Data Regression

Lecture 9 – Monday April 10, 2017

Outline

- Data Regression
- Matlab Interpolation and Curve Fitting
- Example

Outline

- **Data Regression**
- Matlab Interpolation and Curve Fitting
- Example

Data Regression

Regression analysis is a statistical process for **estimating the relationships** among variables.

Regression models involve the following variables:

- The **independent variables**, X.
- The **dependent variable**, Y.
- The **unknown parameters**, denoted as β, which may represent a scalar or a vector.

A **regression model** relates Y to a function of X and β.

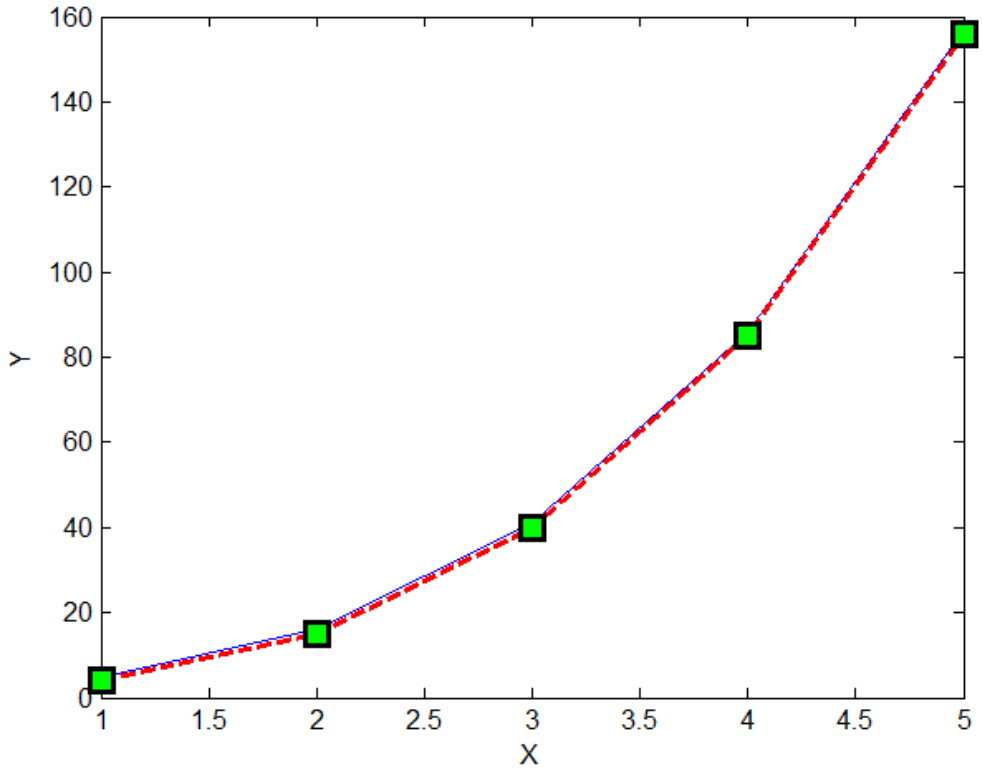
$$Y \approx f(X,\beta)$$

Data Regression

- In the curve-fitting problem, we would like to **fit a polynomial** to a given set of data points.
- Given the **set of data points** in the shown table and assuming we want to fit a **3rd degree polynomial** to these data points.

$$y = ax^3 + bx^2 + cx + d$$

x	*y*
1	5
2	16
3	41
4	86
5	157



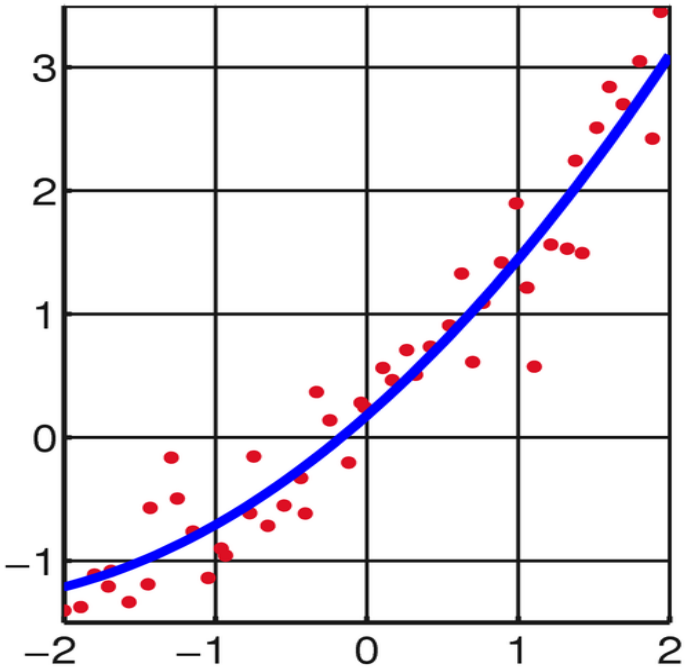
Data Regression

Least squares of errors

$$\min \ f = \min \sum_{i=1}^{\#\text{of points}} (y - y_{desired})^2$$

Note: In curve-fitting, the best fit in the **least-squares** sense minimizes the sum of squared residuals, a residual being the difference between an observed value and the fitted value provided by a model.

Source: Wikipedia

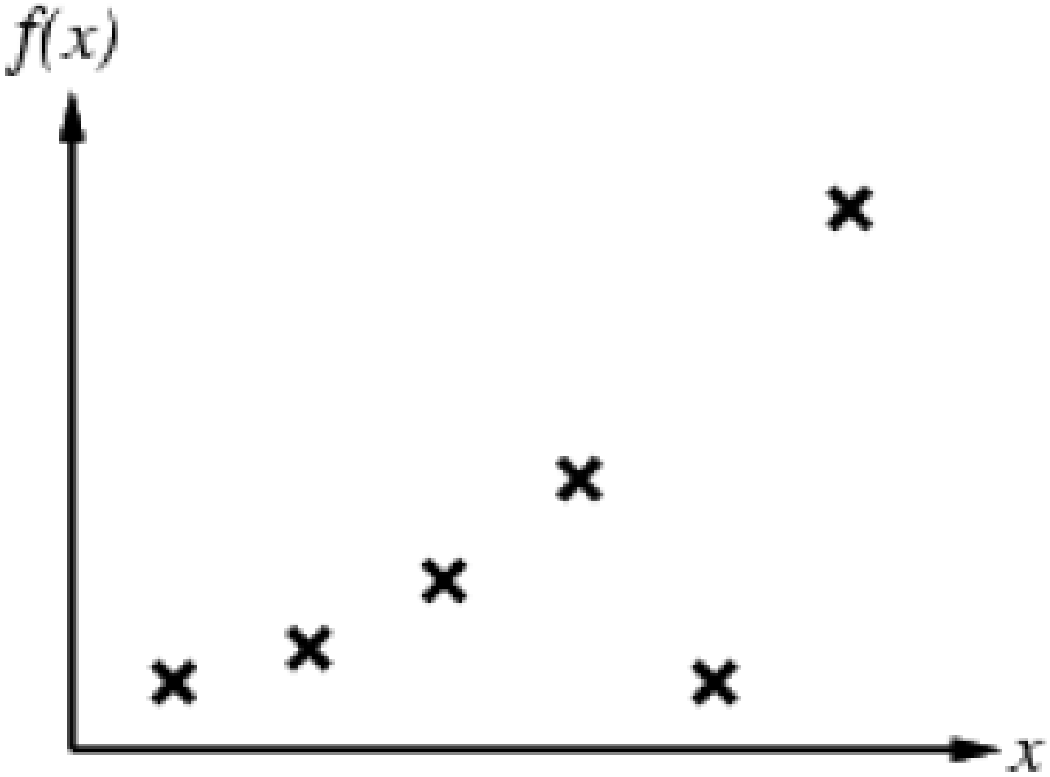


Data Regression

- **Curve fitting (regression)**

  Given: x = Input data point (a training example)

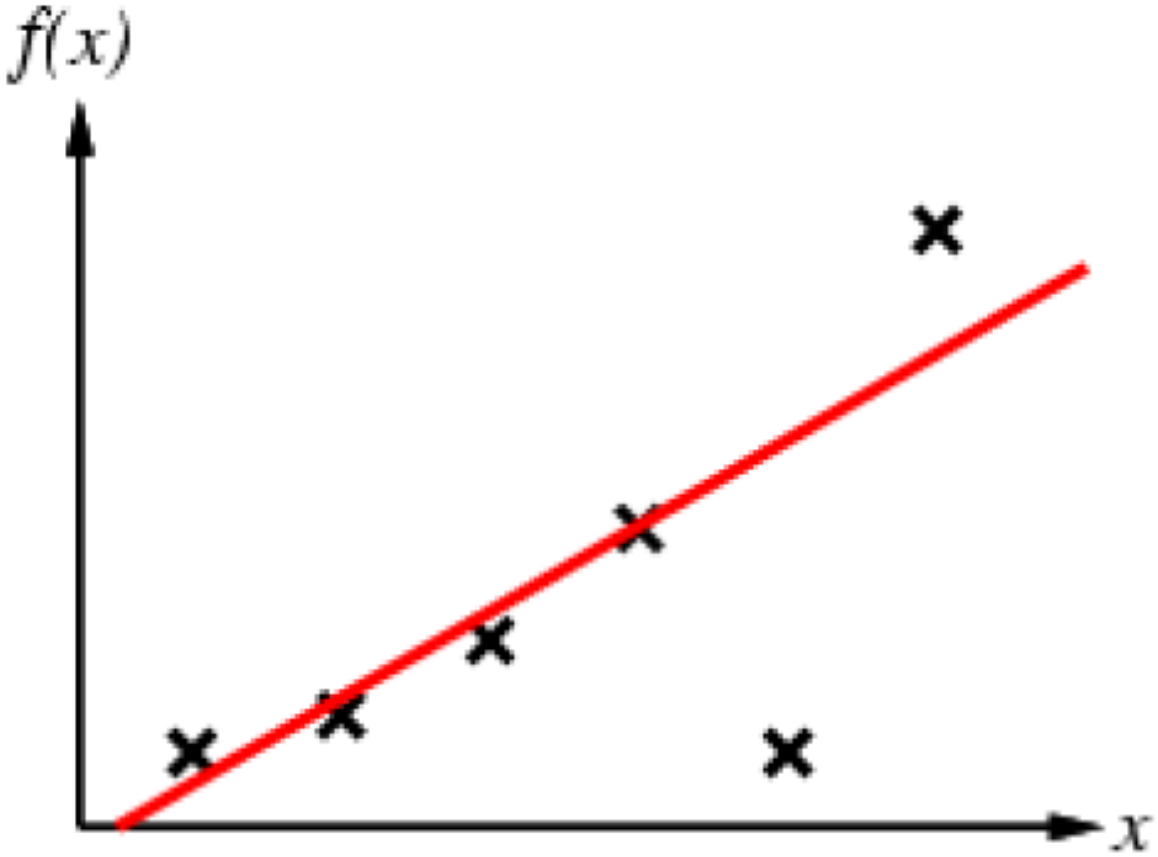
  Required: $y \approx f(x)$



Data Regression

- **Curve fitting (regression)**

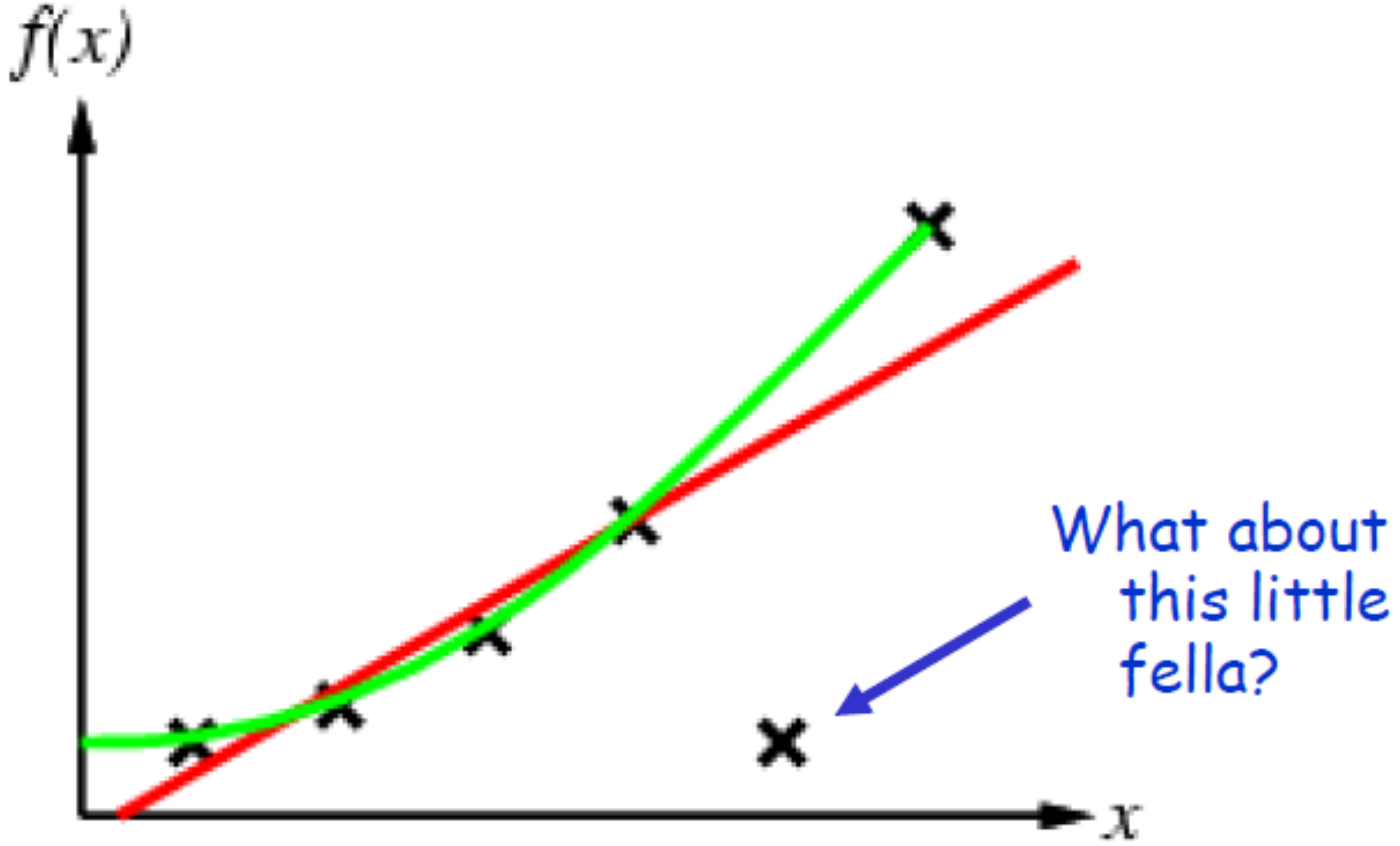
y = Straight line?



Data Regression

- **Curve fitting (regression)**

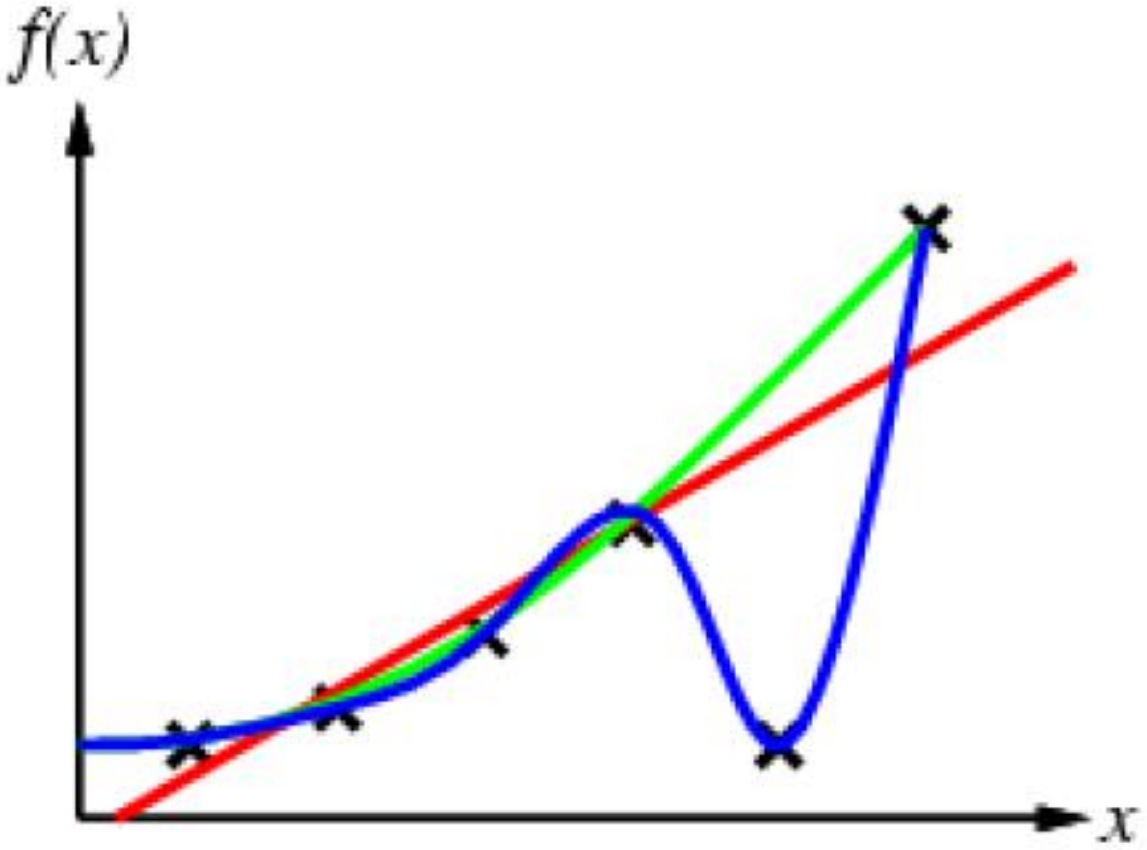
y = quadratic function?



Data Regression

- **Curve fitting (regression)**

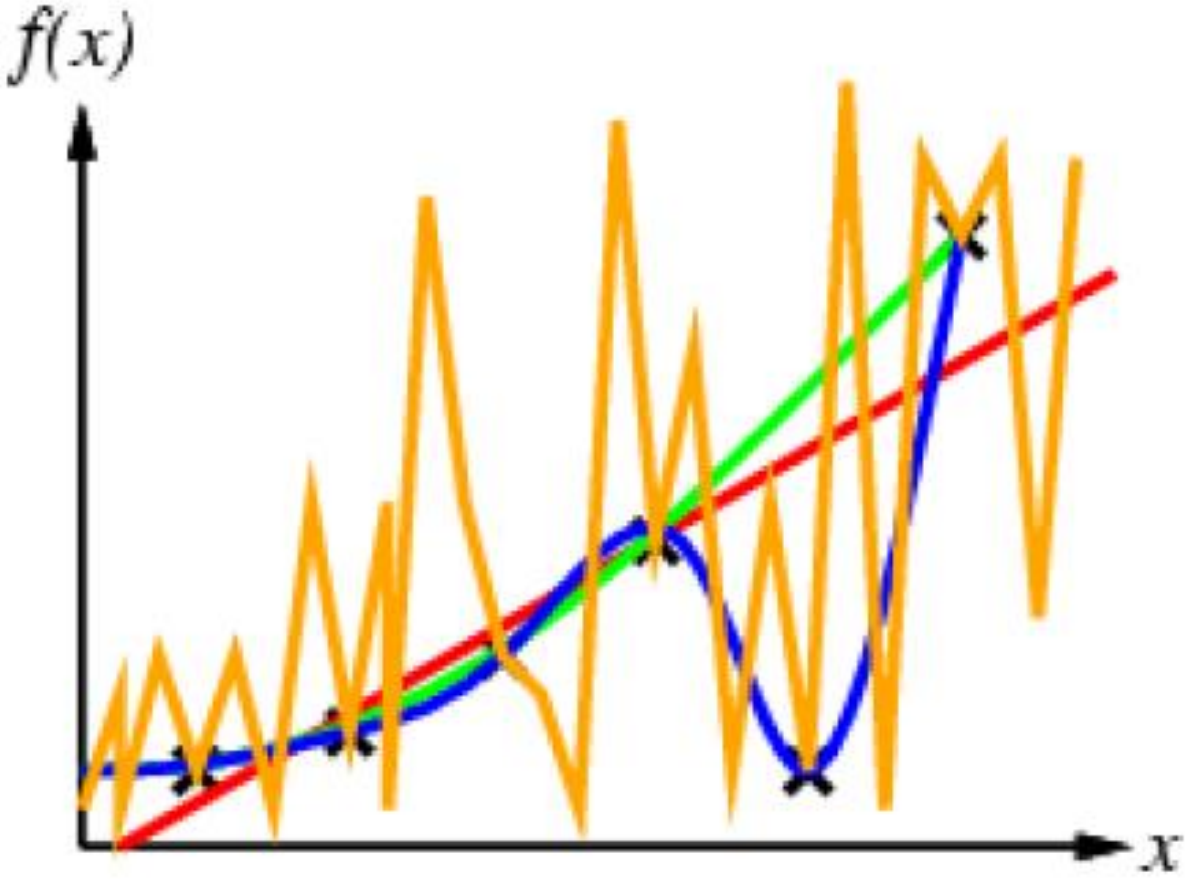
What about a function that satisfies all!



Data Regression

- **Curve fitting (regression)**

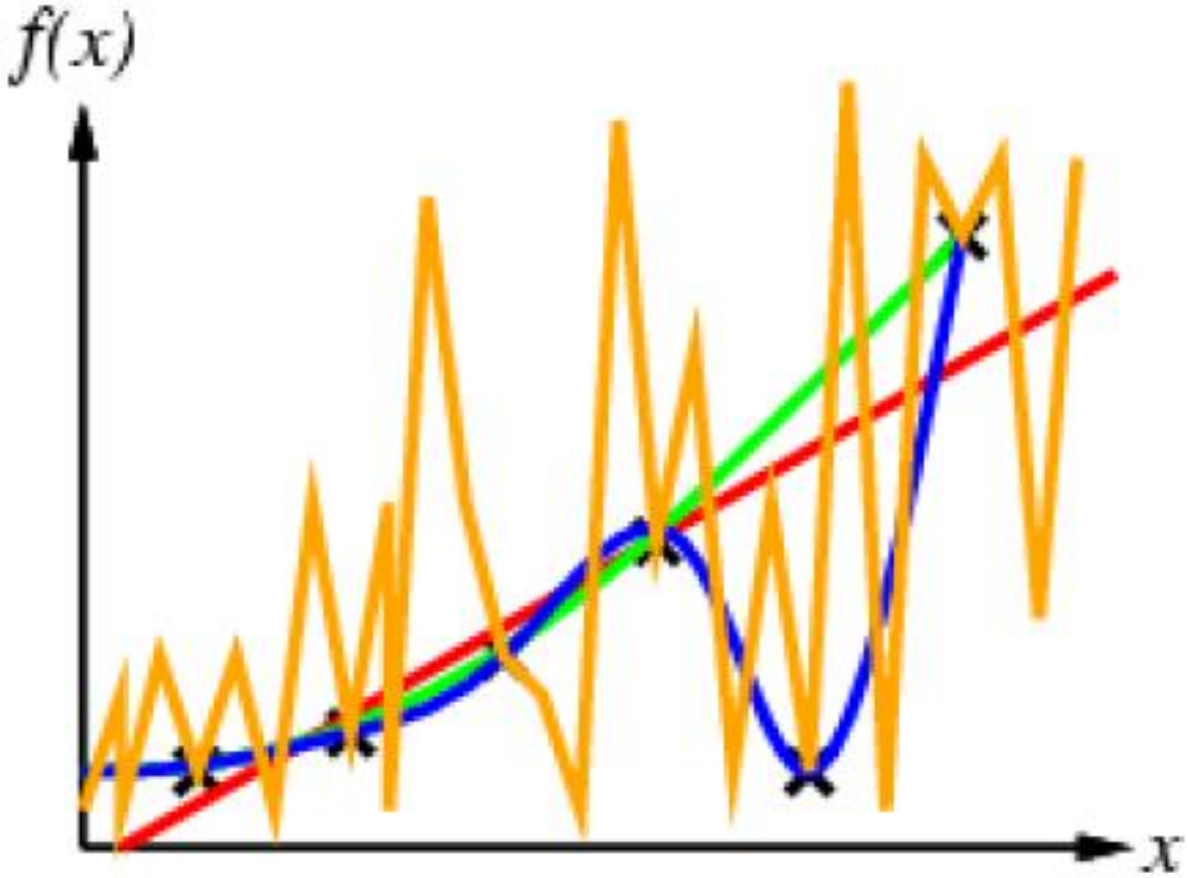
But so does this one…



Data Regression

- **Curve fitting (regression)**

But so does this one…

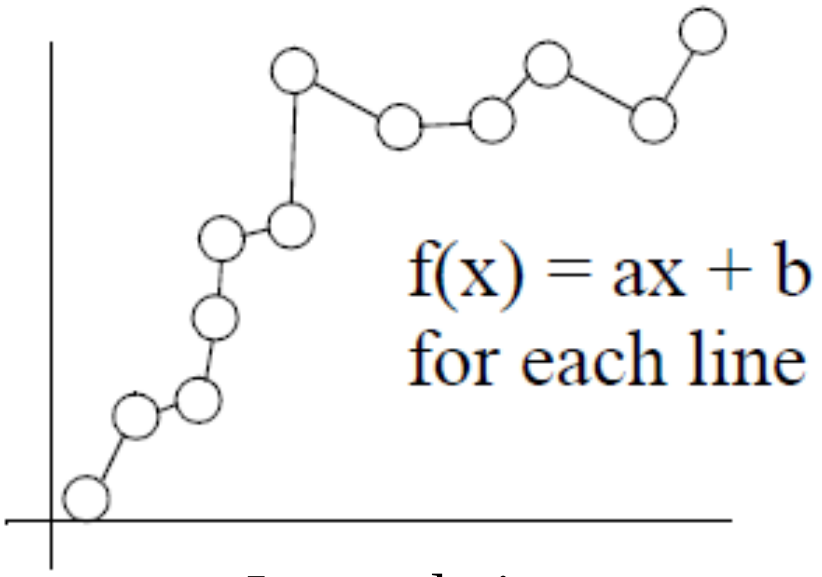


Outline

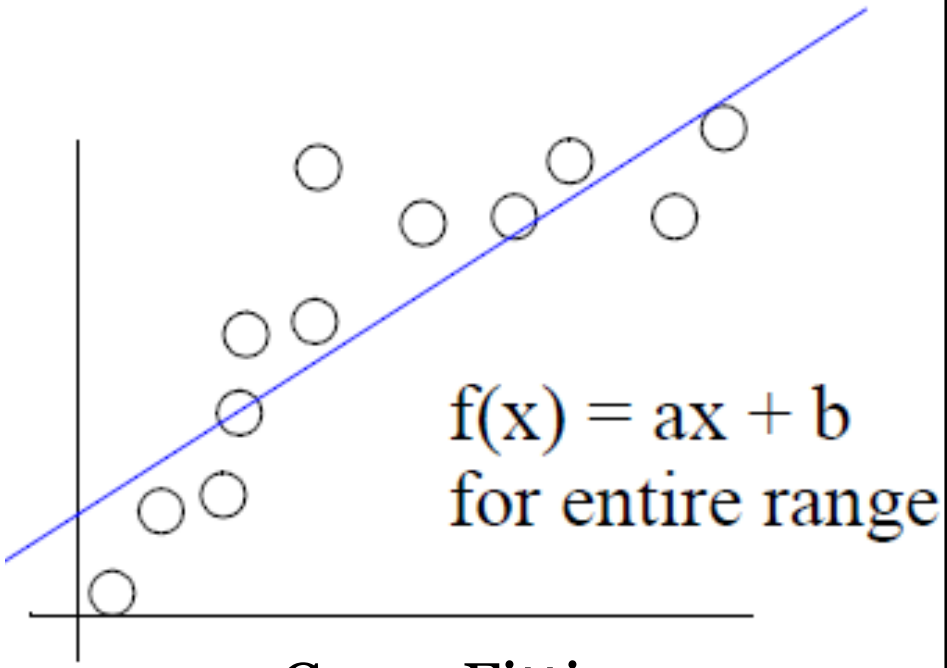
- Data Regression
- **Matlab Interpolation and Curve Fitting**
- Example

Matlab Interpolation and Curve Fitting

- **Interpolation and Curve fitting**



Interpolation



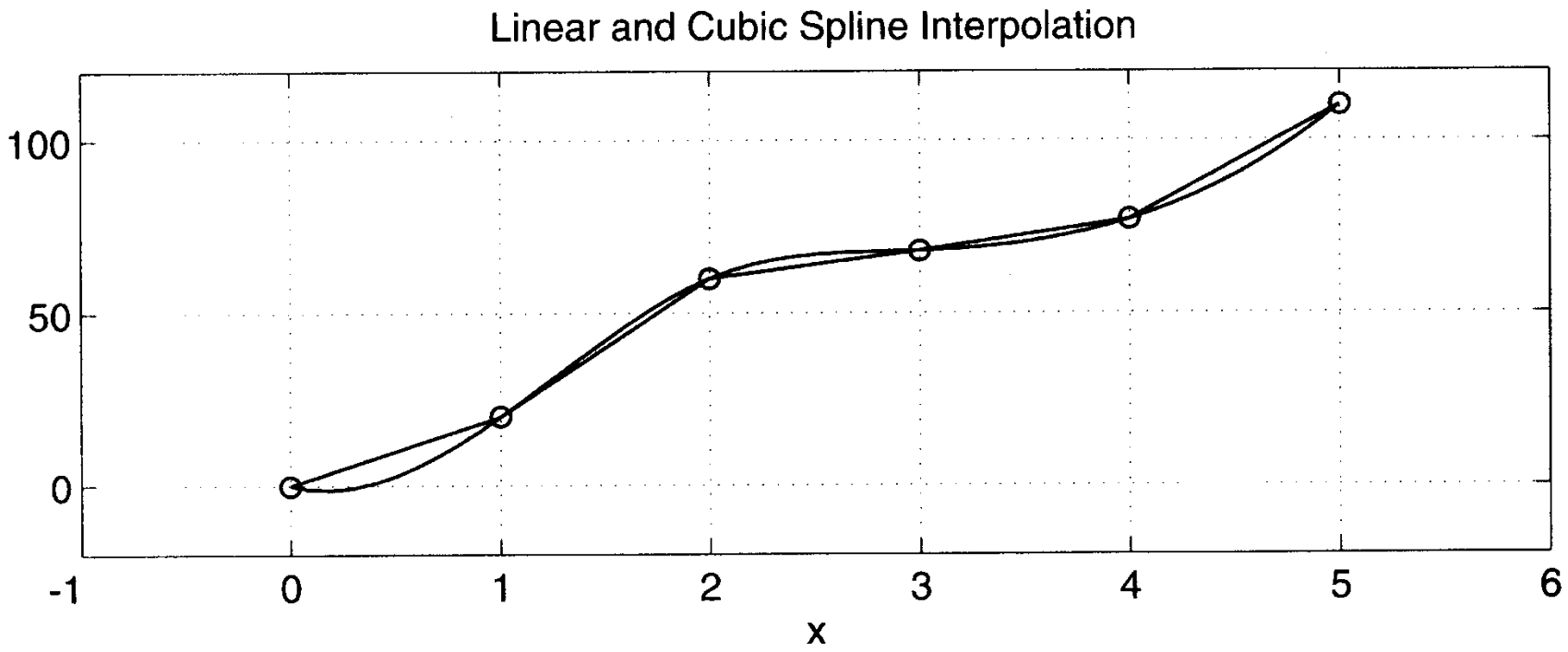
Curve Fitting

If data is reliable, we can plot it and connect the dots This is piece-wise, linear interpolation

Capturing the trend in the data by assigning a single function across the entire range

Matlab Interpolation and Curve Fitting

We present two types of interpolation- linear interpolation and cubic-spline interpolation.



Matlab Interpolation and Curve Fitting

- **Linear Interpolation**

One of the most common techniques for estimating data between two given data points is linear interpolation.

interp1(x,y,x_new)

Returns a vector of the size of y, which contains the interpolated y values that correspond to x_new using linear interpolation.

interp1(x,y,x_new, 'linear')

Returns a vector of the size of y, which contains the interpolated y values that correspond to x_new using linear interpolation.

Matlab Interpolation and Curve Fitting

- **Linear Interpolation**

Example: Given the following temperature measurements taken from the cylinder head in a new engine that is being tested for possible use in a race car.

Times, s	Temperature, F
0	0
1	20
2	60
3	68
4	77
5	110

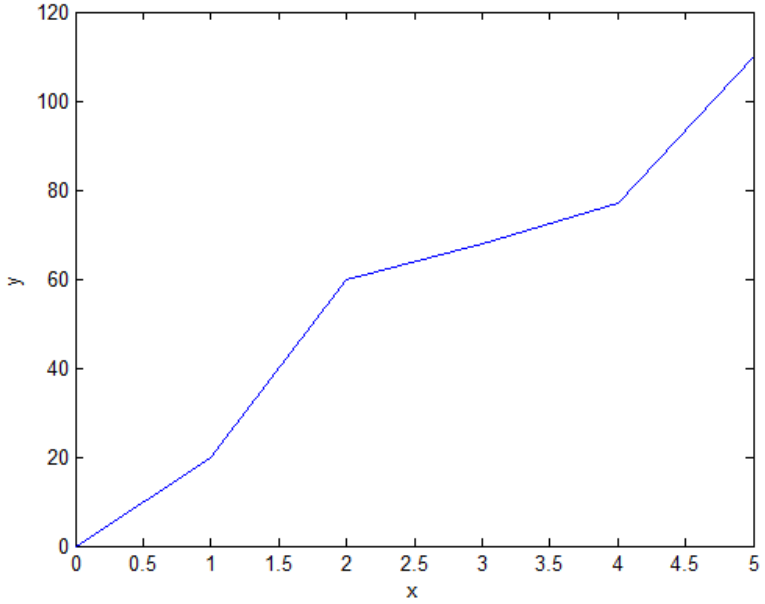
Matlab Interpolation and Curve Fitting

- **Linear Interpolation**

```
x=0:5;
y=[0,20,60,68,77,110];


y1=interp1(x,y,2.6);
y2=interp1(x,y,4.9);
```

Times, s	Temperature, F
0	0
1	20
2	60
3	68
4	77
5	110



Matlab Interpolation and Curve Fitting

- **Cubic-spline Interpolation**

  A cubic-spline is a smooth curve constructed to go through a set of points.

  interpl(x,y,x_new,'spline')

  Returns a vector which contains the interpolated y values that correspond to x_new using cubic-spline interpolation.

Example:

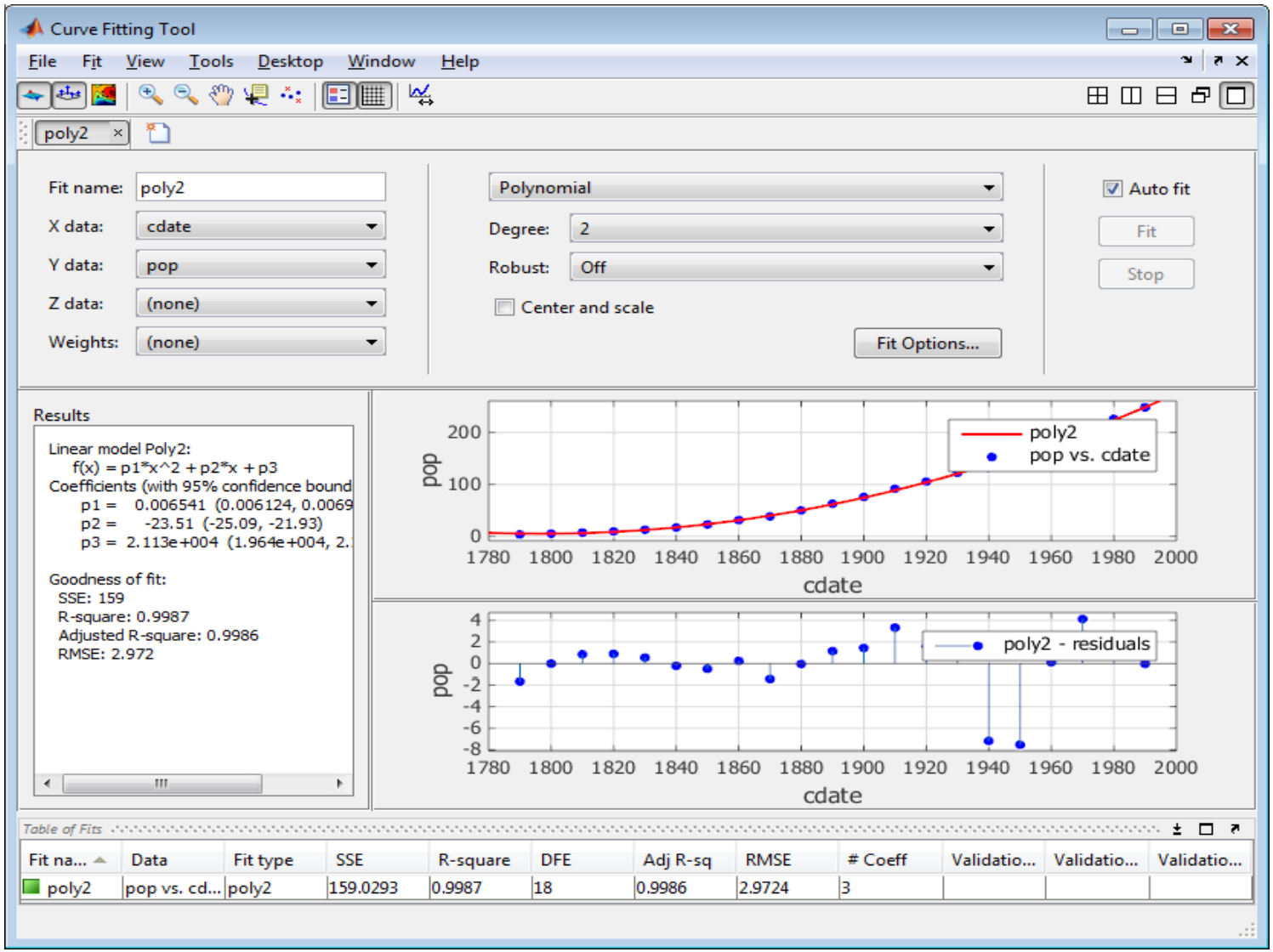
```
x=0:5;
y=[0,20,60,68,77,110];
temp1=interp1(x,y,2.6,'spline');
```

Try this:

```
temp2=interp1(x,y,[2.6,4.9],'spline');
```

Matlab Interpolation and Curve Fitting

- **Curve Fitting Tool**



Matlab Interpolation and Curve Fitting

- **Curve Fitting Tool**

  Curve Fitting Toolbox software allows you to work in two different environments:

  - An interactive environment, with the Curve Fitting app and the Spline Tool
  - A programmatic environment that allows you to write object-oriented MATLAB code using curve and surface fitting methods.

Matlab Interpolation and Curve Fitting

- **Curve Fitting Tool**

cftool	opens Curve Fitting app or brings focus to the app if it is already open.
cftool(x, y)	creates a curve fit to x input and y output. x and y must be numeric, have two or more elements, and have the same number of elements.
cftool(x, y, z)	creates a surface fit to x and y inputs and z output.
cftool(x, y, [], w)	creates a curve fit with weights w. w must be numeric and have the same number of elements as x and y.
cftool(x, y, z, w)	creates a surface fit with weights w. w must be numeric and have the same number of elements as z
cftool(filename)	loads the Curve Fitting session in filename into Curve Fitting app.

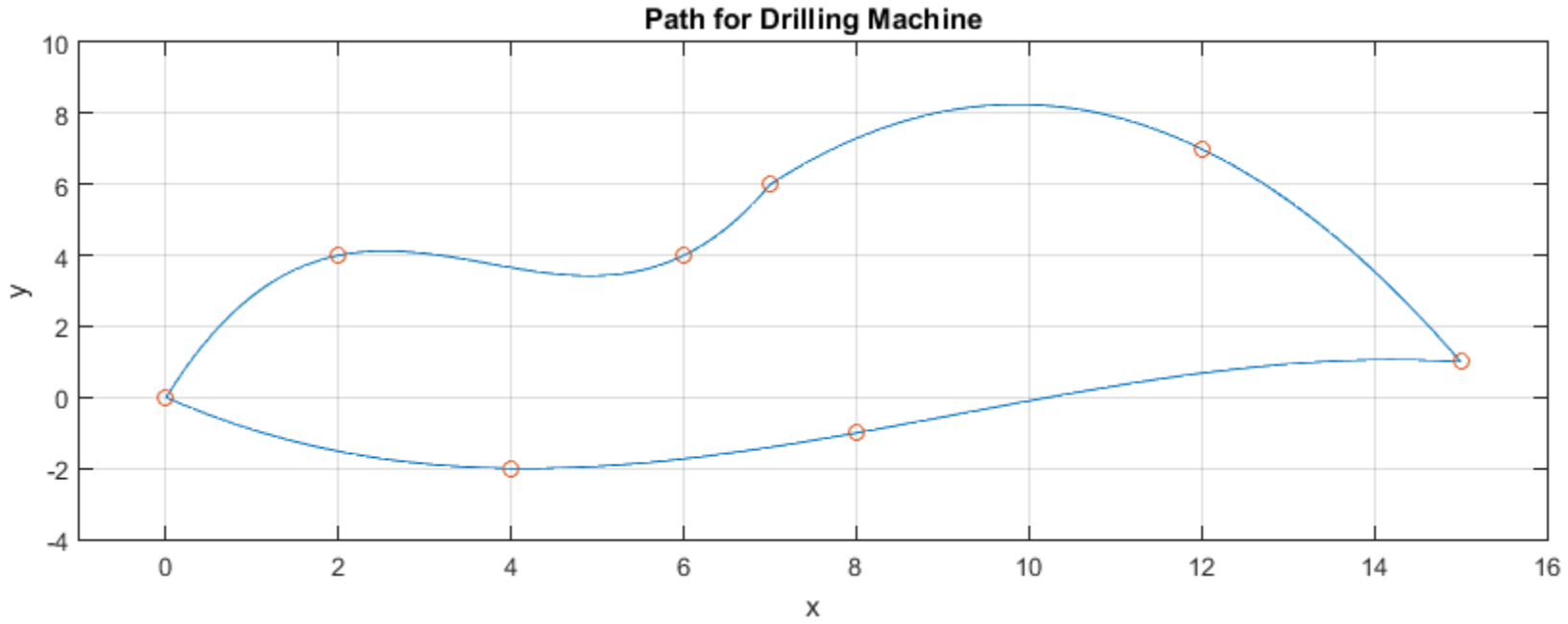
Outline

- Data Regression
- Matlab Interpolation and Curve Fitting
- **Example**

Example

- **Inputs/Outputs Description**

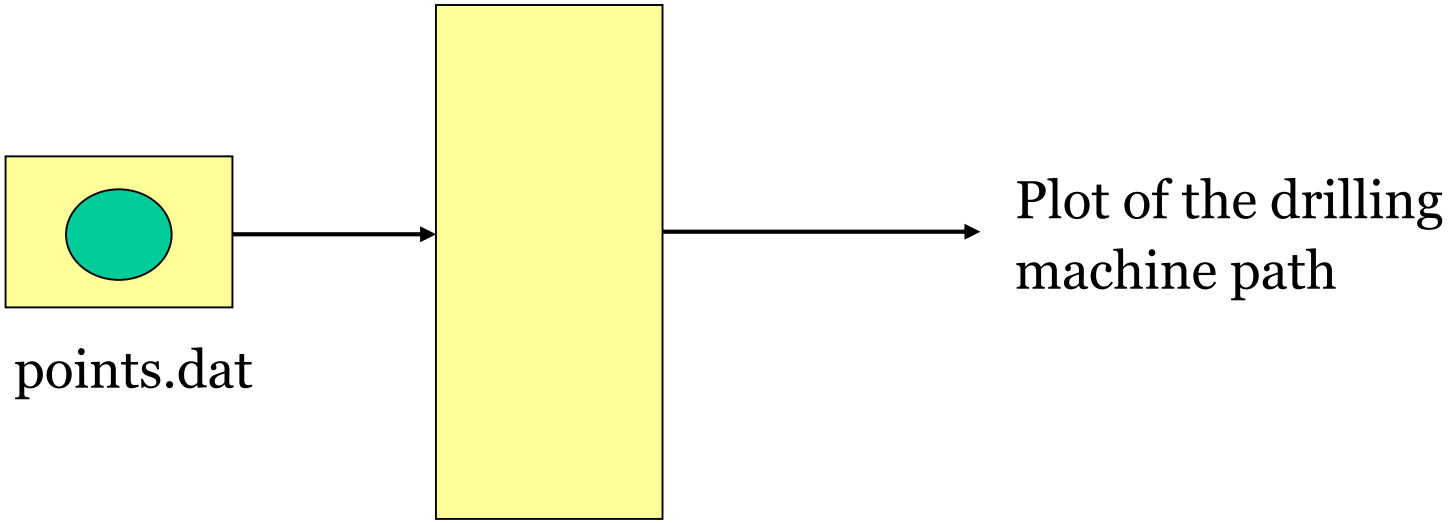
Design a smooth curve, using cubic-spline interpolation, that can be used to guide a drilling machine to several location and then back to the original position.



Example

- **Inputs/Outputs Description**

The following I/O diagram shows that the input is a file containing the xy coordinates of the points over which the drilling machine must pass and its original position.



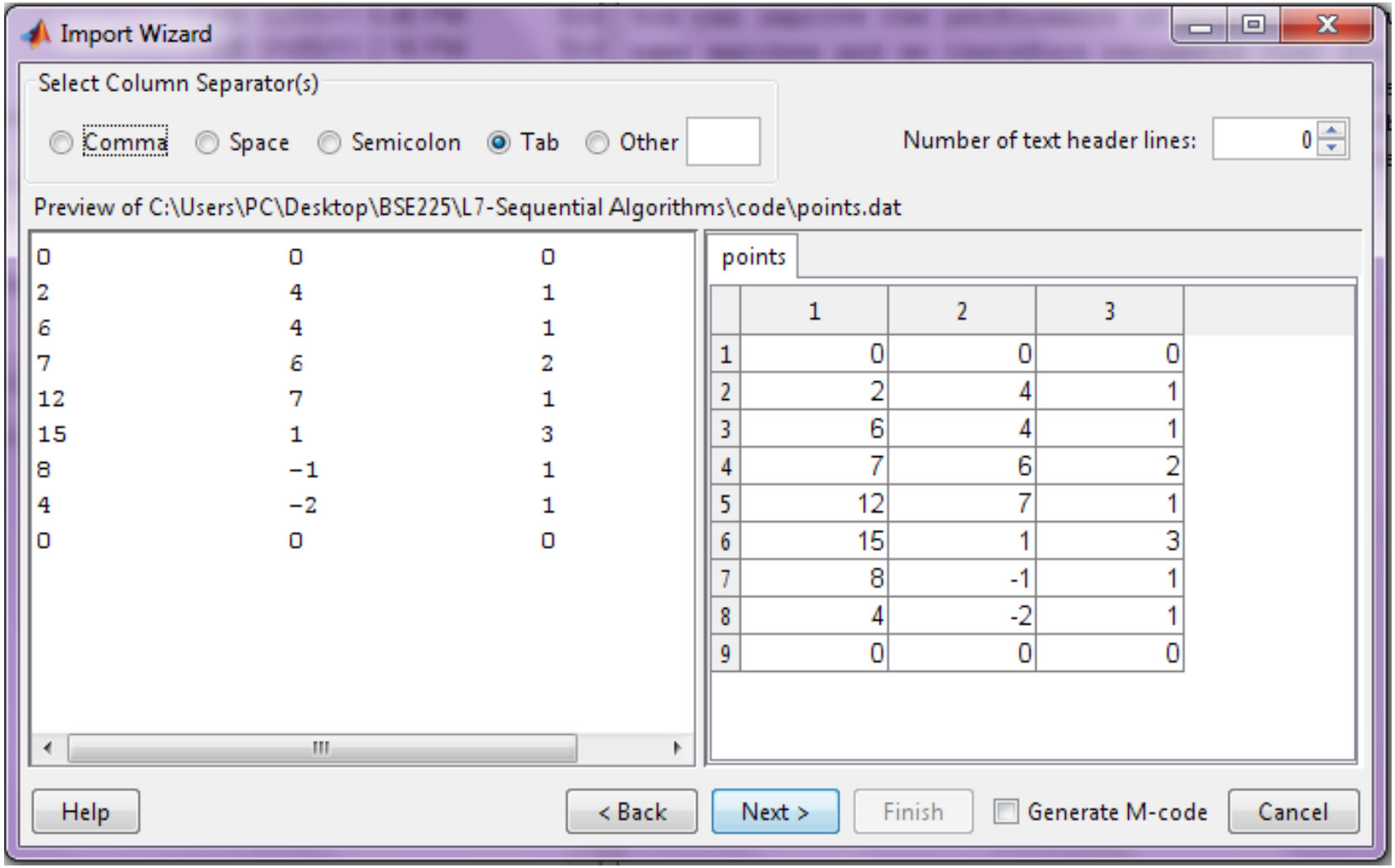
Example

- **Inputs/Outputs Description**

x	y	code	Interpretation
0	0	0	home position
2	4	1	intermediate position
6	4	1	intermediate position
7	6	2	drill
12	7	1	intermediate position
15	1	3	release
8	-1	1	intermediate position
4	-2	1	intermediate position
0	0	0	home position

Example

- **Input file points.dat**



Example

- **Matlab Program**

```
% Drilling Machine path

% read data file.

load points.dat;

x=points(:,1);

y=points(:,2);

code=points(:,3);
```

Example

- **Matlab Program**

```
%generates the three separate paths.

drill=find(code==2);

release=find(code==3);

lenx=length(x);

x1=x(1:drill);        y1=y(1:drill);

x2=x(drill:release);

y2=y(drill:release);

x3=x(release:lenx);       y3=y(release:lenx);
```

Example

- **Matlab Program**

```
% Compute time increment and corresponding
time sequences.
incr=min(abs(x(2:lenx)-x(1:lenx-1)))/10;
t1=x(1):incr*sign(x(drill)-x(1)):x(drill);
t2=x(drill):incr*sign(x(release)-
x(drill)):x(release);
t3=x(release):incr*sign(x(lenx)-
x(release)):x(lenx);
```

Example

- **Matlab Program**

```
% Compute splines

s1=interp1(x1,y1,t1,'spline');

s2=interp1(x2,y2,t2,'spline');

s3=interp1(x3,y3,t3,'spline');
```

Example

- **Matlab Program**

```
% Plot spline path.

plot([t1 t2 t3],[s1 s2 s3],[x1' x2' x3'],…

[y1' y2' y3'], '0'),…

title('Path for Drilling Machine'),…

xlabel('x'),ylabel('y'), grid,…

axis([-1,16,-4,10])
```

Example

- **Program Run**