Bil 108

Introduction to the Scientific and Engineering Computing with MATLAB

Lecture 7

F. Aylin Konuklar Lale T. Ergene

Basic Concepts

The general form of an *n*th degree polynomial function is

$$f(x) = a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + \ldots. + a_{n-1} x^{n-1} + a_n x^n$$

where

a0 , a1, ..., an are coefficients n denotes the degree of the polynomial

Polynomials In Matlab

Polynomial	Matlab Presentation
$8x+3$	p=[8 3]
$2x^2-4x+10$	d=[2 -4 10]
$6x^2-150$	h=[6 0 -150]
$5x^5+6x^2-7x$	c=[5 0 0 6 -7 0]

Value of Polynomials

- The value of a polynomial at a point can be calculated with the function ***polyval*** that has the form

$$polyval(p,x)$$

P is a vector with the coefficients of the polynomial

X is a number, or a variable that has an assigned value or a computable expression

Example

For the polynomial $f(x)=x^3+3x^2-5x-17$

a)Calculate f(5)

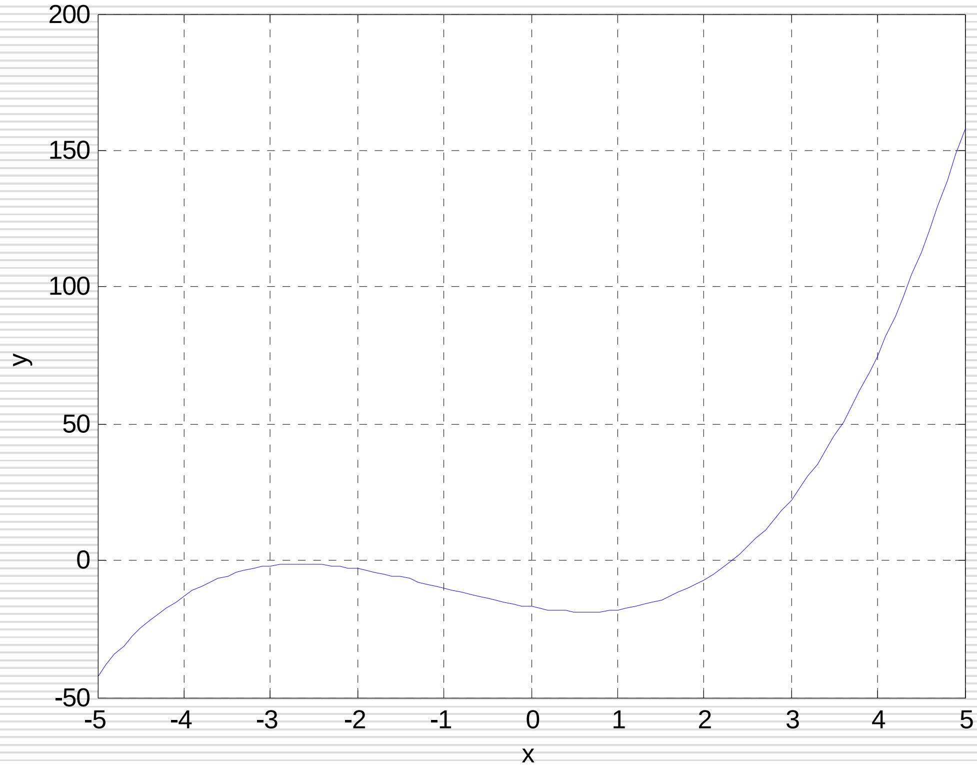
b)Plot the polynomial for $-5 \leq x \leq 5$

a)
```
>> p = [1 3 -5 -17]

p =
 1     3    -5   -17

>> polyval(p,5)

ans =

     158
```

Example

b)

```
>> x=-5:0.1:5;
>> y=polyval(p,x);
>> plot(x,y)
>>grid
>>xlabel('x')
>> ylabel('y')
```



Roots of a polynomial

Roots of a polynomial are the values of the argument for which the value of the polynomial is equal to zero.

r=roots(p)

Example:

f(x)=x3+3x2-5x-17

```
>> r=roots(p)

r =

   2.3186
  -2.6593 + 0.5099i
  -2.6593 - 0.5099i
```

Example:

$f(x)=4x^2+10x-8$

```
>>roots([4 10 -8])

ans=
-3.1375
 0.6375
```

Determining coefficients

- When the roots of a polynomial are known ***poly*** command can be used for determining the coefficients of the polynomial.

p=poly(r)

p is a column vector with the coefficients of the polynomial

r is a vector with the roots of the polynomial

Example:

```
>> r=[-2 2];
>> p=poly(r)

p =

    1    0   -4
```

Addition Multiplication and Division of Polynomials

- Two polynomials can be added and substracted by adding the vectors of the coefficients.

Example:

```
>>p1=[3 15 0 -10 -3 15 -40];
>>p2=[3 0 -2 -6];
```

- If orders are not same ->PADDING

```
>>p=p1+[0 0 0 p2]
p=
        3 15 0 -7 -3 13 -46
```

Multiplication

- Two polynomials can be multiplied with the MATLAB built in function **conv** which has the form:

$$c = conv(a,b)$$

C is a vector of the coefficients of the polynomial
That is the product of the multiplication

a and b are the vectors of the coefficients of the polynomials that are being multiplied

- The two polynomials do not have to be of the same order

- Multiplication of three or more polynomials is done by using conv function repeatedly.

Example

```
>>p1=[3 15 0 -10 -3 15 -40];
>>p2=[3 0 -2 -6];
>>pm=conv(p1,p2)
```

pm=	9	45	6	-78	-99	65	-54	-12	-10	240
	x^9	x^8	x^7	x^6	x^5	x^4	x^3	x^2	x	

Division

[q r]=deconv(u,v)

q quotient polynomial

r the remainder polynomial

u numerator polynomial

v denominator polynomial

Example:

$2x^3+9x^2+7x-6$ by $x+3$

```
>> u=[2 9 7 -6];
>>v=[1 3];
>>[a b]=deconv(u,v)
a =
    2    3    -2
b =
    0    0    0    0
```

Example:(with remainder)

```
>>w=[2 -13 0 75 2 0 -60];
>>z=[1 0 -5];
>>[g h]=deconv(w, z)
g= 2 -13 10 10 52
h= 0 0 0 0 0 50 200
```

Derivatives of Polynomials

- **k=polyder(p)** — **derivative of a single polynomial**
- **k=polyder(a,b)** — **derivative of a product of two polynomials**

Example:

$f_1(x)=3x^2-2x+4$ $\quad$ $f_2(x)=x^2+5$

```
>>f1=[3 -2 4];
>>f2=[1 0 5];
>>k=polyder(f1)

k =
6    -2

>>d=polyder(f1,f2)

d =
12    -6    38    -10
```

Curve Fitting

- When n points are given (x_i, y_i) -> a polynomial func. of degree n-1 that pass through all the points.
  - Two points ----- $y=mx+n$
  - Three points -----$y= ax^2+bx+c$
- The most common method of finding the best fit to data point is the least square methods.
- Homework: Learn least square method !!!!

Curve Fitting

- Regression analysis is a process of fitting a function to a set of data points.
- Linear Regression finds a straight line, which is a first order polynomial
- If the data doesn’t represent a straight line, a polynomial of higher order may be a better fit
- Curve fitting with polynomials is done with polyfit function which uses the least squares method.

polyfit and polyval

- ***polyfit*** finds the coefficients of a polynomial representing the data

  p=polyfit(x,y,n)

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

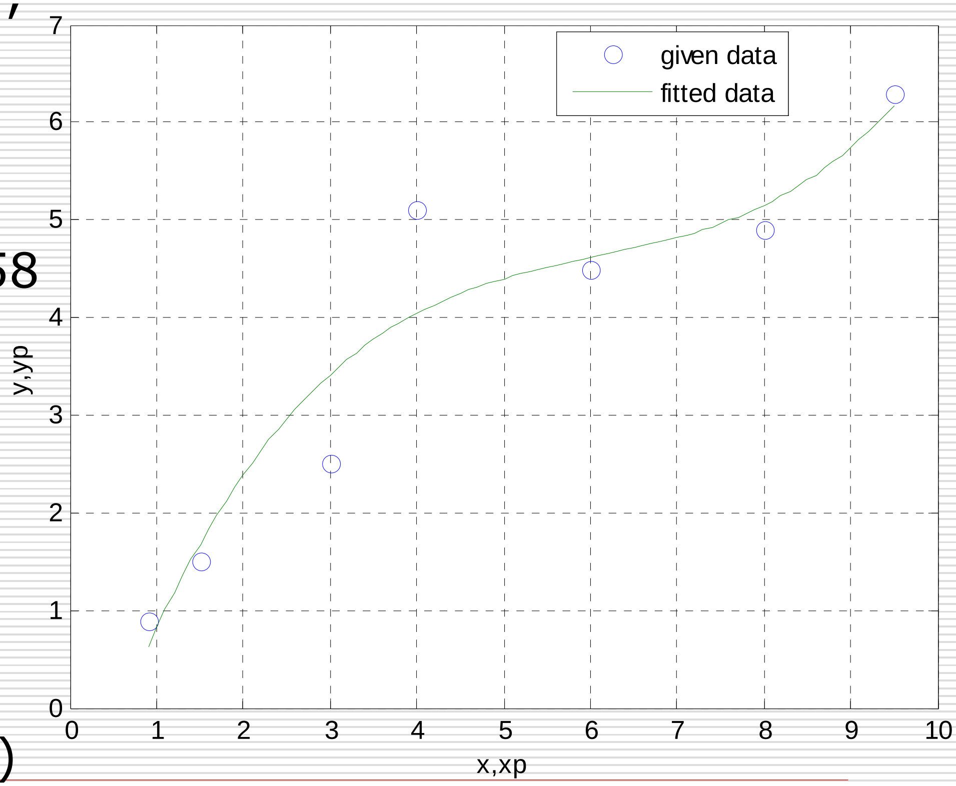
x is a vector with the horizontal coordinate
y is a vector with the vertical coordinate
n is the degree of the polynomial

- ***polyval*** uses those coefficients to find new values of y, that correspond to the known values of x

Example

- (0.9, 0.9) (1.5,1.5)(3,2.5)(4,5.1)(6,4.5) (8,4.9) (9.5,6.3) Let the points fitted using the polyfit fuction.

```
>> y=[0.9 1.5 2.5 5.1 4.5 4.9 6.3];
>> x=[0.9 1.5 3 4 6 8 9.5];
coeff=polyfit (x,y,3)
coeff =
0.0220   -0.4005    2.6138   -1.4158
>>xp=0.9:0.1:9.5;
>> yp=polyval(coeff,xp);
>> plot (x,y,'o',xp,yp)
>> grid
>> xlabel('x,xp')
>> ylabel('y,yp')
>> legend('given data','fitted data')
```



Curve Fitting with Functions Other than Polynomials

- $y=bx^m$ power function
- $y=be^{mx}$ or $y=b10^{mx}$ exponential function
- $y=m\ln(x)+b$ or $y=m\log(x)+b$ logarithmic function
- $y=1/mx+b$ reciprocal function

First rewrite the functions in a form that can be fitted with a linear polynomial (n=1)

$$y=mx+n$$

$\ln(y)=m\ln(x)+\ln(b)$ power function

$\ln(y)=mx+\ln(b)$ or $\log(y)=mx+\log(b)$ exponential function

$1/y=mx+b$ reciprocal function

Logarithmic axis scaling

Command Name	Plot type
loglog	log(y) versus log(x)
semilogx	y versus log(x)
semilogy	log(y) versus x

Other *polyfit* functions in Matlab

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)

Choosing the function

- For a given data it is possible to foresee which of the functions has the potential for providing a good fit. This is done by plotting the data using different combinations of linear and logarithmic axes.

x-axis	Y-axis	Function
linear	linear	$y=mx+b$
logarithmic	logarithmic	$y=bx^m$
linear	logarithmic	$y=be^{mx}$ $y=b10^{mx}$
logarithmic	linear	$y=m\ln(x)+b$ $y=m\log(x)+b$
linear	linear (plot 1/y)	$y=1/(mx+b)$

Other consideration when choosing a function

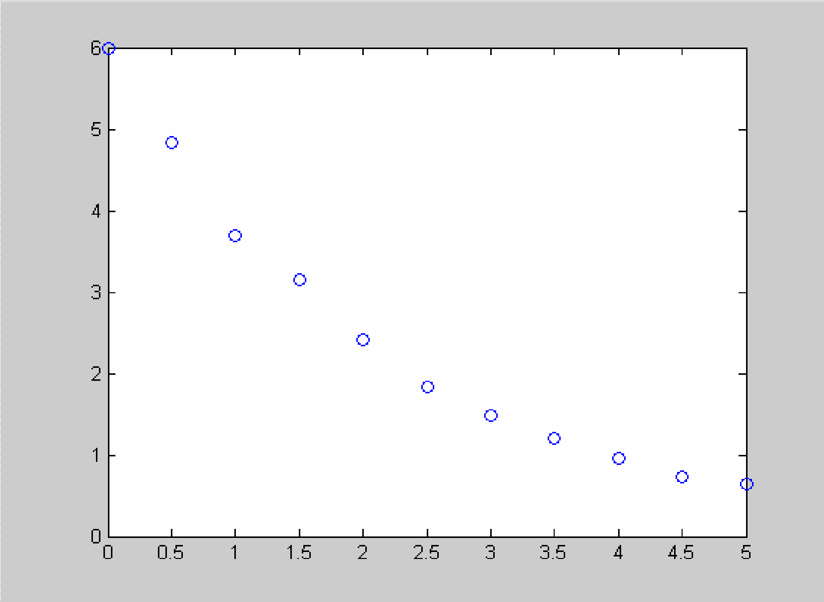
- Exponential functions cannot pass through the origin
- Exponential functions can only fit data with all positive y's or all negative y's
- Logarithmic functions cannot model x=0, or negative values of x
- For the power function y=0 when x=0
- The reciprocal equation cannot model y=0

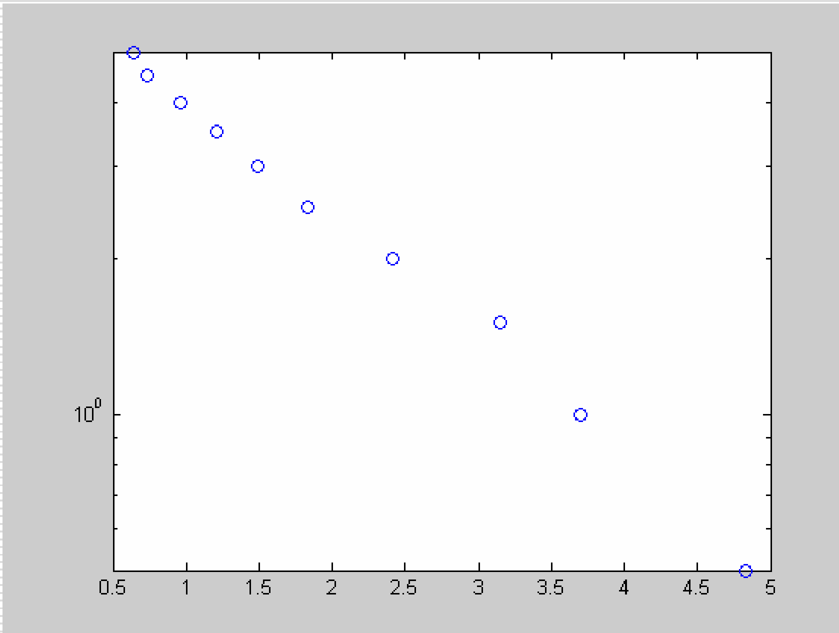
Example

t	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
w	6.00	4.83	3.70	3.15	2.41	1.83	1.49	1.21	0.96	0.73	0.64

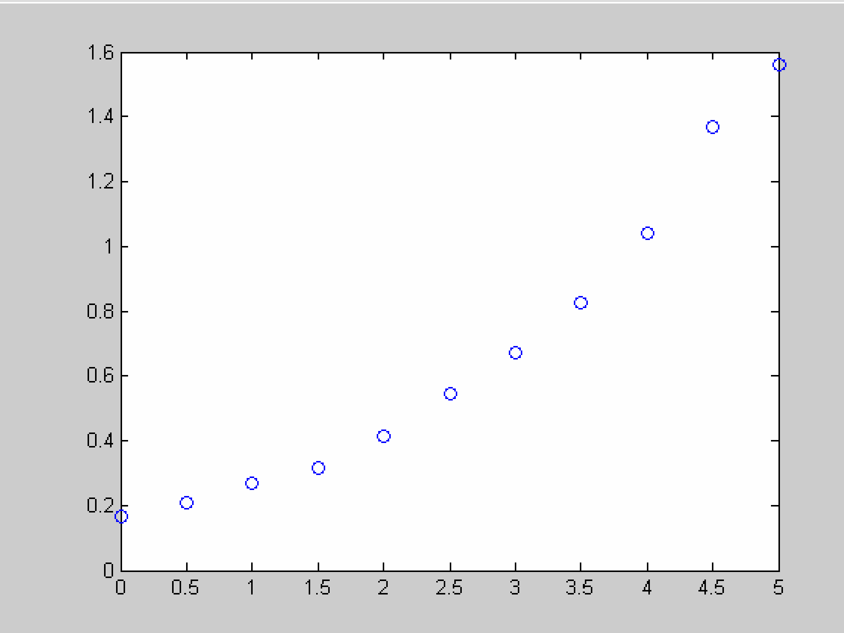
```
>> t=0:0.5:5;
>> w=[6 4.83 3.7 3.15 2.41 1.83
     1.49 1.21 0.96 0.73 0.64];
>> plot(t,w,'o')
```

- Data is first plotted with linear scales on both axis. X
- Power function X
- The logarithmic function X
- Reciprocal or exponential???





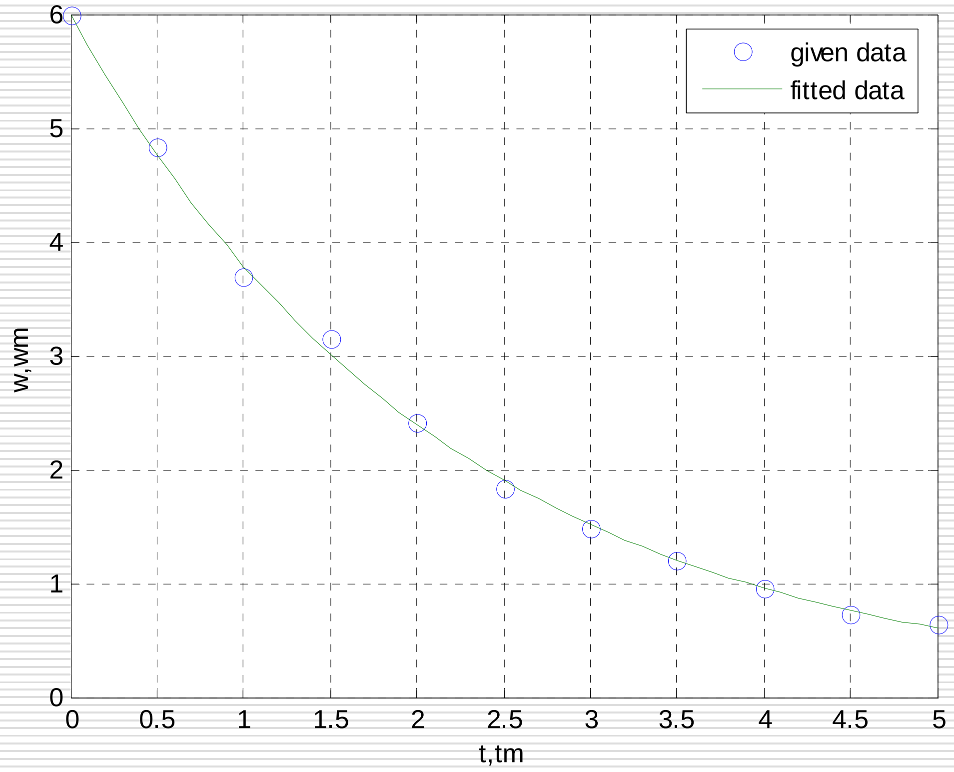
semilogy(w,t,'o')



plot(t,1./w)

```
t=0:0.5:5;
w=[6.00 4.83 3.70 3.15 2.41 1.83 1.49 1.21 0.96 0.73 0.64];
p=polyfit(t,log(w),1);
m=p(1)
b=exp(p(2))   %determine the coefficient b
tm=0:0.1:5;
wm=b*exp(m*tm);       %calculate the fnc. value at each element of tm
plot(t,w,`o`,tm,wm)       %plot the daya points and the function
grid
xlabel('t,tm')
ylabel('w,wm')
legend('given data','fitted data')
```

Example



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

Example

To calibrate an instrument six standart values were measured with it. The following table shows the instrument readings against the standart (true) values. Carry out a first-degree regression on these data and use the results to plot a calibration curve.

Measured	True
0.5030	0
0.7229	1.0000
0.7802	2.0000
1.2106	5.0000
1.7607	10.0000
2.4649	15.0000

Table 1:

```
clear
clc
m=[0.5030 0.7229 0.7802 1.2106 1.7607 2.4649]
t=[0 1 2 5 10 15]
p=polyfit(t,m,1)
x=0:0.1:15
y=polyval(p,x)
plot(t,m,'o',x,y)
title('calibration curve')
xlabel('true value')
ylabel('measured value')
```

Determining wall thickness of a box

- The outside dimensions of a rectangular box (bottom and four sides no top) made of aluminum are 24 by 12 by 4 inches. The wall thickness of the bottom and sides are x. Derive an expression that relates the weight of a box and the wall thickness x. Determine the thickness x for a box that weighs 15lb. The specific weight of aluminum is 0.101 lb/in^3