ECSE 443 – Assignment 3

McGill University

Tristan Bouchard - 260747124

March 17, 2019

Introduction

This assignment required us to make use of the interpolation methods explored in class, namely the Cubic Spline, Lagrange and Newton methods, to interpolate certain values in a provided dataset, as well as use the cubic spline method to extrapolate what future values of the data could be. As questions 1 and 2 are identical in my approaches, I will document in detail the strategy I used for question 1, parts a) through c), the equivalent portion in question 2 being identical and such I will only document the result. As usual, the code to this assignment can be found in the root directory, under the name “Assignment3_260747124.mlx”, with all of the accompanying functions present in the “./Functions/” directory.

Question 1 - Interpolation

Question 1 of this assignment required us to interpolate values from a provided data set using a few different methods.

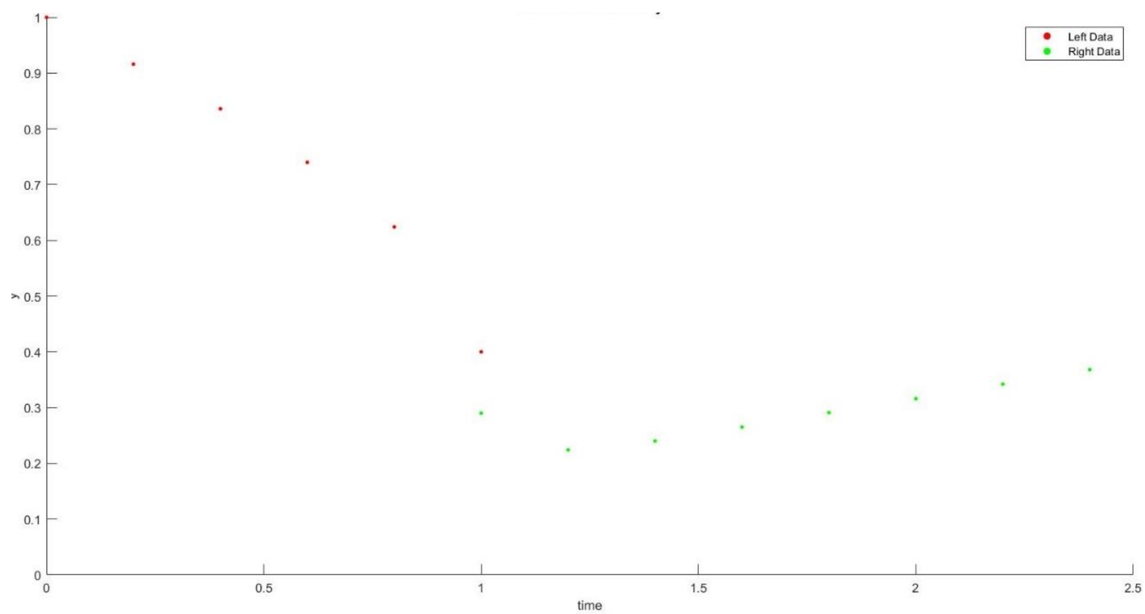

Figure 1: Provided data, with discontinuity

Part a) Cubic Splines

This section of the assignment required us to perform interpolation using the method of cubic splines. As is explained in the book, the cubic splines method fits a third order polynomial to a section of the data and linking each of these cubic sections together provides an interpolation for the entire data set. There are a few different methods to obtaining cubic splines, but here is the method I used for this assignment.

By definition, a cubic spline is a piecewise continuous function that is twice differentiable, with these derivatives also being continuous. Using these constraints, it is possible to build a system of equations to solve for unique cubic equations describing a section of the function. First, I began by providing my function three consecutive data points. Using these data points, it is possible to create the following system of equations.

We wish to describe the function between points

$$p_1 = (t_1, y_1)$$

$$p_2 = (t_2, y_2)$$

$$p_3 = (t_3, y_3)$$

$$where\ t_1 < t_2 < t_3$$

It is possible to do so by defining two third order polynomials over the intervals $[t_1\ t_2]$ and $[t_2\ t_3]$, in the form:

$$p_1(t) = \propto_1 + \propto_2 t + \propto_3 t^2 + \propto_4 t^3, \{ t \in \mathbb{R} :\ t_1 \le t \le t_2 \}$$

$$p_2(t) = \beta_1 + \beta_2 t + \beta_3 t^2 + \beta_4 t^3, \{ t \in \mathbb{R} :\ t_2 \le t \le t_3 \}$$

This set of equations provides 8 unknowns; we must set 8 constraints, or equations on the system. First, we know that

$$y_1 = \propto_1 + \propto_2 t_1 + \propto_3 t_1^2 + \propto_4 t_1^3$$

$$y_2 = \propto_1 + \propto_2 t_2 + \propto_3 t_2^2 + \propto_4 t_2^3$$

And

$$y_2 = \beta_1 + \beta_2 t_2 + \beta_3 t_2^2 + \beta_4 t_2^3$$

$$y_3 = \beta_1 + \beta_2 t_3 + \beta_3 t_3^2 + \beta_4 t_3^3$$

This sets 4 constraints on the system, thus we must provide 4 more. As we wish the system to be continuous, we want the derivatives of both functions to be equal at the center of the interval, t_2. Such, we set their first derivatives to be equal.

$$p_1'(t_2) = p_2{}'(t_2)$$

$$\propto_2 + 2 \propto_3 t_2 + 3 \propto_4 t_2^2 = \beta_2 + 2\beta_3 t_2 + 3\beta_4 t_2^2$$

As well as

$$p_1''(t_2) = p_2''(t_2)$$

$$2 \propto_3 + 6 \propto_4 t_2 = 2\beta_3 + 6\beta_4 t_2$$

To set the final constraints, we use the concept of a natural spline, which entails that the second derivative of a spline at the endpoints equals to zero.

$$2 \propto_1 + \ 6 \propto_4 t_1 = \ 0$$

$$2\beta_3 + \ 6\beta_4 t_3 = 0$$

It is then possible to place all the above equations into the following $\boldsymbol{Ax} = \boldsymbol{y}$ form. I have taken this figure out from the textbook, on page 328, as I could not figure out how to write out an 8 by 8 matrix in word!

$$\begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & 0 & 0 & 0 & 0 \\ 1 & t_2 & t_2^2 & t_2^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & t_2 & t_2^2 & t_2^3 \\ 0 & 0 & 0 & 0 & 1 & t_3 & t_3^2 & t_3^3 \\ 0 & 1 & 2t_2 & 3t_2^2 & 0 & -1 & -2t_2 & -3t_2^2 \\ 0 & 0 & 2 & 6t_2 & 0 & 0 & -2 & -6t_2 \\ 0 & 0 & 2 & 6t_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6t_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_2 \\ y_3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Figure 2: System of equations. Taken from textbook.

Solving this system of equations for the coefficients vector requires simple linear algebra, easily performed with MATLAB.

So, this question required us to find the value of the function at t = 0.23. To do this, I computed the cubic splice by using the points $(0.2, 0.916), (0.4, 0.836)$ *and* $(0.6, 0.74)$. Graphing the functions upon the data yields the following figure.

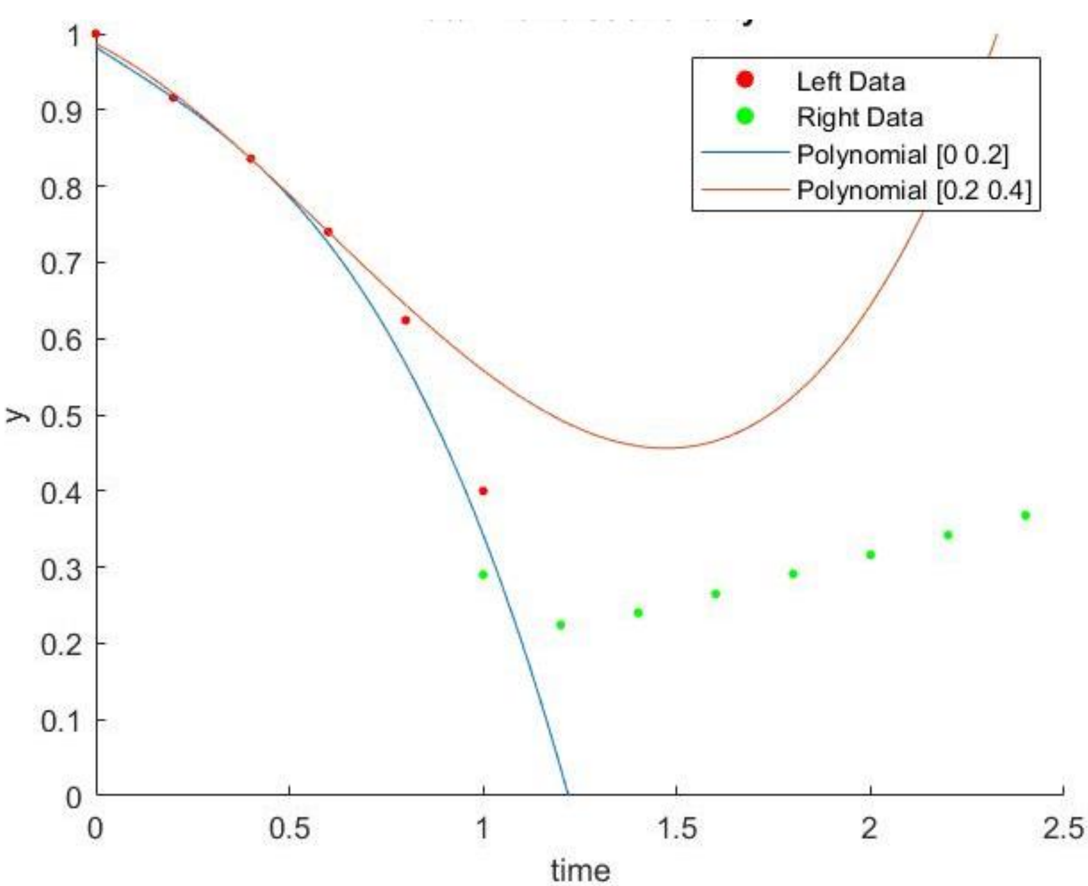

Figure 3: Cubic splines [0.2 0.6], untrimmed

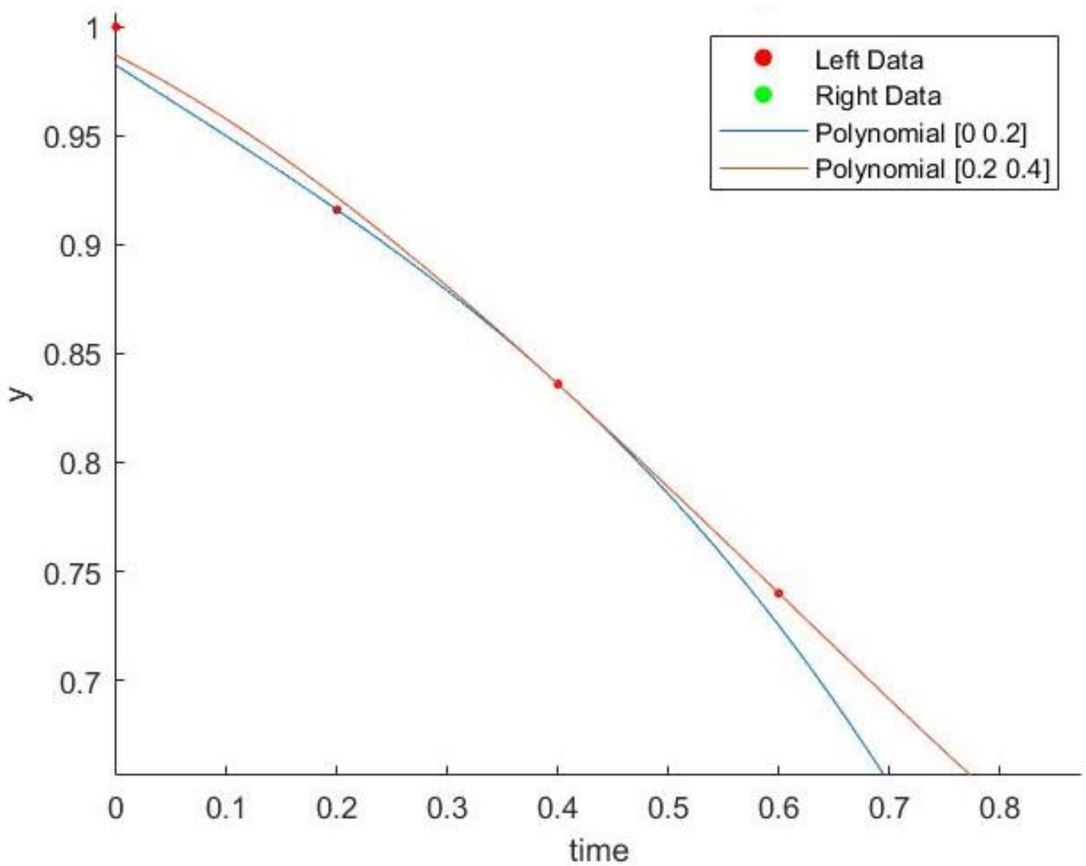

Figure 4: Cubic splines [0.2 - 0.6], zoomed

Then, using the coefficients computed for the cubic equation for the section from [0.2 0.4]:

$$p_1(t) = 0.982285714285715 - 0.325714285714286t + 0.0428571428571 43t^2 - 0.357142857142856t^3$$

Plugging in $t = 0.23$:

$$f(0.23) \cong p_1(0.23) = 0.905293214285714$$

Part b) Lagrange Interpolation

The method to the Lagrange interpolation is different to the Cubic Spline method, in the sense that it makes use of basis functions. This method is a form of polynomial interpolation, making use of the basis functions to provide a unique polynomial for any data set.

The Lagrange basis function is defined as follows, where n is the number of data points present in the dataset, and j is the $j'th$ basis function.

$$L_j(t) = \frac{\prod_{k=1,k\neq j}^{n}(t - t_k)}{\prod_{k=1,k\neq j}^{n}(t_j - t_k)}$$

Using the basis function above, it is possible to complete the polynomial interpolating the data as follows:

$$p_{n-1}(t) = y_1 L_1(t) + y_2 L_2(t) + \cdots + y_n L_n(t)$$

The basis functions, which I interchangeably name "weight" function, essentially determines the weight of any and all the data values in their contribution towards the final interpolation value.

Solving this system in MATLAB is straightforward, and the value I have computed for $t = 0.23$ equals:

$$f(0.23) \cong p_{n-1}(0.23) = 0.905058821093750$$

Part c) Newton Polynomial Interpolation

As in part b) above, the Newton Polynomial Interpolation method is another method which attempts to fit a unique polynomial to the data provided, by the means of the basis functions. The Newton basis functions are as follows:

$$\pi_j(t) = \prod_{k=1}^{j-1} (t - t_k)$$

This basis function is simpler than the Lagrange method, and is thus a little faster to compute. However, the computation involved in the following steps is a little bit more involved. The interpolant function is defined as follows:

$$p_{n-1}(t) = x_1 + x_2(t - t_1) + x_3(t - t_1)(t - t_2) + \cdots + x_n(t - t_1)(t - t_2) \dots (t - t_n)$$

Where the coefficients to the interpolant function can be computed using the basis / weight function in a very straightforward fashion, as this system is lower triangular. Essentially, the system can be turned into the matrix form, where the entries to the matrix are the values of the weight/basis function in the form $a_{ij} = \pi_j(t_i)$

$$\begin{bmatrix} \pi_1(t_1) & \pi_2(t_1) \dots & \pi_n(t_1) \\ \vdots & \ddots & \vdots \\ \pi_1(t_n) & \dots & \pi_n(t_n) \end{bmatrix} * \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Thus, using the discrete time values $[t_1 \dots t_n]$ provided in the dataset, it is possible, using simple linear algebra, to rearrange the system and solve for the coefficients of the function by computing:

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \pi_1(t_1) & \pi_2(t_1) \dots & \pi_n(t_1) \\ \vdots & \ddots & \vdots \\ \pi_1(t_n) & \dots & \pi_n(t_n) \end{bmatrix}^{-1} * \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

This is easily done in MATLAB. For the data points to the left of the discontinuity, I have computed the following coefficients for the function representing the fit. As there are 6 data points, we need 6 coefficients:

$$x_1 = 1$$

$$x_2 = -0.420000000000000$$

$$x_3 = 0.049999999999999$$

$$x_4 = -0.416666666666663$$

$$x_5 = 0.416666666666662$$

$$x_6 = -2.604166666666665$$

With these coefficients, I construct a vector of the form:

$$X = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6]$$

Which I multiply by the following vector, which is the basis/weight function evaluated at the point we wish to interpolate for (t = 0.23):

$$e = \begin{bmatrix} \pi_1(0.23) \\ \vdots \\ \pi_6(0.23) \end{bmatrix}$$

Such, by multiplying the two vectors together, we obtain the interpolated value:

$$Interpolated\ Value = X * e$$

Using this algorithm and my MATLAB code which I have provided, I obtained an interpolated value using the Newton Method of

$$f(0.23) \cong p_{n-1}(0.23) = 0.905058821093750$$

Question 1 – Conclusion

A summary of the above results:

Method	f(0.23)
Cubic Splines	0.905293214285714
Lagrange Interpolation	0.905058821093750
Newton Interpolation	0.905058821093750

The values obtained from the Lagrange interpolation and the Newton interpolation are identical. This is expected of course, as the polynomial that fits all 6 of the data points is unique. Thus, both methods contribute to computing the same polynomial, albeit in a slightly different manner. The result computed using the cubic splines method also falls within the same range, but it is expected that it is slightly different, as the Cubic Spline method does not consider all the data points when performing the spline.

Question 2 - Interpolation

Question 2 is very similar to question 1, the only difference being that the interpolation is to be computed for $t = 0.78$.

Part a) Cubic Splines

Again, for this section I made use of the data to the left of the discontinuity, however I changed my data points selected to be $(0.6, 0.74), (0.8, 0.624)\ and\ (1, 0.4)$. This allowed me to choose the interval from $[0.6\ 0.8]$ to use the polynomial and compute the value at $t = 0.78$.

The computed fit functions are as show in the graph below:

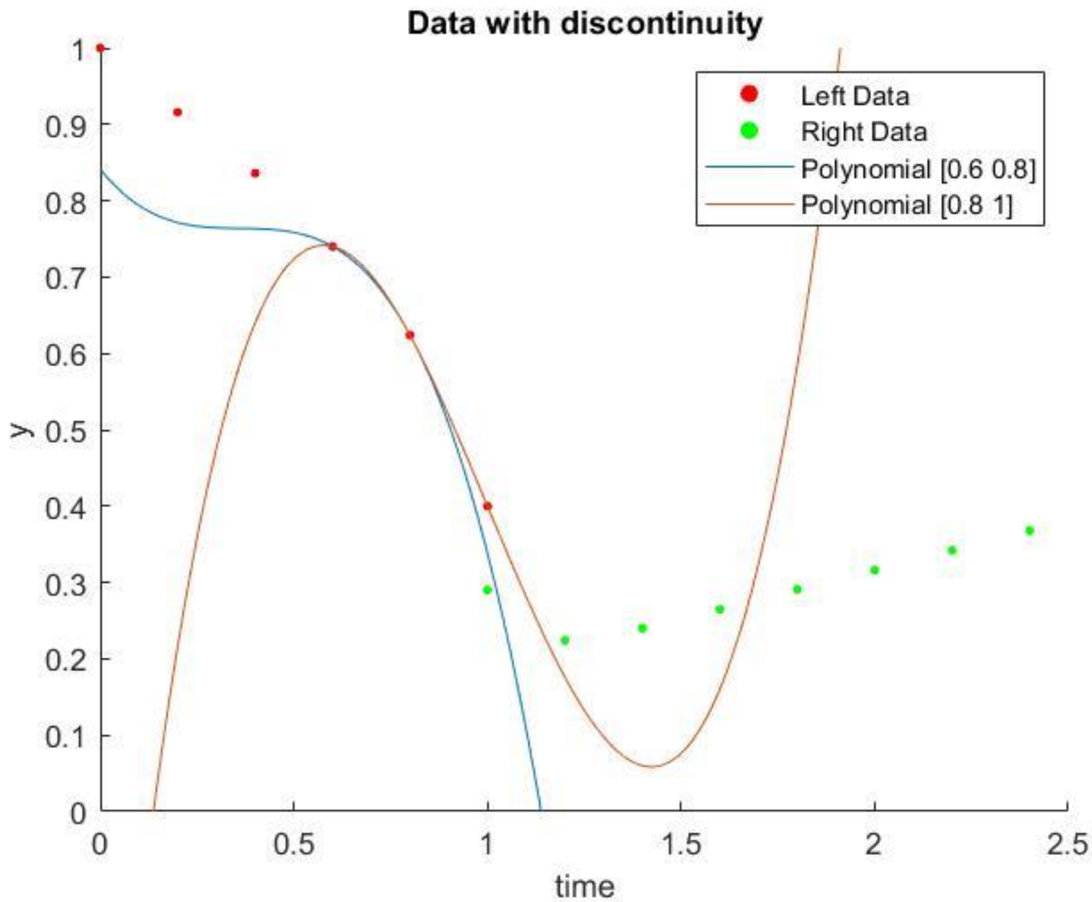

Figure 5: Cubic Splines (0.78), untrimmed

The function I obtain describing the interval [0.6 0.8] is:

$$p_{n-1} = 0.841142857142858 - 0.631428571428572t + 1.735714285714285t^2 - 1.607142857142856t^3$$

Using the method described in **Question 1 – Part a)** and the polynomial above, I obtain the following interpolated value:

$$f(0.78) \cong p_{n-1}(0.78) = 0.641964285714286$$

Part b) Lagrange Interpolation

Using the method described in **Question 1 – Part a)** and using the data points to the left of the discontinuity, I obtain the following interpolated value:

$$f(0.78) \cong p_{n-1}(0.78) = 0.637895075000000$$

Part c) Newton Polynomial Interpolation

Using the method described in **Question 1 – Part c)** and using the data points to the left of the discontinuity, I obtain the same coefficients as above, which is expected as we are supplying the same data points:

$$x_1 = 1$$

$$x_2 = -0.420000000000000$$

$$x_3 = 0.049999999999999$$

$$x_4 = -0.416666666666663$$

$$x_5 = 0.416666666666662$$

$$x_6 = -2.604166666666665$$

Using these coefficients as above, the interpolated value for $t = 0.78$:

$$f(0.78) \cong p_{n-1}(0.78) = 0.637895075000000$$

Question 2 – Conclusion

Method	f(0.78)
Cubic Splines	0.641964285714286
Lagrange Interpolation	0.637895075000000
Newton Interpolation	0.637895075000000

As in question 1, the results obtained by the Lagrange and the Newton Interpolation methods are identical, as they both compute the same unique polynomial. The Cubic Splines result is a little off compared to the other two results, but still falls within the acceptable margin of error

Question 3 - Extrapolation

This question on the assignment required us to extrapolate what the value of the function, given the data set, could be at the value of $t = 3.0$. To do this, I simply used the data to the right of the discontinuity and used the last three data points to compute their cubic splines. The data points I used were $(2, 0.316), (2.2, 0.342)\ and\ (2.4, 0.368)$. I then used the cubic spline defined over the interval $[2.2\ 2.4]$ and obtained the following equation:

$$p(t) = 0.055999999999975 + 0.130000000000027t - (8.666536875900679e - 15)t^2 + (7.222114063250566e - 16)t^3$$

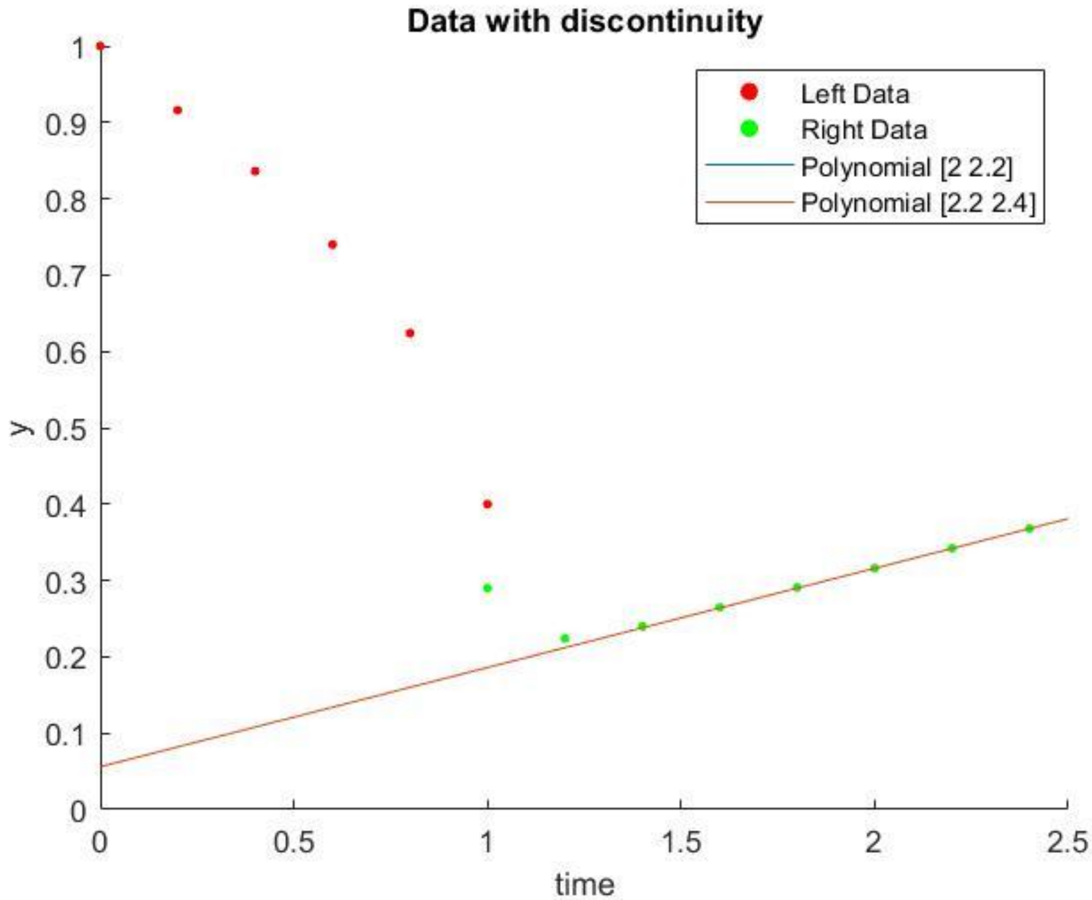

Figure 6: Cubic Spline of extrapolation section

As this function is roughly linear (the square and cubic coefficients being to the order of 10^{-15} and 10^{-16} respectively), it can be expected that the data points will tend to follow the same trend. Such, when using the above polynomial to extrapolate what the value of the data at $t = 3$, I obtain the following result:

$$f(3) \cong p(3) = 0.445999999999999$$