Bil 108

Introduction to the Scientific and Engineering Computing with MATLAB

Lecture 8

F. Aylin Konuklar   Lale T. Ergene

Polynomial Function Summary

Function	Description
conv(p1,p2)	Multiply polynomials
deconv(pnum,pdenom)	Divide polynomials.
poly(r)	Polynomial with specified roots.
polyder(p)	Polynomial derivative
polyfit(x,y,ndegree)	Polynomial curve fitting.
polyval	Polynomial evaluation
polyvalm	Matrix polynomial evaluation
roots(r)	Find polynomial roots

Review of Basic Statistics

- **mean(x)**

Computes the mean(average value) of the elements of the vector x.

$$\mu = \frac{\sum_{k=1}^{N} x_k}{N}$$

$$where \sum_{k=1}^{N} x_k = x_1 + x_2 + \ldots\ldots + x_N$$

- **std(x)** Computes the standard deviation of the values in x

The standard deviation σ is defined as the square root of the variance

$$\sigma^2 = \frac{\sum_{k=1}^{N} (x_k - \mu)^2}{N-1}$$

Curve Fitting (polyfit)

- Regression analysis is a process of fitting a function to a set of data points.

- Curve fitting with polynomials is done with polyfit function which uses the least squares method.

p=polyfit(x,y,n)

p is the vector of the coefficients of the polynomial that fits the data

x is a vector with the horizontal coordinate
y is a vector with the vertical coordinate
n is the degree of the polynomial

'polyfit' for different functions

Function		polyfit function form
$y=bx^m$	power function	**p=polyfit(log(x),log(y),1)**
$y=be^{mx}$ or $y=b10^{mx}$	exponential function	**p=polyfit(x,log(y),1)** or **p=polyfit(x,log10(y),1)**
$y=m\ln(x)+b$ or $y=m\log(x)+b$	logarithmic function	**p=polyfit(log(x),y,1) or p=polyfit(log10(x),y,1)**
$y=1/mx+b$	reciprocal function	**p=polyfit(x,1./y,1)**

Other consideration when choosing a function are

- Exponential functions cannot pass through the origin
- Exponential functions can only fit data with all positive y’s or all negative y’s
- Logarithmic functions cannot model x=0, or negative values of x
- For the power function y=0 when x=0
- The reciprocal equation cannot model y=0

Least-Squares Curve Fitting

- Experimental data always has a finite amount of error included in it, due to both accumulated instrument inaccuracies and also imperfections in the physical system being measured. Even data describing a linear system won’t all fall on a single straight line.
- Least-squares curve fitting is a method to find parameters that fit the error-laden data as best we can.

least squares objective

- Find a function,e.g. a polynomial of whatever order, that minimizes the mean square error

Least Squares

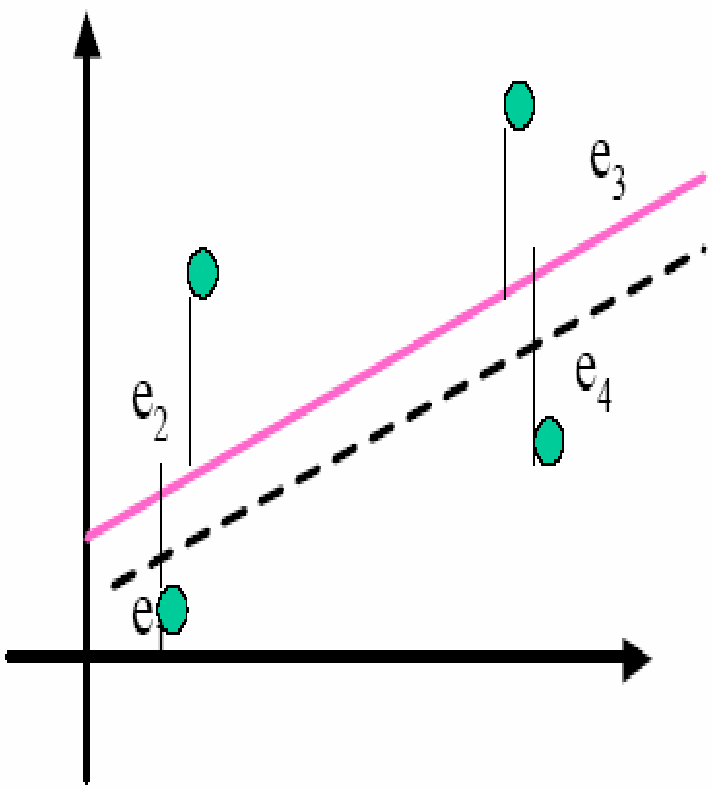
- Finds the "best fit" straight line
- Minimizes the amount each point is away from the line
- It's possible none of the points will fall on the line

Advantages

- Positive errors do not cancel negative ones.
- Differentiation of the sum of squares is easy

Least Square Regression

- The method seeks to *minimize* the *sum* of the *squares* of the *differences* between the function



$$\text{Minimize} \quad \left(\sum_{i=1}^{N} |e_i|\right)$$

Sum of absolute value of errors

$$f(x) = a_1 x + a_0$$

Linear Regression

Linear regression is the method of finding the slope and y intercept of a line that best fits a set of data, and is the most common least-squares method used in mechanical engineering.

- Consider a series of data points (x_1, y_1), (x_2, y_2), $\cdots$, (x_n, y_n).
- A linear function that attempts to fit this data would be $y = a_0 + a_1 x$, where a_0 and a_1 are constants not yet determined.
- After plugging each known value x_i into the equation, we'll find some amount of error between the y_i points in the original data the the predicted y value from the linear model. This error e_i is found with the relationship $e_i = y_i - a_0 - a_1 x_i$.

The error e_i at each point may be positive or negative, large or small. One way to quantify how good a fit a particular line is would be to:

- Treat negative and positive errors the same.
- Penalize lines that have large errors at one or more points.
- Don't encourage lines that fit a few points exactly if they fit several other points poorly.

One easy way to achieve these goals is to square the individual errors and add them all up:

$$S = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

We can treat the sum of the squares of the error S as a function of a_0 and a_1. An optimal pair of a_0 and a_1 values would minimize S.

- How do we find these optimal a_0 and a_1 values?
- Minimize S by finding values of a_0 and a_1 where the derivatives of S with respect to a_0 and a_1 are zero simultaneously.

$$\frac{\delta S}{\delta a_0} = -2\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i) = 0$$

$$\frac{\delta S}{\delta a_1} = -2\sum_{i=1}^{n}x_i(y_i - a_0 - a_1 x_i) = 0$$

Reformat the previous two equations as

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} a_0 - a_1 \sum_{i=1}^{n} x_i = 0$$

$$\sum_{i=1}^{n} x_i y_i - a_0 \sum_{i=1}^{n} x_i - a_1 \sum_{i=1}^{n} x_i^2 = 0$$

Since $\sum_{i=1}^{n} a_0 = na_0$, these two equations can be reformatted as two simultaneous linear algebraic equations with a_0 and a_1 as the unknowns:

$$na_0 + \left(\sum_{i=1}^{n} x_i\right) a_1 = \sum_{i=1}^{n} y_i$$

$$\left(\sum_{i=1}^{n} x_i\right) a_0 + \left(\sum_{i=1}^{n} x_i^2\right) a_1 = \sum_{i=1}^{n} x_i y_i$$

In matrix form:

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{Bmatrix}$$

Solve the matrix form of the equations with Cramer's rule:

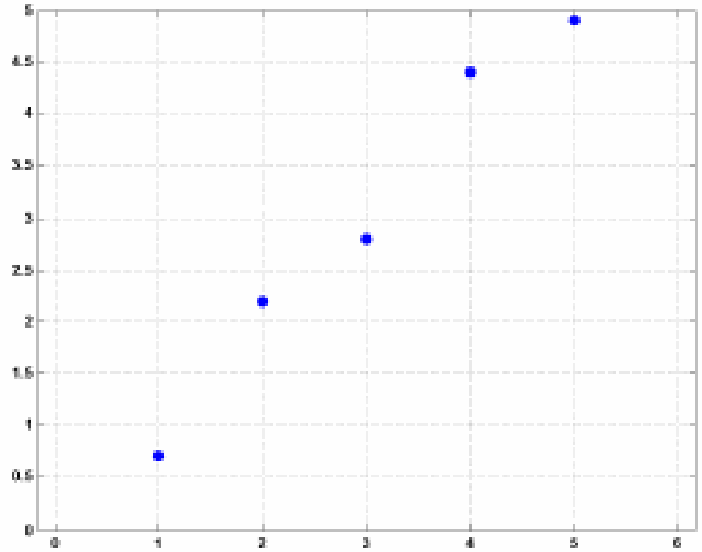
$$a_0 = \frac{\left| \begin{bmatrix} \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \right|}{\left| \begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \right|} = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$a_1 = \frac{\left| \begin{bmatrix} n & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i y_i \end{bmatrix} \right|}{\left| \begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \right|} = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

EXAMPLE:

Use linear regression to fit the following data points

i	1	2	3	4	5
x_i	1	2	3	4	5
y_i	0.7	2.2	2.8	4.4	4.9



For this problem:

$$n = 5$$

$$\sum_{i=1}^{n} x_i = 1 + 2 + 3 + 4 + 5 = 15$$

$$\sum_{i=1}^{n} y_i = 0.7 + 2.2 + 2.8 + 4.4 + 4.9 = 15.0$$

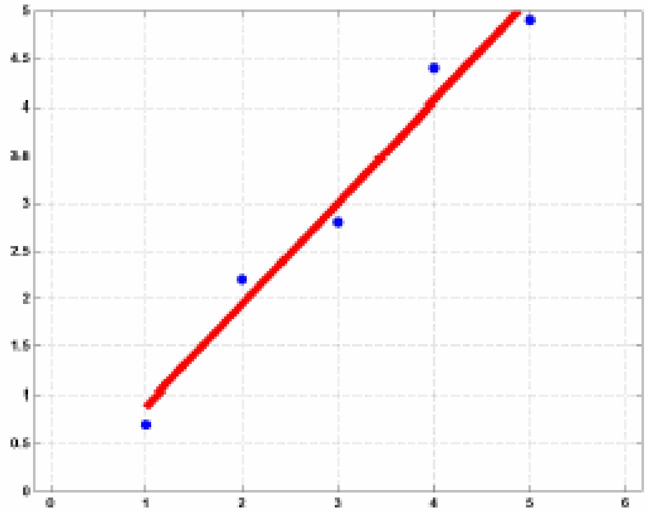
$$\sum_{i=1}^{n} x_i^2 = 1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$$

$$\sum_{i=1}^{n} x_i y_i = 1(0.7) + 2(2.2) + 3(2.8) + 4(4.4) + 5(4.9) = 55.6$$

$$a_0 = \frac{15(55) - 15(55.6)}{5(55) - 15^2} = -0.18$$

$$a_1 = \frac{5(55.6) - 15(15.0)}{5(55) - 15^2} = 1.06$$

So $y = -0.18 + 1.06x$:

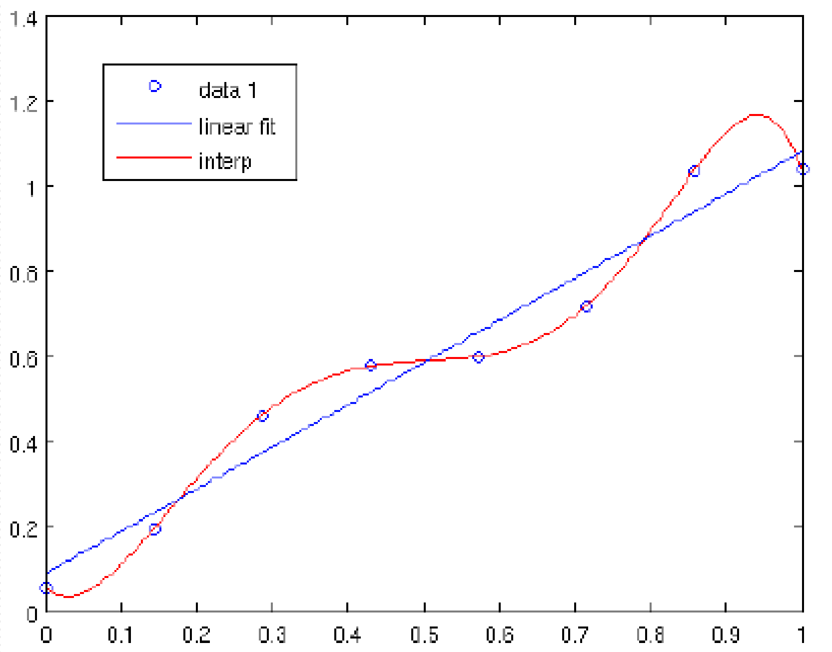


Interpolation

Given a set of data

$$y_i \text{ at } x_i \quad i = 1 \ldots n$$

find a function $\mathcal{F}(x)$ that represents this data



exactly pass through y_i with $\mathcal{F}(x_i)$ for each i

Matrix Evaluation of Interpolation

$$\mathcal{F}(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}$$

So for each x_i we have

$$\mathcal{F}(x_i) = c_0 + c_1 x_i + c_2 x_i^2 + \cdots + c_{n-1} x_i^{n-1} = y_i$$

OR

$$\begin{aligned}
c_0 + c_1 x_1 + c_2 x_1^2 + \cdots + c_{n-1} x_1^{n-1} &= y_1 \\
c_0 + c_1 x_2 + c_2 x_2^2 + \cdots + c_{n-1} x_2^{n-1} &= y_2 \\
c_0 + c_1 x_3 + c_2 x_3^2 + \cdots + c_{n-1} x_3^{n-1} &= y_3 \\
&\vdots \\
c_0 + c_1 x_n + c_2 x_n^2 + \cdots + c_{n-1} x_n^{n-1} &= y_n
\end{aligned}$$

Vandermonde Matrix

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ & & & \vdots & \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

[X].[C] = [Y]

C=inv(X)*Y or C=X\Y

- The coefficients of the polynomial are obtained by solving this system of equations.

Interpolation

- Interpolation involves computing approximate values between limiting or endpoint values
- One dimensional interpolation :
- Independent variable x
- Dependent variable y
- The function **interp1** performs one-dimensional interpolation, an important operation for data analysis and curve fitting.

xi is a vector containing the points at which to interpolate.

yi = interp1(x,y,xi,method)

yi is the interpolated value

y is a vector containing the values of a function,

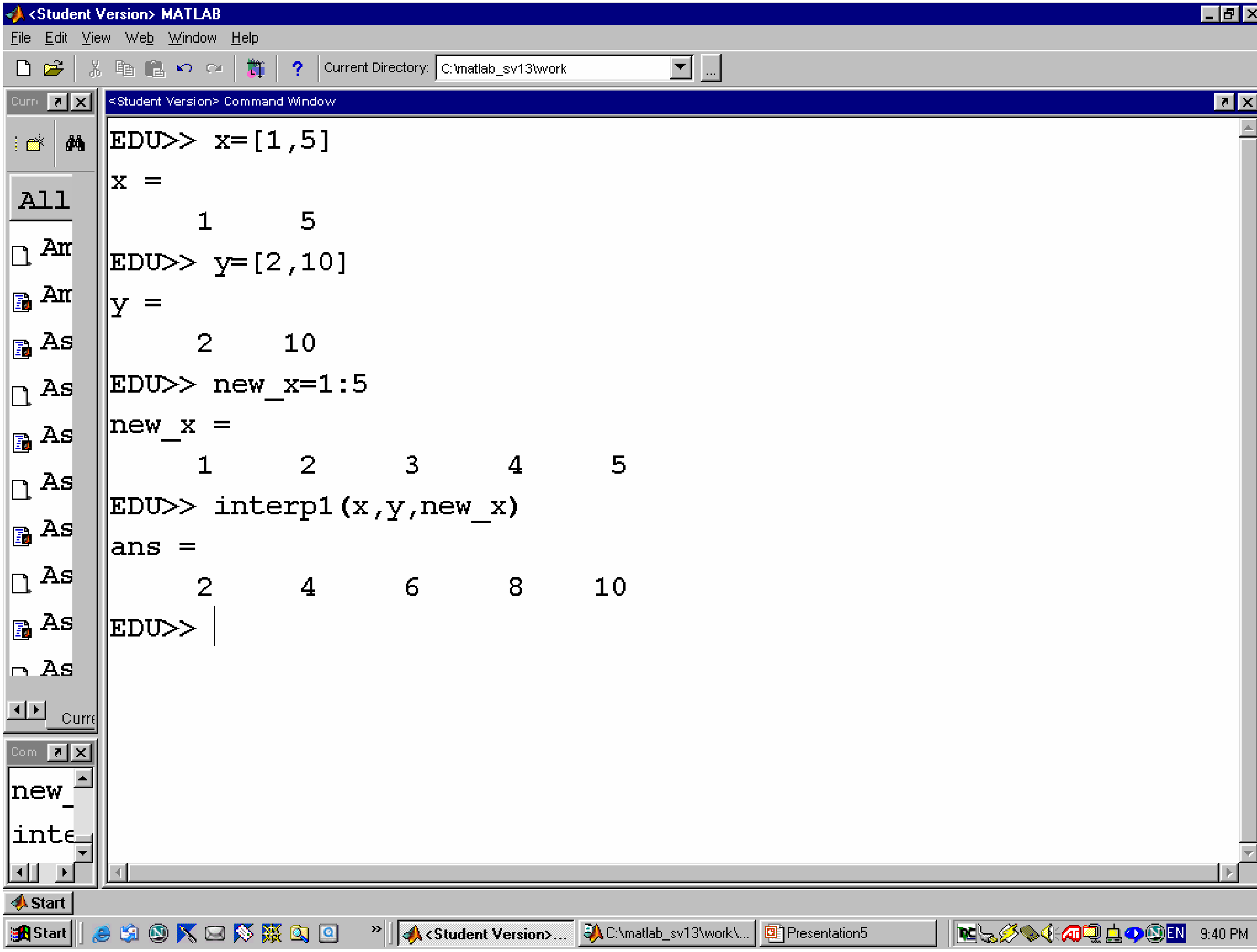
method is an optional string specifying an interpolation method

x is a vector of the same length containing the points for which the values in y are given.

MATLAB Code

- interp1 is the MATLAB function for linear interpolation
- First define an array of x and y
- Now define a new x array, that includes the x values for which you want to find y values
- new_y=interp1(x,y,x_new)

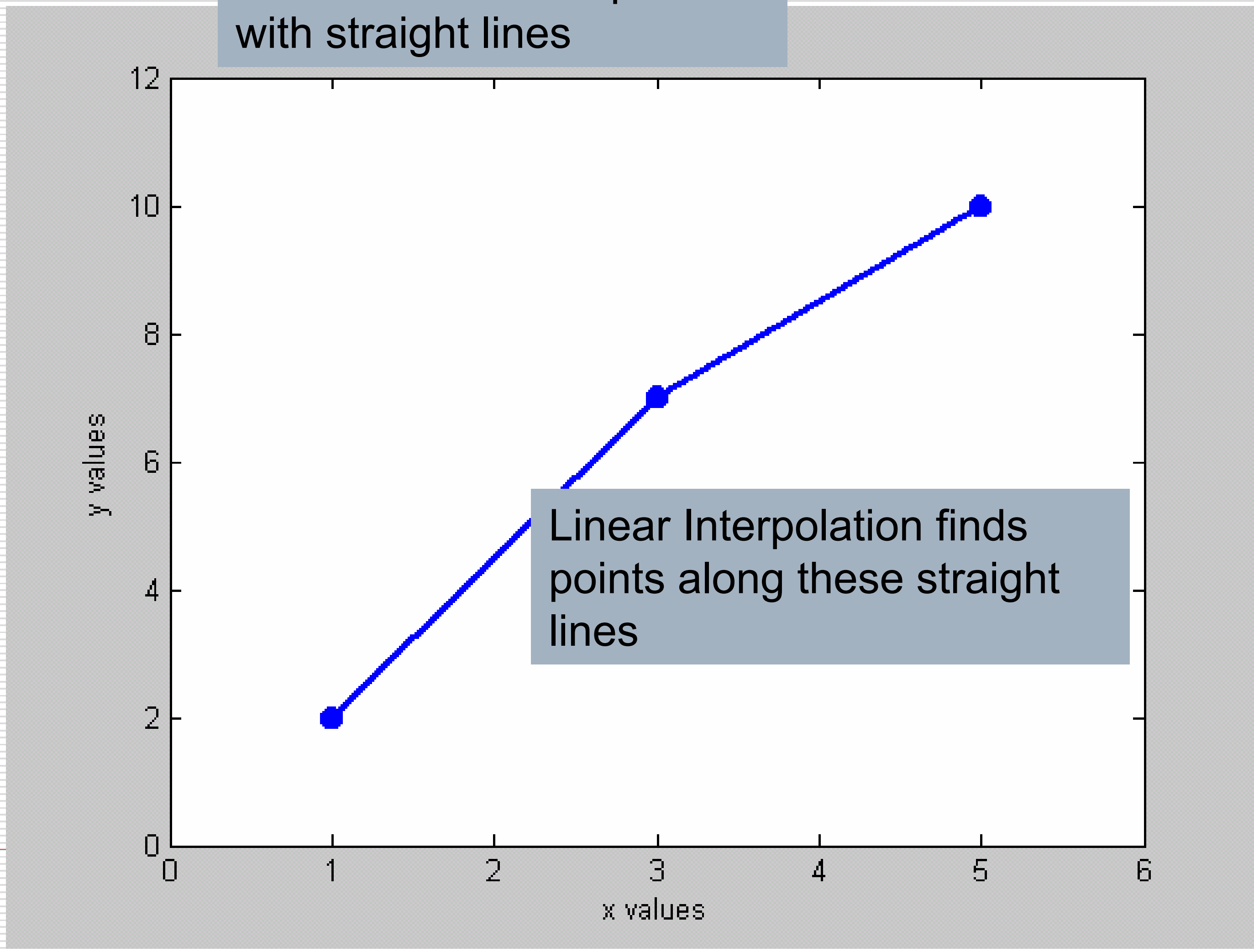
In Matlab



<Student Version> MATLAB
File Edit View Web Window Help
Current Directory: C:\matlab_sv13\work
<Student Version> Command Window
EDU>> x=[1,5]
x =
1 5
EDU>> y=[2,10]
y =
2 10
EDU>> new_x=1:5
new_x =
1 2 3 4 5
EDU>> interp1(x,y,new_x)
ans =
2 4 6 8 10
EDU>>
Start
<Student Version> ...
C:\matlab_sv13\work\...
Presentation5
9:40 PM

Graphically

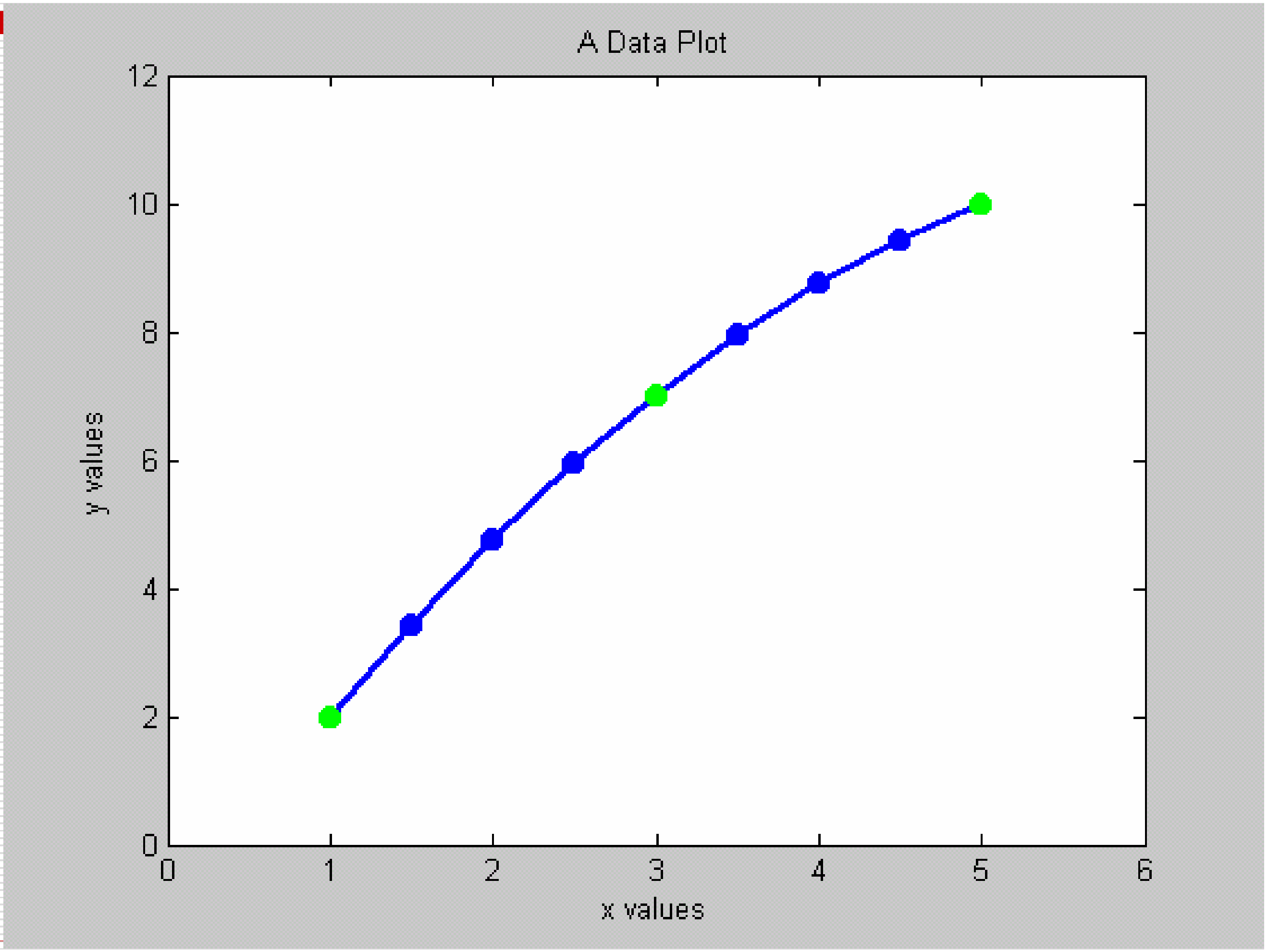
MATLAB connects points with straight lines



Linear Interpolation finds points along these straight lines

Cubic Spline

- A cubic spline creates a smooth curve, using a third degree polynomial



Methods

- ***Nearest neighbor interpolation* (method = 'nearest').** This method sets the value of an interpolated point to the value of the nearest existing data point.

- ***Linear interpolation* (method = 'linear').** This method fits a different linear function between each pair of existing data points, and returns the value of the relevant function at the points specified by xi. This is the default method for the interp1 function.

- ***Cubic spline interpolation* (method = 'spline').** This method fits a different cubic function between each pair of existing data points, and uses the spline function to perform cubic spline interpolation at the data points.

- ***Cubic interpolation* (method = 'pchip' or 'cubic').** These methods are identical. They use the pchip function to perform piecewise cubic Hermite interpolation within the vectors x and y.

Hints for the methods

- When the ***nearest*** and ***linear*** methos are used the values of xi must be within the domain of x If the ***spline*** or the ***pchip*** methods are used, xi can have values outside the domain of x and the function **interp1** performs extrapolation.

- The spline method can also give large errors if the input data points are nonuniform such that some points are much closer than others.

Example: Interpolation

- The following data points which are points of the function
- $f(x) = 1.5^x \cos(2x)$ are given. Use linear, spline and pchip interpolation methods to calculate the value of y between the points. Make a figure for each of the interpolation methods. In the figure show the points, a plot of the function, and a curve that corresponds to the interpolation method.

x	0	1	2	3	4	5
y	1,0	-0,6242	-1,4707	3,2406	-0,7366	6,3717

1)The population of China from the year 1940 to the year 2000 is given in the following table:

Year	1940	1950	1960	1970	1980	1990	2000
Population(milions)	537	557	682	826	981	1135	1262

a)Determine the exponential function that best fits the data. Use the function to estimate the population in 1955.

b)Curve fit the data with a quadratic equation (second order polynomial). Use the function to estimate the population in 1955.

c)Fit the data with linear and spline interpolations. Estimate the population in 1955 with linear and spline interpolations.

In each part make a plot of the data points (circle markers) and the curve fitting or the interpolation curves. Note that part c has two interpolation curves. The actual population in China in 1955 was 614.4 million.

china.m
chinab.m
chinac.m

- Find the velocity as a function of time. To do this, you will need to download velocity.dat file.
- (a) Fit a spline through the data.
- (b) Fit the data with the parabolic fit

$f(x) = Ax^3 + Bx^2 + Cx+D$ and calculate the error at each point. Use POLYFIT and POLYVAL

- (c) Compare your results by providing the graphs of all that you do. Comment on which method you think is best.