} \\ 1 & \frac{-1}{15} & \frac{-5}{6} & \frac{-1}{10} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

We know that $Ax = b$ and we need to solve it to find matrix *x* and get the value of matrix *x.*

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \frac{4}{15} \\ \frac{-17}{30} \\ \frac{3}{10} \\ 2 \end{pmatrix} @ \begin{pmatrix} 0.267 \\ -0.567 \\ 0.300 \\ 2 \end{pmatrix}$$

Finally, the cubic equation for 4 points *(1,2), (3,5),(0,2)* and *(-2,-3)* is ;

$$\boldsymbol{y = 0.267x^3 - 0.567x^2 + 0.300x + 2}$$

3.4.2. Cubic equation by using MATLAB

The coding that are researcher using to convert the manual calculation into M-file is same as linear function, but the difference is only at the number of inputs which is 4 input, the matrix notation, the equation which is cubic function and the value that need to calculate which are a, b, c and d . The results are shown in Figure 5 and Figure 6.

```
We want to find the value of a, b, c and d for y=ax^3 + bx^2 + c + d if four points are given
Enter coordinate x1: 1
Enter coordinate y1: 2
Enter coordinate x2: 3
Enter coordinate y2: 5
Enter coordinate x3: 0
Enter coordinate y3: 2
Enter coordinate x4: -2
Enter coordinate y4: -3
Matrix A is :
     1     1     1     1
    27     9     3     1
     0     0     0     1
    -8     4    -2     1

Matrix y is :
     2
     5
     2
    -3

The value of a is :
    0.2667

The value of b is :
   -0.5667

The value of c is :
    0.3000

The value of d is :
     2
```

Figure 5. Command window in MATLAB for cubic function

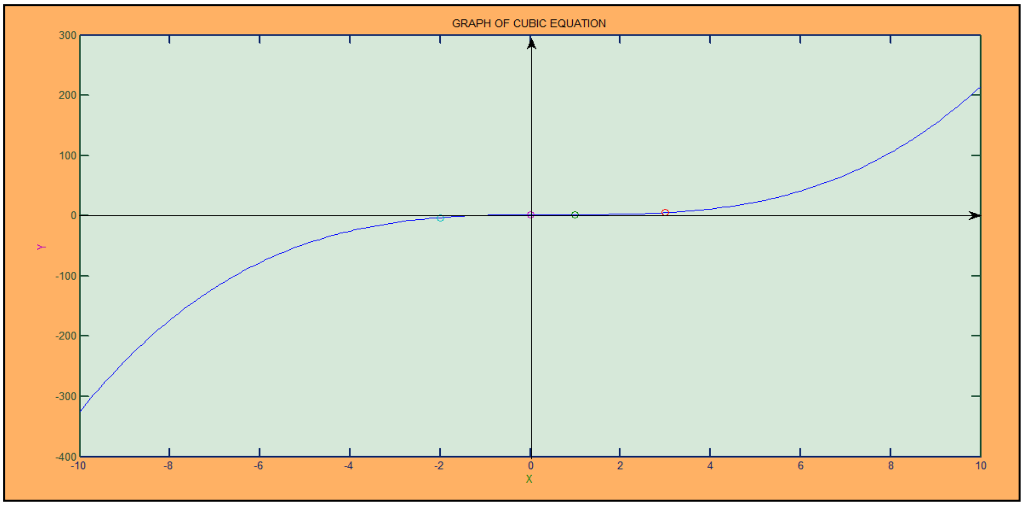

Figure 6. Graph generated in MATLAB for cubic function.

4. Discussion

In this study, researcher build a user friendly interface or a mathematical modeling to plot a graph of polynomial for a given points. The command that are using depends on manual calculation. Then it was converted into a suitable command which is already in MATLAB. In MATLAB, the main command that researcher use for the three types of polynomial are *input* which user can insert the value of *x* and *y*, *inv* which is to find the inverse of the matrix, *hold on* to hold the inserted points in a graph, *for loops* and *plot* which is important in plotting graph. *for loops* command is quite important in plotting graph because it allow a group of commands to be repeated for a fixed, predetermined number of times. The basic calculation that researcher use to find the coefficient of the polynomial, x_n is the matrix method. User can insert different points repeatedly into the model and it will automatically calculate the coefficient of the x_n and generate the graph. If we are using the built-in command *polyval* and *plot*, we need to rewrite the command every time we change the points as in Figure 7. So, by using this mathematical modeling it is easier to find the coefficient of the x_n and show the function into a graph.

```
x=[2,-3];% (we need to change the value of x everytime we want to insert different input)
y=[-5,8]; % (we need to change the value of y everytime we want to insert different input)
p=polyfit (x,y,1); % (x is the value of x ; y is the value of y ; 1 is the degree of the polynomial)

p =

   -2.6000    0.2000  %(value of m and c for equation y=mx+c)

x2=-5:.1:5;  %(limit of the graph)
y2=polyval(p,x2); %(returns the value of a polynomial of degree n evaluated at x)
plot(x,y,'o',x2,y2) % (plotting the graph)
grid on %(on the grid in the graph)
```

Figure 7. An example of command for linear equation by using built-in command.

5. Conclusion and recommendation

This mathematical modelling is more user friendly than the built-in command in MATLAB. After doing some modification with the command in MATLAB, researcher make it more user friendly which is user or students just need to insert an input and it will automatically determine the equation and show the graph. If we compare with the built-in command in MATLAB, to plot a polynomial graph for two points, it is complicated as we need to rewrite the command every time we want to insert an input. However, there is some limitation in this mathematical modelling. If new interpolation points are added, all of the polynomial interpolation must be recomputed. Researcher built a mathematical modelling for a certain points only. So, we hope, in the future, there is a general mathematical modelling by using matrix methods which can determine polynomial interpolation for any number of points. Besides that, we may design this modelling using other popular programming such as Maple and etc.

References

Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Craz, J., Greenbaum, A.,Hammarling, S. , McKenney, A., Ostrouchov, S. & Sorensen, D. *LAPACK User's Guide, Second Edition.* SIAM, 1995.

Gasca, M., & Sauer, T. (2000). Polynomial interpolation in several variables. *Advances in Computational Mathematics*.12(4), 377-410.

Lane, E.J. (1995) , *Fitting Data Using Piecewise G^1 Cubic Bezier Curves*.Naval Postgraduate School Monterey.

Lawrence, Piers W. (2013). Eigenvalue Methods for Interpolation Bases. *Electronic Thesis and Dissertation Repository*. 1359. Retrieved from https://ir.lib.uwo.ca/etd/1359

Li, X. (2002). Interpolation by Ridge Polynomials and its application in neural networks. *Journal of Computational and Applied Mathematics*.144, 197-209.

Meijering, E. (2002). A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing. In: *Proceedings of the IEEE*. vol. 90, no. 3, pp. 319-42. March 2002.

Saeur, T. & Xu, Y. (1995). On Multivariate Lagrange Interpolation. *Mathematics of Computation*. 64 (211) , 1147-1170.

Vikstrom, A. (2009), *A study of automatic translation of MATLAB code to C code using software from the MathWorks*. (Master), Lulea University of Technology.

Yabe, T. & Aoki, T. A universal solver for hyperbolic-equations by cubic-polynomial interpolation. I. one-dimensional solver, Comput.Phys.Cpmmun,66(1991),219-232.