INTRODUCTION TO SCIENTIFIC & ENG. COMPUTING
BIL 108E, CRN 44448

Week.8_ Solving of Simple Equations
Week.9_ Polynomials Examples

Dr. Feyzi HAZNEDAROĞLU

İstanbul Teknik Üniversitesi - İnşaat Fakültesi
Room : 250A, e-mail :haznedar@itu.edu.tr

08 - 04 - 2011

TENTATIVE SCHEDULE

Wk	Date	Topics
1	Feb. 11	Introduction to Scientific and Engineering Computing
2	Feb. 18	Introduction to Program Computing Environment
3	Feb. 25	Variables, Operations and Simple Plot
4	Mar. 04	Algorithms and Logic Operators
5	Mar. 11	Flow Control, Errors and Source of Errors
6	Mar. 18	Functions & Linear Algebra
7	Mar. 25	Arrays
8	Apr. 01	Solving of Simple Equations
9	Apr. 08	Polynomials Examples Exam 1
10	Apr. 15	Applications of Curve Fitting
11	Apr. 22	Applications of Interpolation
12	Apr. 29	Applications of Numerical Integration
13	May.06	Symbolic Mathematics
14	May.13	Ord Dif.Equations (ODE) solution with Built-in Functions

Lecture # 9 - Polynomials Examples

1 INTERPOLATION
- 1 Lagrange Interpolation
- 2 Chebyshev Interpolation
- 3 Linear Interpolation
- 4 Spline Functions

2 APPROXIMATION
- 1 Least Squares Approximation
- 2 Linear Regression

INTERPOLATION

- Data points for a function (x_i , y_i) $i = 0, 1, 2, \ldots, n$
- x_i are all distinct and are called nodes.
- Approximate function should satisfy $f(x_i) = y_i$, $i = 0, 1, \ldots, n$ f is called interpolant of the set of data y_i
- polynomial interpolant
  $f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$
- trigonometric interpolant
  $f(x) = a_{-M} e^{-iMx} + \ldots + a_0 + \ldots + a_M e^{iMx}$
- rational interpolant

INTERPOLATION

VANDERMONDE MATRIX

$\tilde{f}(x) = \sum_{k=0}^{n} c_k \varphi_k(x_j) = y_j,\ j = 0, 1, 2, \ldots, n$

If we choose $\varphi_k(x) = x^k$

$p(x) = \sum_{k=0}^{n} c_k x^k$

$$\begin{pmatrix} 1 & x_0 & \ldots & x_0^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

Xc = y

n+1 equations for n+1unknowns $c_0, c_1, \ldots, c_n$.

The matrix with the given structure is named as Vandermonde matrix

INTERPOLATION

INERPOLATION WITH POLYNOMIALS

LAGRANGE INTERPOLATION

$$\varphi_k(x) = \prod_{j=0}^{n} \frac{x - x_j}{x_k - x_j}$$

$$\Pi_n(x) = \sum_{k=0}^{n} y_k \varphi_k(x)$$

write the equation for all n+1 points.

$L_j(x) = \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i}$

$= \frac{(x-x_0)\ldots(x-x_{j-1})(x-x_{j+1})\ldots(x-x_n)}{(x_j-x_0)\ldots(x_j-x_{j-1})(x_j-x_{j+1})\ldots(x_j-x_n)}$

for $j = 0, 1, 2, \ldots, n$ every $L_j(x)$ has the property

$L_j(x_j) = 1$ and $L_j(x_i) = 0$ if $i \neq j$.

INTERPOLATION

EXAMPLE

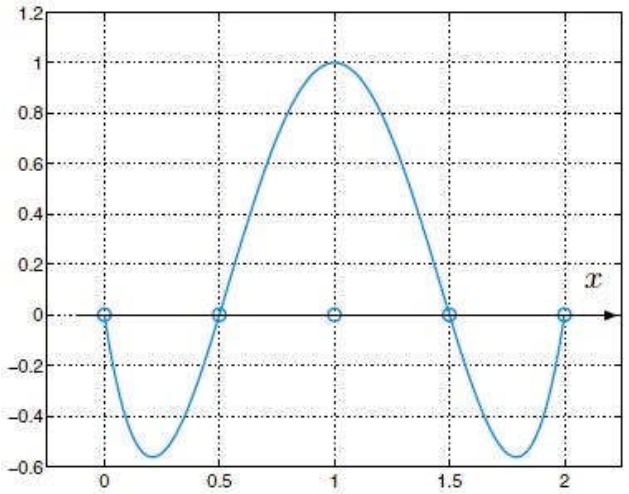
- Draw the graph of the Lagrange polynom $L_2(x)$ for the $x_j = j$, j = 0, 1, 2, 3, 4 supporting points. The supporting points are equispaced.

Lagrange –Polynom $L_2(x) = \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)(x_2-x_4)}$

Here $L_2(x_2) = 1$ and $L_2(x_i) = 0$ if $i \neq 0$

$$p(x) = \sum_{j=0}^{n} y_j L_j(x)$$

INTERPOLATION



INTERPOLATION

EXAMPLE:

```
function v = polyinterp(x,y,u)
n = length(x);
v = zeros(size(u));
for k = 1:n
   w = ones(size(u));
   for j = [1:k-1 k+1:n]
      w = (u-x(j))./(x(k)-x(j)).*w;
   end
   v = v + w*y(k);
end
```

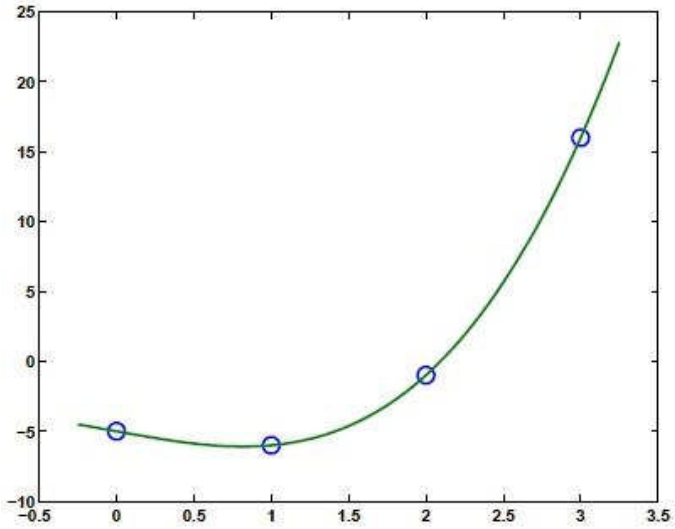
To illustrate `polyinterp`, create a vector of densely spaced evaluation points.

```
u = -.25:.01:3.25;
```

Then

```
v = polyinterp(x,y,u);
plot(x,y,'o',u,v,'-')
```


INTERPOLATION

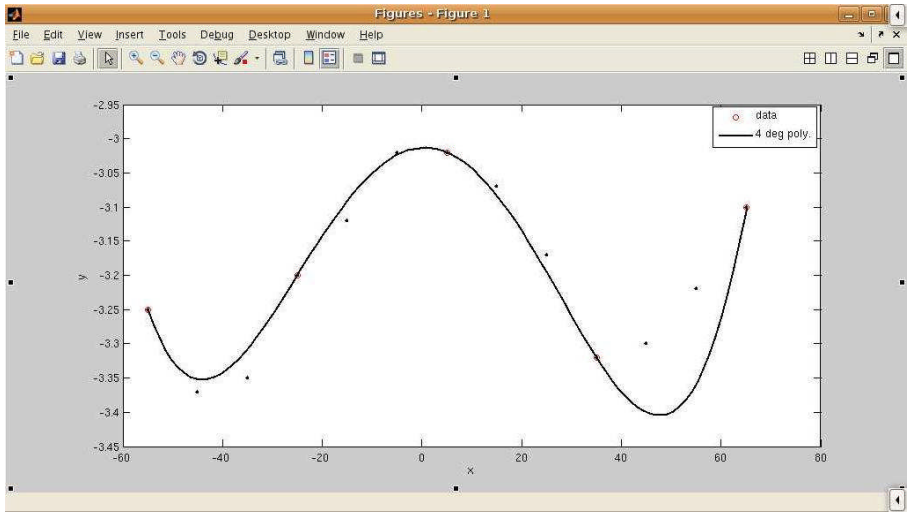


INTERPOLATION

EXAMPLE:

- x = [-55 -25 5 35 65];
- y = [-3.25, -3.2, -3.02, -3.32, -3.1];
  format short e;
- c = polyfit(x, y, 4);
- p4x = linspace(x(1), x(end), 100); p4y = polyval(c,p4x);
- plot(x,y,'or')
  hold('on')
- plot(p4x, p4y, 'k-')
  xlabel('x')
- ylabel('y')
- legend('data', '4 deg poly.')
  xdat = [-55:10:65];
- ydat = [-3.25, -3.37, -3.35, -3.2, -3.12, -3.02, -3.02 ...
- -3.07, -3.17, -3.32, -3.3, -3.22, -3.1];
- plot(xdat,ydat,'.k')
-

INTERPOLATION



INTERPOLATION

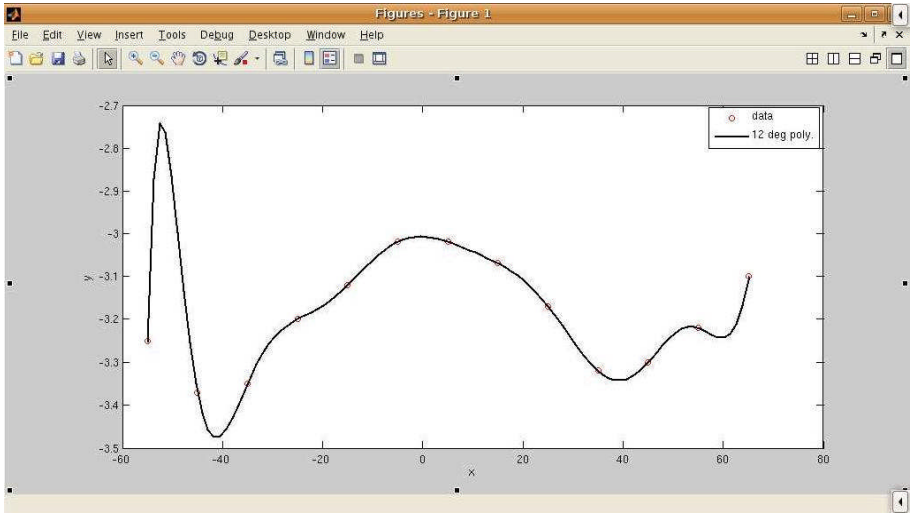
LAGRANGE INTERPOLATION ERROR

EXAMPLE

clear;
clf;
clc;

- x = [-55:10:65];
- y = [-3.25, -3.37, -3.35, -3.2, -3.12, -3.02, -3.02 ...
- -3.07, -3.17, -3.32, -3.3, -3.22, -3.1];
- format short e;
- c = polyfit(x, y, 12);
- p12x = linspace(x(1), x(end), 100); p12y = polyval(c,p12x);
- plot(x,y,'or')
  hold('on')
- plot(p12x, p12y, 'k-')
  xlabel('x')
- ylabel('y')
- legend('data', '12 deg poly.')

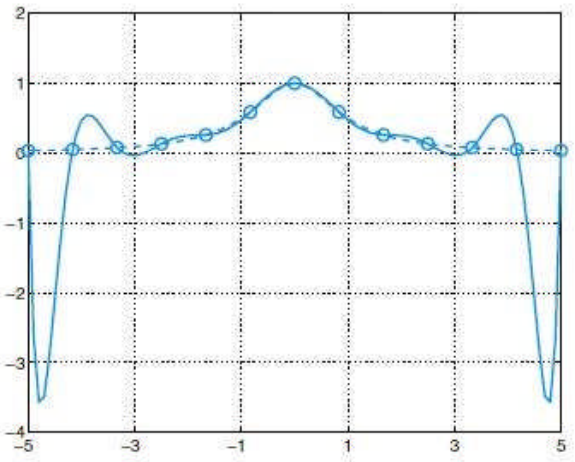
INTERPOLATION



INTERPOLATION

CHEBYSHEV INTERPOLATION

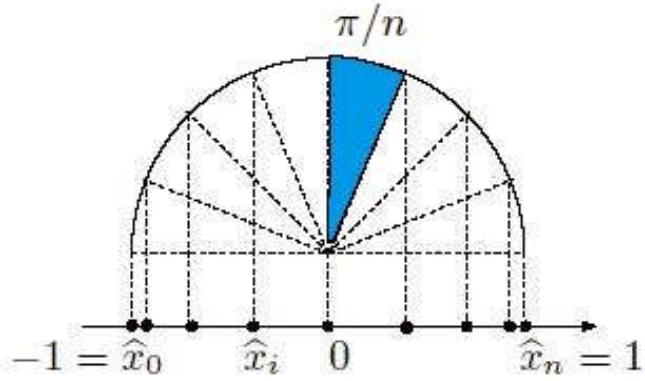
Runge function f (x) = 1 / (1+x 2)



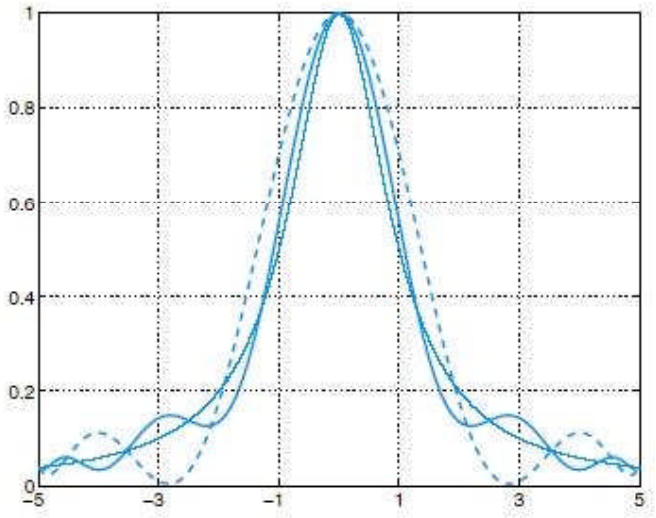
INTERPOLATION

CHEBYSHEV INTERPOLATION

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\widehat{x}_i, \text{ where } \widehat{x}_i = -\cos(\pi i/n),\ i = 0, \ldots, n$$



CHEBYSHEV INTERPOLATION



PIECEWISE LINEAR INTERPOLATION

- Use linear composite interpolation
  - When the function f is nonsmooth or
  - When f is known by its values at a set of given points

Given: nodes(not necessarily uniform) $x_0 < x_1 < \ldots < x_n$,
interval I_i , $|x_i , x_{i+1}|$ Approximate the function f by a continuous function which,on each interval, is given by the segment joining the two points $(x_i , f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$

PIECEWISE LINEAR INTERPOLATION

Piecewise linear interpolation polynomial of f is $\Pi_1^H f$ for $x \in I_i$,

$$\Pi_1^H f(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i)$$

The upper – index H denotes the maximum length of the interval I_i.
For all x in the interpolation interval, $\Pi_1^H f(x)$ tends to $f(x)$ when $H \longrightarrow 0$ provided that f is sufficiently smooth.

PIECEWISE LINEAR INTERPOLATION

- Use linear composite interpolation
  - When the function f is nonsmooth or
  - When f is known by its values at a set of given points

Given: nodes(not necessarily uniform) $x_0 < x_1 < \ldots < x_n$,
interval I_i , $|x_i , x_{i+1}|$ Approximate the function f by a continuous function which,on each interval, is given by the segment joining the two points $(x_i , f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$

INTERPOLATION

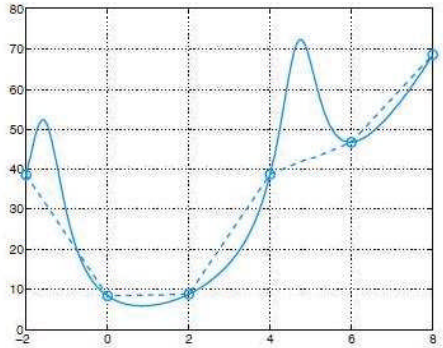
PIECEWISE LINEAR INTERPOLATION

`s1=interp1(x,y,z)` is used to calculate the linear interpolation value in a given interval.

- x, y: data points
- z: arbitrary points with an arbitrary dimension

INTERPOLATION

EXAMPLE



The function $f(x) = x^2 + 10/(\sin(x) + 1.2)$ (*solid line*) and its piecewise linear interpolation polynomial $\Pi_1^H f$ (*dashed line*)

INTERPOLATION

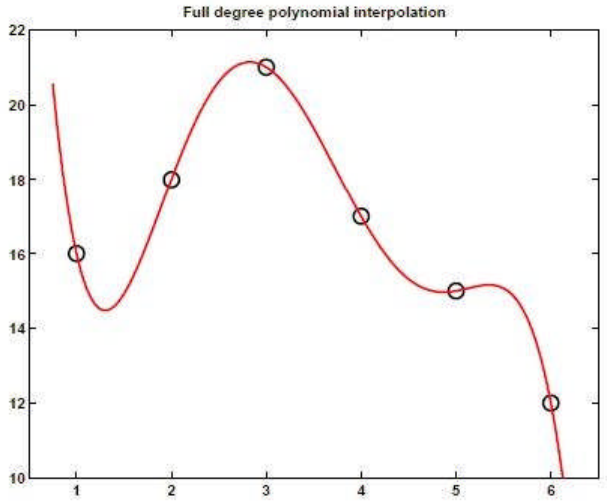
EXAMPLE

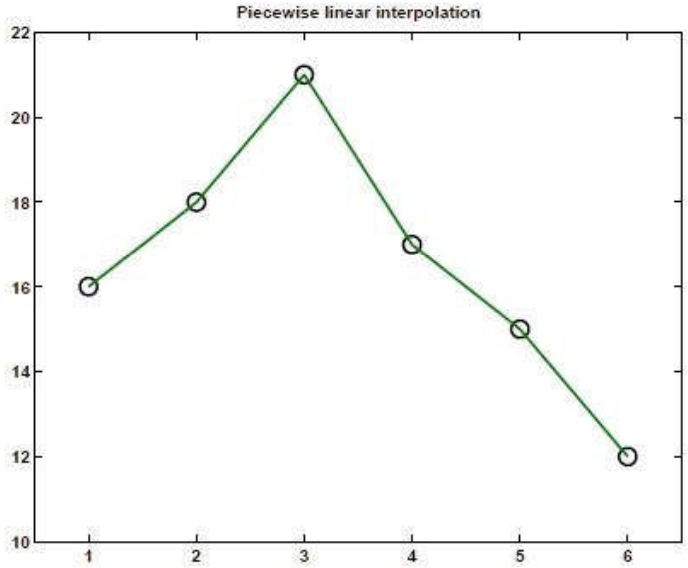
```
x = 1:6;
y = [16 18 21 17 15 12];
plot(x,y,'o',x,y,'-');
```


INTERPOLATION

EXAMPLE

```
x = 1:6;
y = [16 18 21 17 15 12];
plot(x,y,'o',x,y,'-');
```





SPLINE FUNCTIONS

- Piecewise polynomial interpolation of degree n ≥ 2 can be defined. In several applications, it is desirable to get
- approximation by smooth functions which have at least a continuous derivative. Function properties:

1 on each interval $I_i = [x_i , x_{i+1}]$, for i = 0, . . . , n − 1, s_3 is a polynomial of degree 3 which interpolates the pairs of values $(x_j , f(x_j))$ for j = i , i + 1;

2 s_3 has continuous first and second derivatives in the nodes x_i , i = 1, . . . , n − 1

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

SPLINE FUNCTIONS
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 x_i . This is the default method for the interp1 function.

SPLINE FUNCTIONS
METHODS

- 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 .

INTERPOLATION

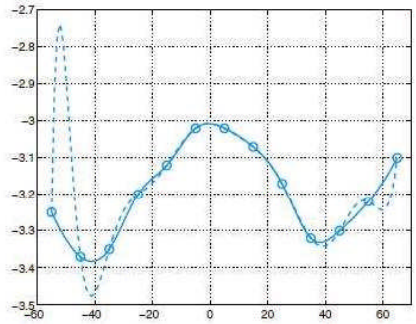
SPLINE FUNCTIONS

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

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

INTERPOLATION

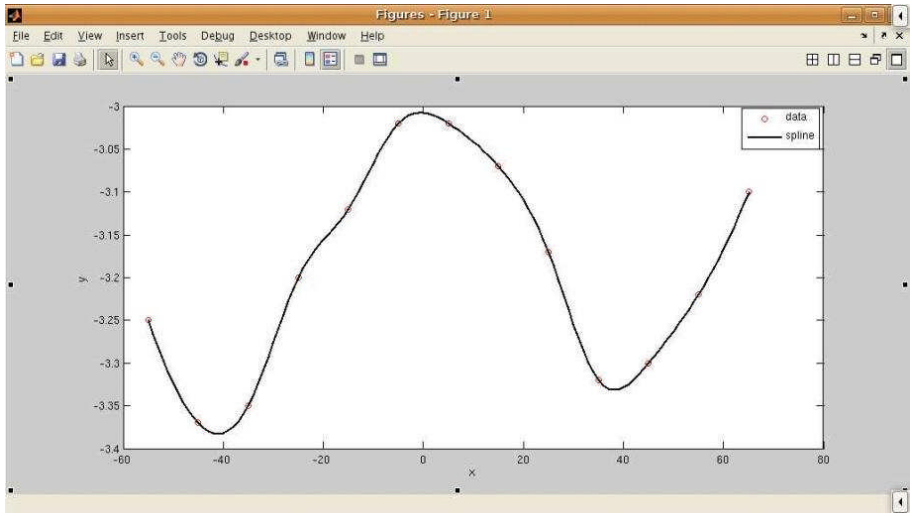
EXAMPLE:



Comparison between the interpolating cubic spline and the Lagrange interpolant for the case considered in Example

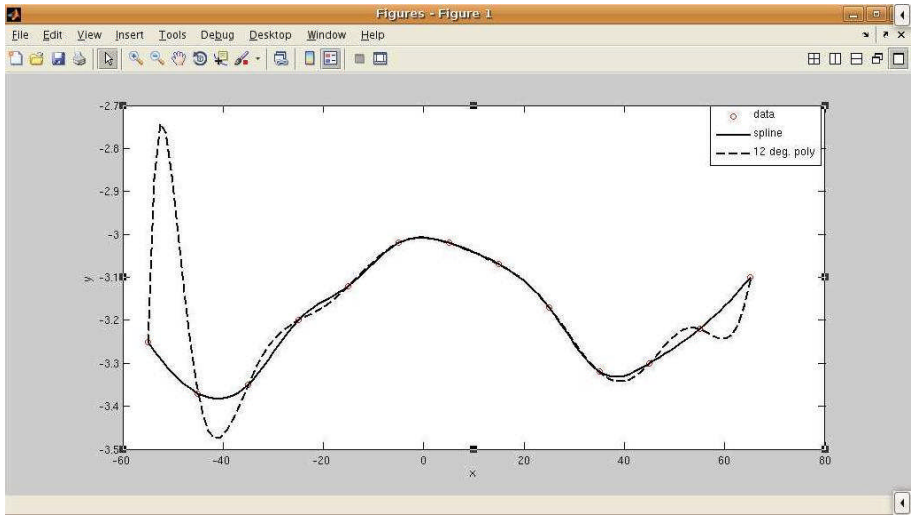
INTERPOLATION

EXAMPLE:



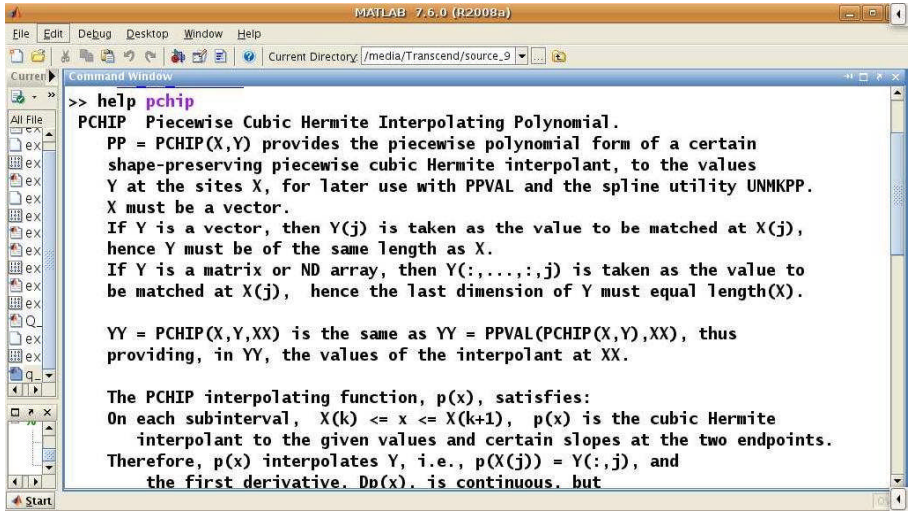
INTERPOLATION

EXAMPLE:



APPROXIMATION

EXAMPLE:



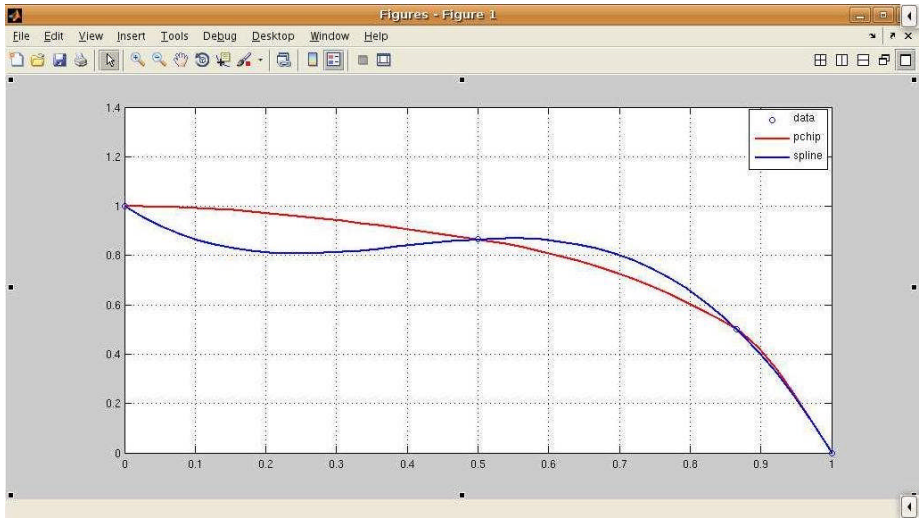
APPROXIMATION

EXAMPLE:

```
clear;clf;clc;
t = linspace (0 , pi /2 ,4)
x = cos ( t ); y = sin ( t );
xx = linspace (0 ,1 ,40);
plot (x ,y , 'o');
hold('on');
plot(xx, pchip (x, y, xx))
plot(xx, spline (x, y, xx))
grid('on')
legend('data', 'pchip', 'spline')
```


APPROXIMATION

EXAMPLE:



APPROXIMATION

EXAMPLE:

- The following data points which are points of the function $f(x) = 1.5^x \cos(2x)$ are given.

x	0.0	1.0	2.0	3.0	4.0	5.0
y	1.0	-0.6242	-1.4707	3.2406	-0.7366	6.3717

- Use linear, spline and pchip interpolation methods to calculate the value of y between the points. Create 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.

APPROXIMATION

LEAST SQUARES METHOD

For a linear equation:

- If the number of linear equations is less than the unknowns, the equation system is under -determined (or infinite solutions)
- If the number of linear equations is more than the unknowns, the equation system is over -determined.

REGRESSION ANALYSIS

REGRESSION ANALYSIS

- 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.
- 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.

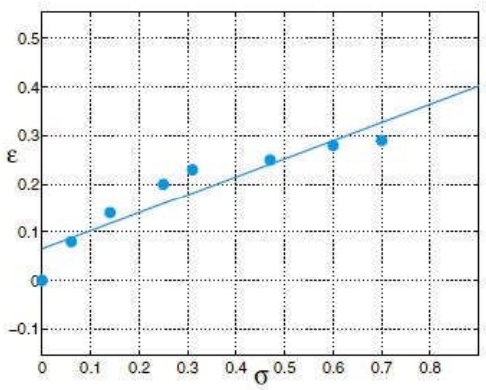
LINEAR REGRESSION

LINEAR REGRESSION

- Linear form of the least squares method.
  - n+1 functions are given.
  - Linear function y = a + b x where a and b are constants
  - Minimize the euclid norm of the function for the given data points.

APPROXIMATION

EXAMPLE:



Linear least-squares approximation of the data of Problem

LEAST SQUARES METHOD

euclid norm

$$||x||_2 = \sqrt{\sum_{j=1}^{n} |x_j|^2}$$

$$f(x_1, x_2, \ldots, x_n) = ||Ax - b||_2^2$$

$$= (Ax - b)^T (Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b$$

$$= ||(\sum_{j=1}^{n} a_{k,j} x_j - b_k)_{k=1}^{m}||_2^2$$

$$= \sum_{k=1}^{m} (\sum_{j=1}^{n} a_{k,j} x_j - b_k)^2$$

LEAST SQUARES METHOD

$$0 = \frac{df}{dx_i} = 2 \sum_{k=1}^{m} (\sum_{j=1}^{n} a_{k,j} x_j - b_k) a_{k,i}$$

$$\begin{pmatrix} a + b x_1 \\ \vdots \\ a + b x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

or

$$A \begin{pmatrix} a \\ b \end{pmatrix} = y$$

here

$$\begin{pmatrix} 1 + x_1 \\ \vdots \\ 1 + x_n \end{pmatrix}$$

$$S = \left\| A \begin{pmatrix} a \\ b \end{pmatrix} - y \right\|_2^2 = \sum_{j=1}^{n} (a + b x_j - y_j)^2$$

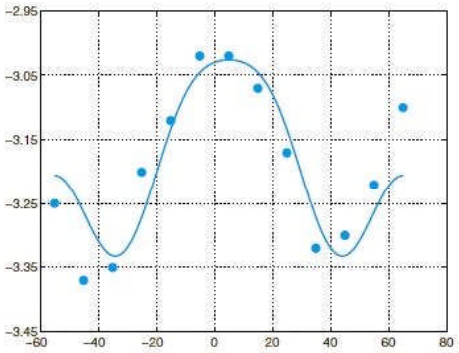
Minimize the euclid norm by finding values of a and b where the derivatives of S with respect to a and b are zero simultaneously.

$$\frac{\partial S}{a} = 2 \sum_{j=1}^{n} (a + b x_j - y_j) = 0$$

$$\frac{\partial S}{b} = 2 \sum_{j=1}^{n} x_j (a + b x_j - y_j) = 0$$

$$\begin{bmatrix} n & \sum_{j=1}^{n} x_j \\ \sum_{j=1}^{n} x_j & \sum_{j=1}^{n} x_j^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{n} y_j \\ \sum_{j=1}^{n} x_j y_j \end{bmatrix}$$

EXAMPLE:



The least-squares approximation of the data of the Problem using a cosine basis. The exact data are represented by the small circles

APPROXIMATION

EXAMPLE:

- Ali and Veli are twin boys born on October 27, 2001. Here is a table of their weights, in pounds and ounces, over their first few months.

```
% Date Tom Ben
W = [10 27 2001 5 10 4 8
     11 19 2001 7 4 5 11
     12 03 2001 8 12 6 4
     12 20 2001 10 14 8 7
     01 09 2002 12 13 10 3
     01 23 2002 14 8 12 0
     03 06 2002 16 10 13 10];
```

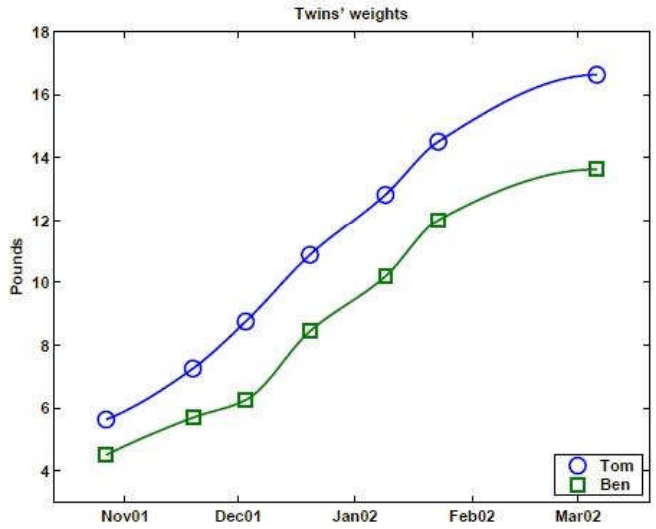

APPROXIMATION

EXAMPLE (cont'd):

- Use datenum to convert the date in the first three columns to a serial date number measuring time in days.
- t = datenum(W(:,[3 1 2]));
- Make a plot of their weights versus time, with circles at the data points and the pchip interpolating curve in between. Use datetick to relabel the time axis. Include a title and a legend.

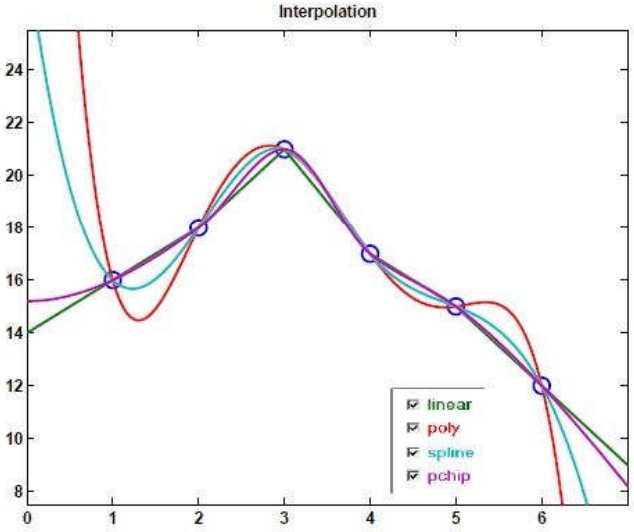
APPROXIMATION

EXAMPLE (cont'd):



APPROXIMATION

EXAMPLE:



References for Week 9

1 Alfio Quarteroni, Fausto Saleri, Scientific Computing with Matlab and Octave, Springer, 2006.

2 Moler C, NumericalComputing with Matlab, Mathworks Inc., 2004 (http://www.mathworks.com/moler).

3 Hans Rudolf Schwarz, Norbert Köckler, Numerische Mathematik, Vieweg + Teubner, 2009.

4 Thomas Huckle, Stefan Schneider, Numerische Methoden, Springer, 2006.

Have a nice Week End