Ahsanullah University of Science & Technology

Department of Electrical & Electronic Engineering

Project

Course No : EEE3110

Course Name : Numerical Technique Laboratory

Project Name : Interpolation

Project Group : 02

Submitted by

SL	ID	Name
01	190205182	Rifat Bin Arif
02	190205183	Md. Ekramul Hasan Sharon
03	190205184	Hashin Israq
04	190205185	Anamika Rahman Prity
05	190205186	Abdullah-Al- Noman

Topic: Linear Interpolation.

Interpolation:

In the mathematical sub field of numerical analysis, interpolation is a method of constructing new data points from a discrete set of known data points.

In engineering and science, one often has a number of data points, as obtained by sampling or some experiment, and tries to construct a function which closely fits those data points. This is called curve fitting. Interpolation is a specific case of curve fitting, in which the function must go exactly through the data points.

Theory:

A sequence of n distinct numbers X_k called nodes and for each X_k a second number Y_k, we are looking for a function f so that

$$\mathbf{f(X_k)=Y_k,\ k=1,2,3,4\ldots n}$$

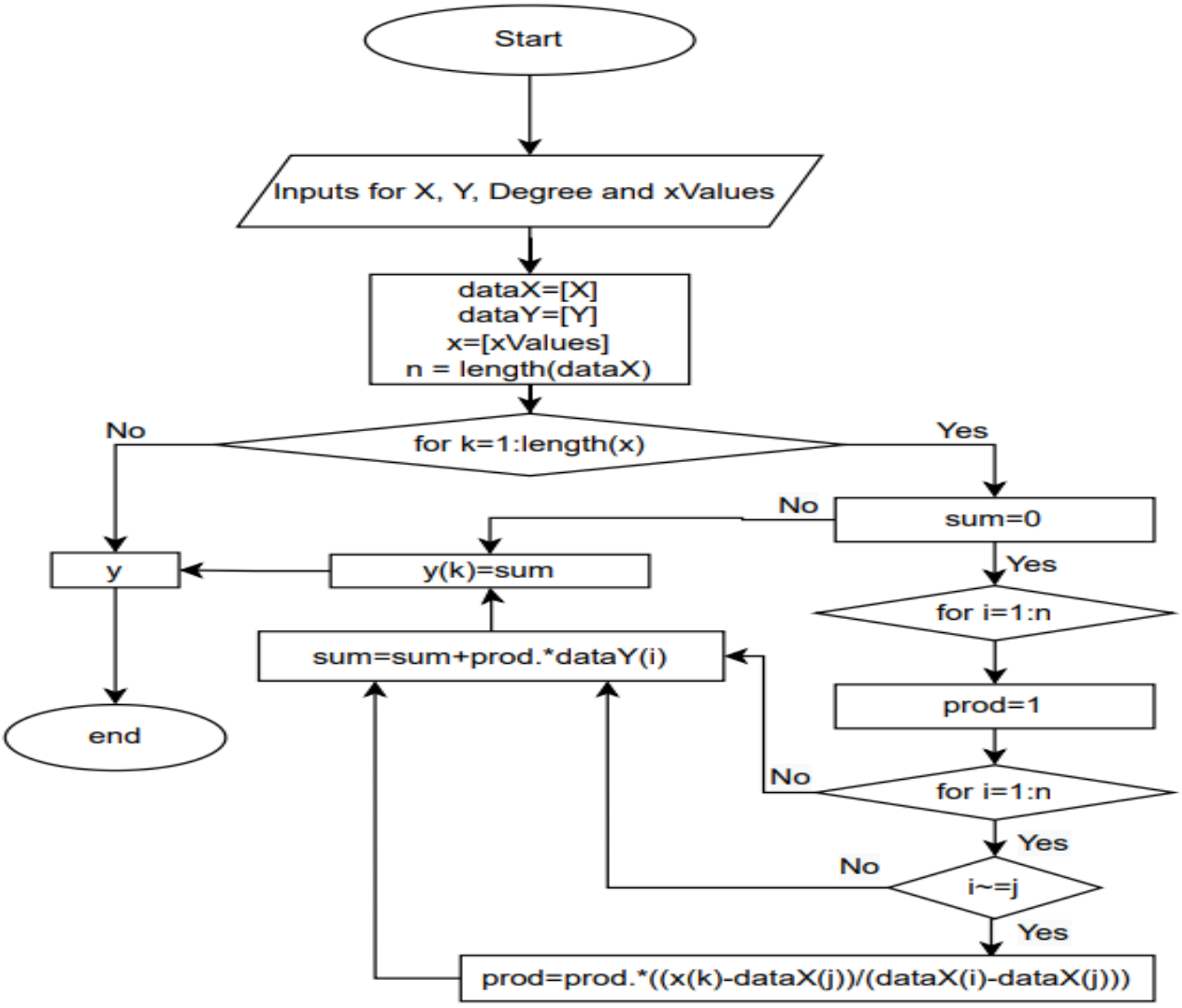
A pair X_k, Y_k is called a data point and f is called the interpolant for the data points.

Generally, linear interpolation takes two data points, say (X_a, Y_a) and (X_b, Y_b), and the interpolant is given by

$$F(x)=\frac{x-xb}{xa-xb}ya+\frac{x-xa}{xa-xb}yb$$

This formula can be interpreted as a weighted mean.

Algorithm:


Start
Inputs for X, Y, Degree and xValues
dataX=[X]
dataY=[Y]
x=[xValues]
n = length(dataX)
for k=1:length(x)
No
Yes
y
end
y(k)=sum
No
sum=0
Yes
for i=1:n
sum=sum+prod.*dataY(i)
prod=1
No
for i=1:n
Yes
No
i~=j
Yes
prod=prod.*((x(k)-dataX(j))/(dataX(i)-dataX(j)))

Exmaple: For example a table like this, which gives some values of an unknown function f. The data are given in the table:

X	F(X)
0	0
1	0.8415
2	0.9093
3	0.1411
4	-0.7568
5	-0.9589
6	-0.2794
7	0.2858
8	0.7853

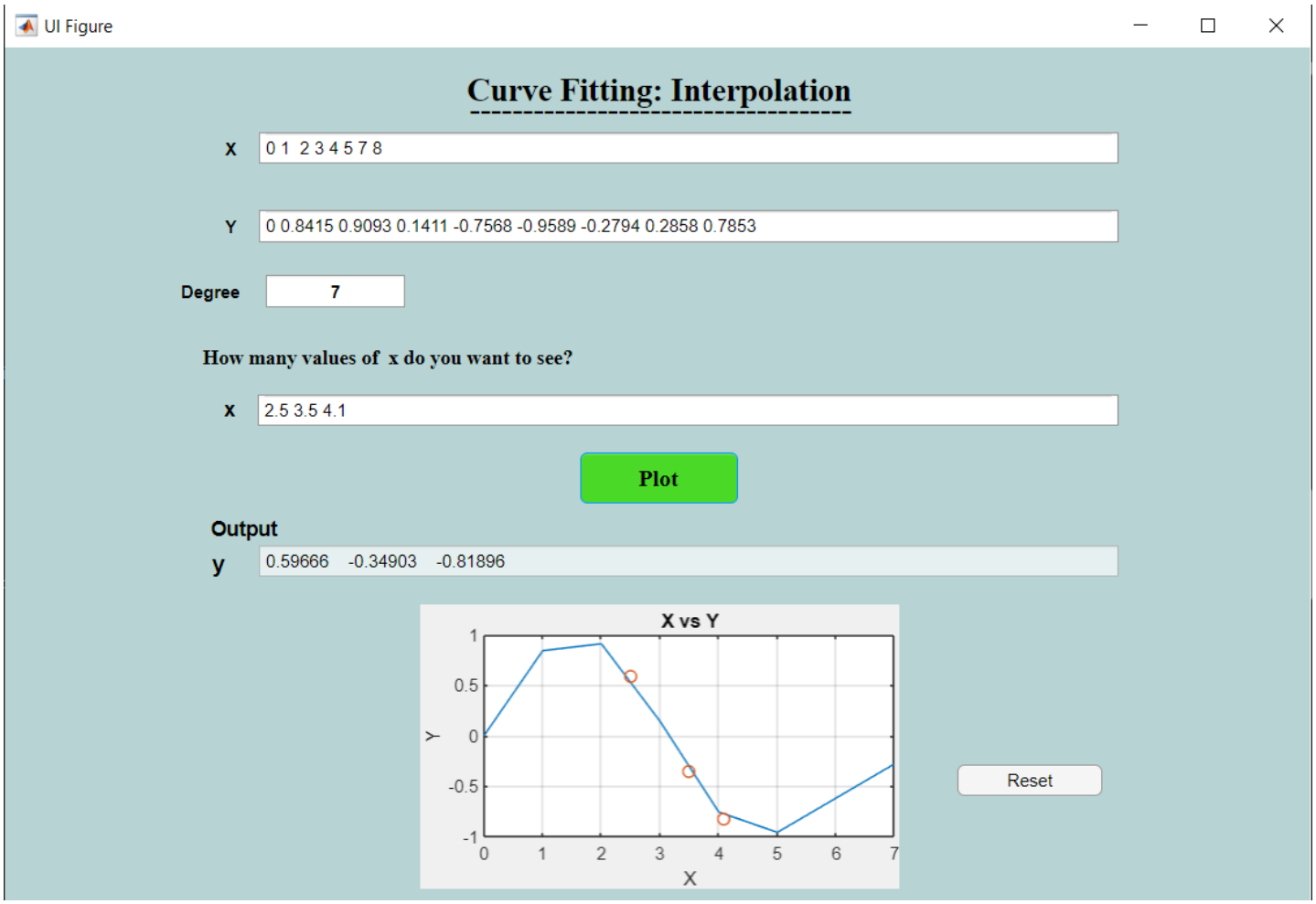
x=2.5, 3.5, 4.5

Output: Using Interpolation App

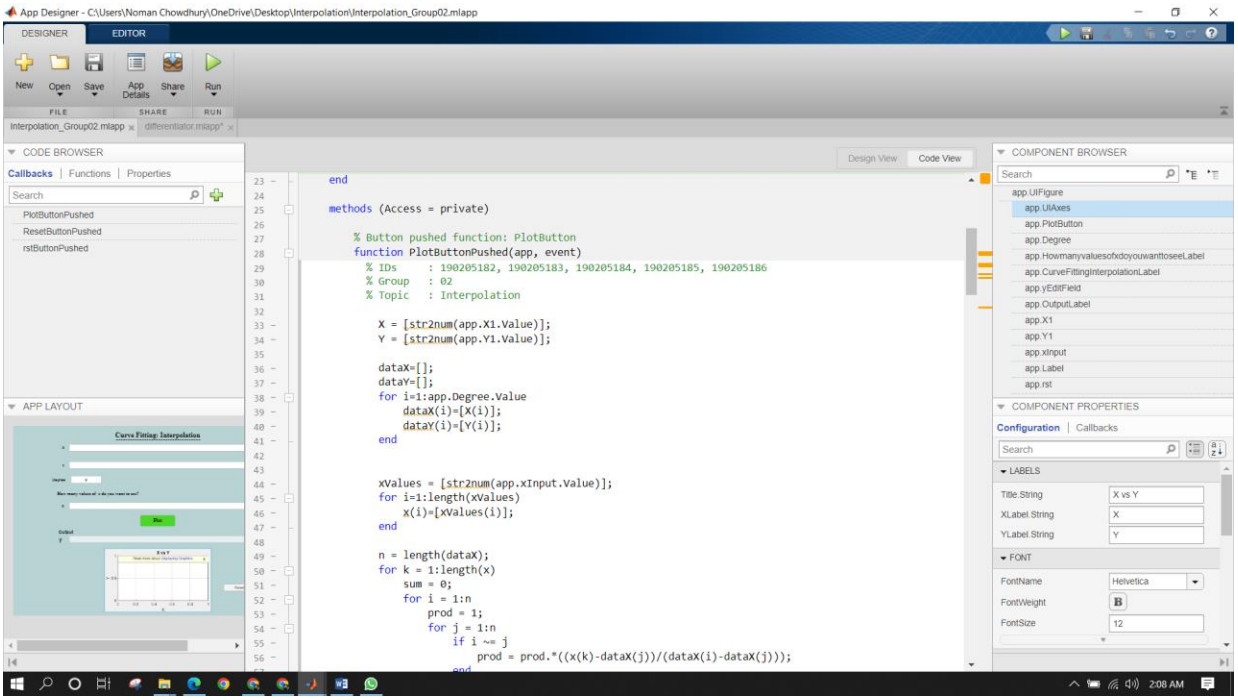
Plot:

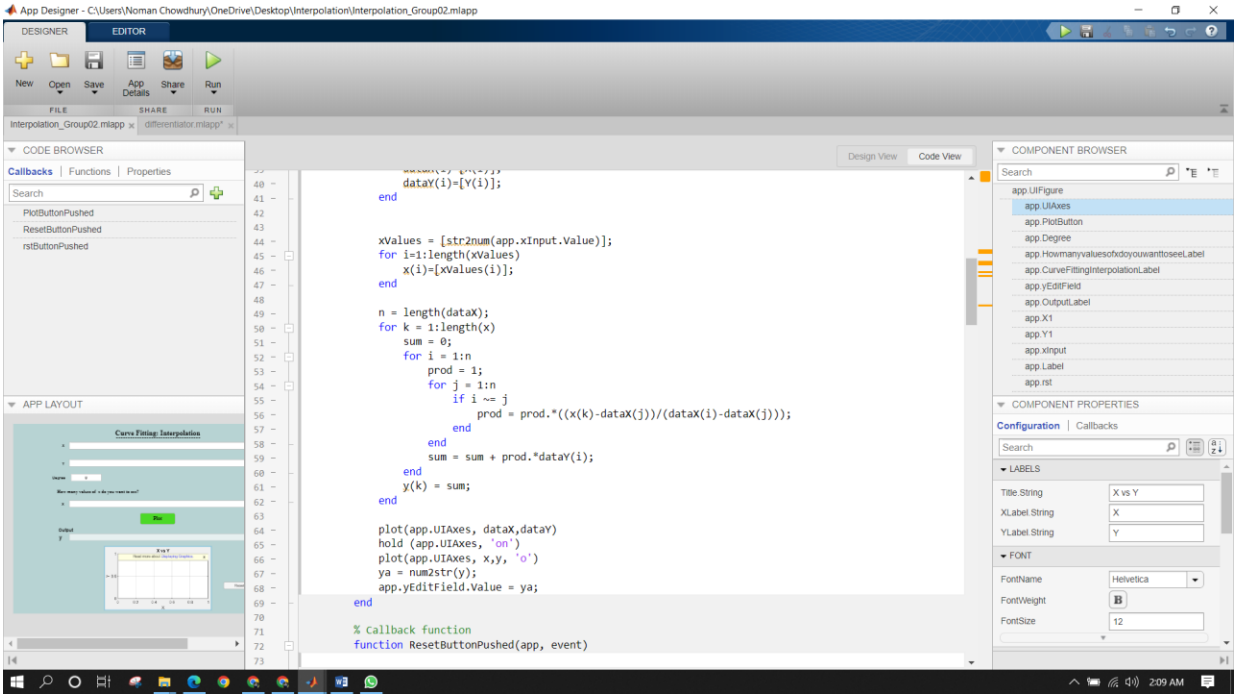
Using Interpolation App,

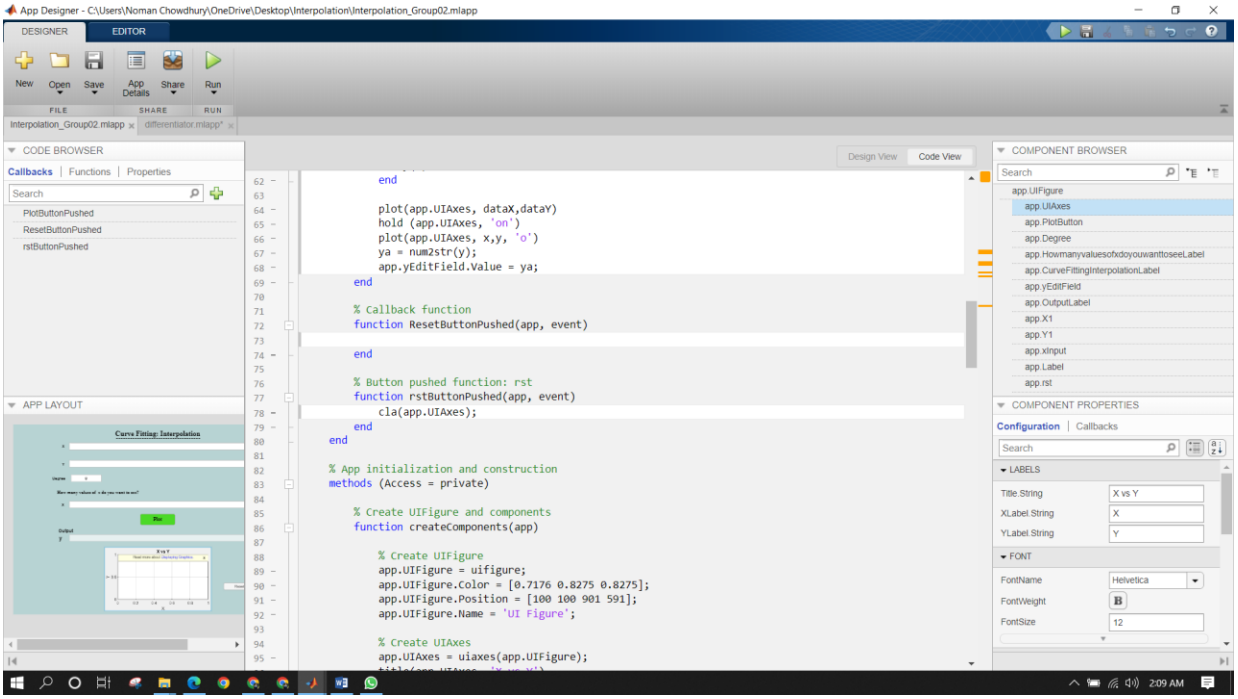
When order=7



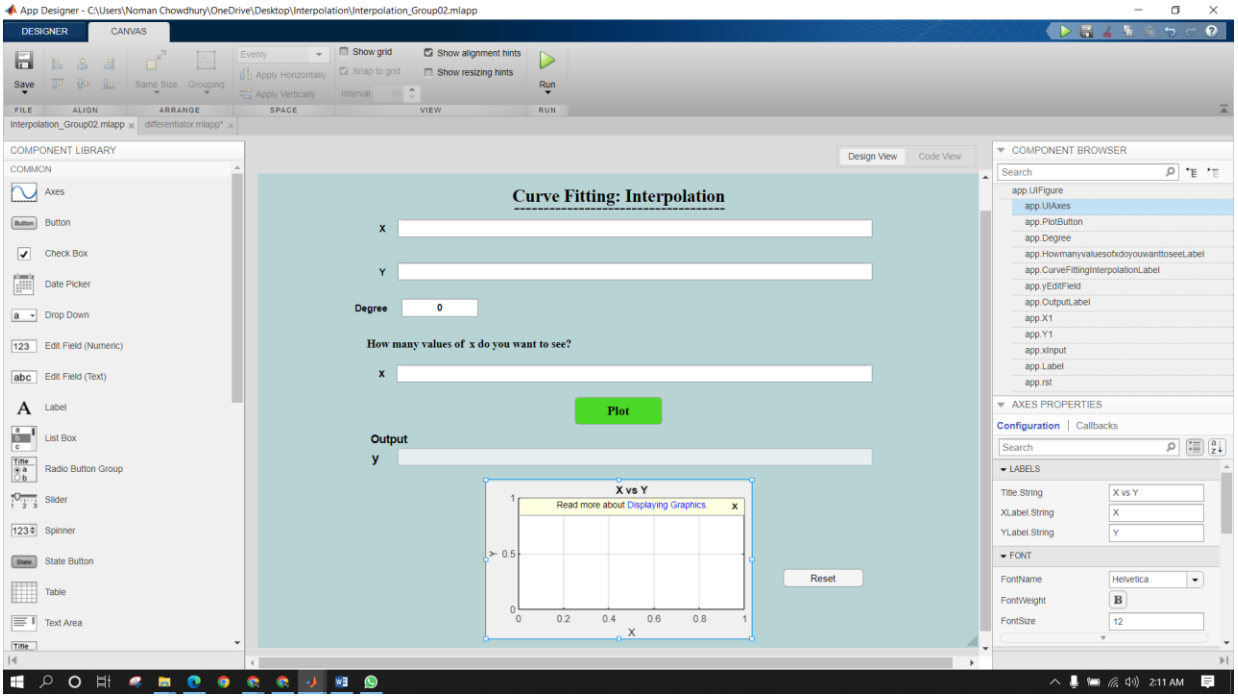
Code View:


App Designer - C:\Users\Noman Chowdhury\OneDrive\Desktop\Interpolation\Interpolation_Group02.mlapp
DESIGNER
EDITOR
New
Open
Save
App Details
Share
Run
FILE
SHARE
RUN
Interpolation_Group02.mlapp
differentiator.mlapp*
CODE BROWSER
Callbacks | Functions | Properties
Search
PlotButtonPushed
ResetButtonPushed
rstButtonPushed
APP LAYOUT
Curve Fitting: Interpolation
Design View
Code View
COMPONENT BROWSER
app.UIFigure
app.UIAxes
app.PlotButton
app.Degree
app.HowmanyvaluesofxdoyouwanttoseeLabel
app.CurveFittingInterpolationLabel
app.yEditField
app.OutputLabel
app.X1
app.Y1
app.xInput
app.Label
app.rst
COMPONENT PROPERTIES
Configuration | Callbacks
LABELS
Title.String
X vs Y
XLabel.String
X
YLabel.String
Y
FONT
FontName
Helvetica
FontWeight
FontSize
12
2:08 AM


App Designer - C:\Users\Noman Chowdhury\OneDrive\Desktop\Interpolation\Interpolation_Group02.mlapp
DESIGNER
EDITOR
New
Open
Save
App Details
Share
Run
FILE
SHARE
RUN
Interpolation_Group02.mlapp
differentiator.mlapp*
CODE BROWSER
Callbacks | Functions | Properties
Search
PlotButtonPushed
ResetButtonPushed
rstButtonPushed
APP LAYOUT
Curve Fitting: Interpolation
Design View
Code View
COMPONENT BROWSER
app.UIFigure
app.UIAxes
app.PlotButton
app.Degree
app.HowmanyvaluesofxdoyouwanttoseeLabel
app.CurveFittingInterpolationLabel
app.yEditField
app.OutputLabel
app.X1
app.Y1
app.xInput
app.Label
app.rst
COMPONENT PROPERTIES
Configuration | Callbacks
LABELS
Title.String
X vs Y
XLabel.String
X
YLabel.String
Y
FONT
FontName
Helvetica
FontWeight
FontSize
12
2:09 AM


App Designer - C:\Users\Noman Chowdhury\OneDrive\Desktop\Interpolation\Interpolation_Group02.mlapp
DESIGNER
EDITOR
New
Open
Save
App Details
Share
Run
CODE BROWSER
Callbacks | Functions | Properties
PlotButtonPushed
ResetButtonPushed
rstButtonPushed
APP LAYOUT
Design View
Code View
COMPONENT BROWSER
app.UIFigure
app.UIAxes
app.PlotButton
app.Degree
app.HowmanyvaluesofxdoyouwanttoseeLabel
app.CurveFittingInterpolationLabel
app.yEditField
app.OutputLabel
app.X1
app.Y1
app.xInput
app.Label
app.rst
COMPONENT PROPERTIES
Configuration | Callbacks
LABELS
Title.String
X vs Y
XLabel.String
X
YLabel.String
Y
FONT
FontName
Helvetica
FontWeight
FontSize
12
2:09 AM

Design View:


App Designer - C:\Users\Noman Chowdhury\OneDrive\Desktop\Interpolation\Interpolation_Group02.mlapp
DESIGNER
CANVAS
COMPONENT LIBRARY
COMMON
Axes
Button
Check Box
Date Picker
Drop Down
Edit Field (Numeric)
Edit Field (Text)
Label
List Box
Radio Button Group
Slider
Spinner
State Button
Table
Text Area
Curve Fitting: Interpolation
x
Y
Degree
0
How many values of x do you want to see?
Plot
Output
y
X vs Y
Read more about Displaying Graphics
Reset
COMPONENT BROWSER
AXES PROPERTIES
Configuration | Callbacks
LABELS
Title.String
X vs Y
XLabel.String
X
YLabel.String
Y
FONT
FontName
Helvetica
FontWeight
FontSize
12
2:11 AM

MATLAB CODE:

```
% Button pushed function: PlotButton
        function PlotButtonPushed(app, event)
          % IDs     : 190205182, 190205183, 190205184, 190205185, 190205186
          % Group   : 02
          % Topic   : Interpolation

            X = [str2num(app.X1.Value)];
            Y = [str2num(app.Y1.Value)];

            dataX=[];
            dataY=[];
            for i=1:app.Degree.Value
                dataX(i)=[X(i)];
                dataY(i)=[Y(i)];
            end



            xValues = [str2num(app.xInput.Value)];
            for i=1:length(xValues)
                x(i)=[xValues(i)];
            end

            n = length(dataX);
            for k = 1:length(x)
                sum = 0;
                for i = 1:n
                    prod = 1;
                    for j = 1:n
                        if i ~= j
                            prod = prod.*((x(k)-dataX(j))/(dataX(i)-dataX(j)));
                        end
                    end
                    sum = sum + prod.*dataY(i);
```

```
            end
            y(k) = sum;
        end

        plot(app.UIAxes, dataX,dataY)
        hold (app.UIAxes, 'on')
        plot(app.UIAxes, x,y, 'o')
        ya = num2str(y);
        app.yEditField.Value = ya;
    end
    % Callback function
    function ResetButtonPushed(app, event)

    end
    % Button pushed function: rst
    function rstButtonPushed(app, event)
        cla(app.UIAxes);
    end
end
```

Discussion: Linear interpolation is useful when looking for a value between given data points. It can be considered as “filling in the gaps” in a table of data. The strategy for linear interpolation is to use a straight line to connect the known data points on either side of the unknown point. Linear interpolation is often not accurate for non-linear data. If the points in the data

set to change by a large amount, linear interpolation may not give a good estimate.

Interpolation can be described as guessing data points that fall within the range of the data you are already provided with that is between your existing data points.

Interpolation can be defined as an estimation of a value within two known values in a given sequence of values. When graphical data contains a gap, but the data is available on either side of the gap or at a few specific points within the gap, Interpolation is a method that allows us to estimate the values within the gap. So, it can be understood that the formula for Interpolation is a method of curve fitting using the linear polynomials and hence constructing new data points within the given range of a discrete set of known data points (the data points). Linear Interpolation can be used since very early antiquity for filling the unknown values in any table. As we know, Interpolation can be defined as a process of using the points with known values or the given sample points to estimate values at other unknown points. Interpolation Methods can be used to predict unknown values for any geographic point data, for example, elevation, rainfall, chemical concentrations,

noise levels, and so on. Interpolation is widely used in Statistical models for commercial and Mathematical research, as it helps predict the future probable points in data analysis. From the obtained sets, one can determine where the general consistent trend will lead the price (of a product or service), potential yield (and growth for a company), or gain insights into the stock market under a given market dynamic. Financial analysts have consistently exploited this method to reach logical conclusions in the bonds market and financial world.

Interpolation is a technique for adding new data points within a range of a set of known data points. We can use interpolation to fill in missing data, smooth existing data, make predictions, and more. Interpolation in MATLAB is divided into techniques for data points on a grid and scattered data points.

The concept of Interpolation is used to simplify complicated functions by sampling any given data points and interpolating these data points using a simpler function. Commonly.

While Interpolation is known to solve a lot of Mathematical and Statistical problems, it does have certain drawbacks and criticisms. One such drawback is that although the method of Interpolation is simple and has been known to Mathematicians

and people in general, for a long time, it has been known to lack the necessary accuracy and precision. People use these methods of Interpolation for the fairly unpredictable stock markets, in solving data related to security analysis, for determining volatility of the highly unpredictable public-traded shares and bonds, and this overpowering mass of data makes the employment of Interpolation unreasonable as it can lead to many faulty predictions. More often than not, the use of Interpolation in regression analysis, in this way leads to the yielding of an “error term”, that is obtaining a set of values that do not represent the factual relationship between the variables most crucial for successful prediction. Interpolation must be employed for simple predictions such as determining the interest rate or value of any variable for which the data point is missing.