MATLAB

for Engineering Applications

Fifth Edition

William J. Palm III

Chapter 06

Model Building and Regression

Model building and regression

An important application of the plotting techniques is function discovery, the technique for using data plots to obtain a mathematical function or "mathematical model" that describes the process that generated the data.

Function Discovery: A systematic way of finding an equation that best fits the data is regression (also called the least-squares method).

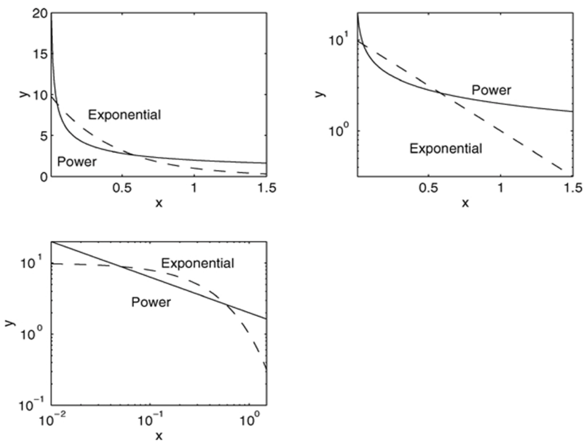
Using the Linear, Power, and Exponential Functions to Describe Data

Each function gives a straight line when plotted using a specific set of axes:

1. The linear function $y = mx + b$ gives a straight line when plotted on rectilinear axes. Its slope is m and its intercept is b.
2. The power function $y = bx^m$ gives a straight line when plotted on log-log axes.
3. The exponential function $y = b(10)^{mx}$ and its equivalent form $y = be^{mx}$ give a straight line when plotted on a semilog plot whose y-axis is logarithmic.

Function Discovery

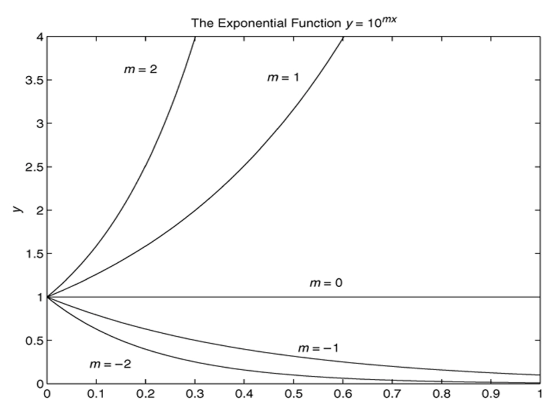
The power function $y = 2x^{-0.5}$ and the exponential function $y = 10^{1-x}$ plotted on linear, semi-log, and log-log axes.



Access the text alternative for slide images.

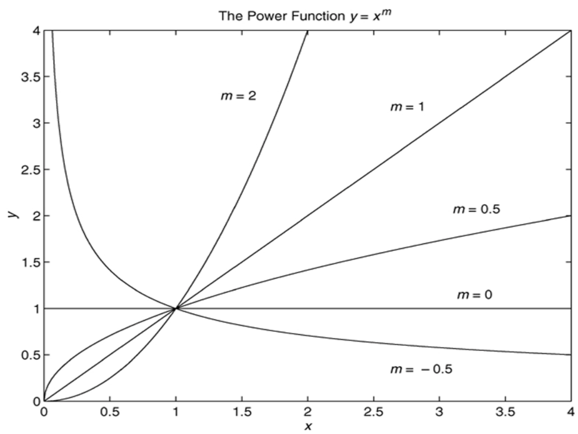
Steps for Function Discovery

1. Examine the data near the origin. The exponential function $y = b(10)^{mx}$ or $y = b(10)^{mx}$ can never pass through the origin (unless of course $b = 0$, which is a trivial case). (See Figure below for examples with $b = 1$.)



Example Plots of Exponential Functions

The linear function $y = mx + b$ can pass through the origin only if $b = 0$. The power function $y = bx^m$ can pass through the origin but only if $m > 0$. (See Figure for examples with $b = 1$.)



Steps for Function Discovery

2. Plot the data using rectilinear scales. If it forms a straight line, then it can be represented by the linear function, and you are finished. Otherwise, if you have data at $x = 0$, then

   a. If $y(0) = 0$, try the power function.

   b. If $y(0) \neq 0$, try the exponential function.

   If data is not given for $x = 0$, proceed to step 3.

3. If you suspect a power function, plot the data using log-log scales. Only a power function will form a straight line on a log-log plot. If you suspect an exponential function, plot the data using the semilog scales. Only an exponential function will form a straight line on a semilog plot.

4. In function discovery applications, we use the log-log and semilog plots *only* to identify the function type, but not to find the coefficients b and m. The reason is that it is difficult to interpolate on log scales.

Power function $y = bx^m$
Exponential function $y = b(10)^{mx}$

The `polyfit` Function

Command	Description
`p = polyfit(x,y,n)`	Fits a polynomial of degree n to data described by the vectors x and y, where x is the independent variable. Returns a row vector `p` of length n + 1 that contains the polynomial coefficients in order of descending powers.

Syntax: `p = polyfit(x,y,n)`

where x and y contain the data, `n` is the order of the polynomial to be fitted, and `p` is the vector of polynomial coefficients.

Using `polyfit` to Fit Equations to Data

The linear function: $y = mx + b$.

In this case the variables w and z in the polynomial $w = p_1 z + p_2$ are the original data variables x and y, and we can find the linear function that fits the data by typing `p = polyfit(x,y,1)`. The first element p_1 of the vector `p` will be m, and the second element p_2 will be b.

The Power Function

The power function: $y = bx^m$.

In this case

$$\log_{10} y = m \log_{10} x + \log_{10} b$$

which has the form

$$w = p_1 z + p_2$$

where the polynomial variables w and z are related to the original data variables x and y by $w = \log_{10} y$ and $z = \log_{10} x$. Thus, we can find the power function that fits the data by typing

```
p = polyfit(log10(x),log10(y),1)
```

The first element p_1 of the vector `p` will be m, and the second element p_2 will be $\log_{10} b$. We can find b from $b = 10^{p2}$.

The Exponential Function

The exponential function: $y = b(10)^{mx}$.

In this case

$$\log_{10} y = mx + \log_{10} b$$

which has the form

$$w = p_1 z + p_2$$

where the polynomial variables w and z are related to the original data variables x and y by $w = \log_{10} y$ and $z = x$. We can find the exponential function that fits the data by typing

```
p = polyfit(x, log10(y),1)
```

The first element p_1 of the vector `p` will be m, and the second element p_2 will be $\log_{10} b$. We can find b from $b = 10^{p2}$.

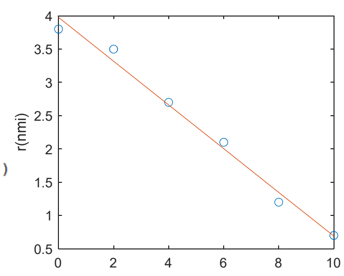
Fitting a Linear Function: Speed Estimation from Sonar Range Measurements: Example 6.1-1

Sonar measurements of the range of an approaching underwater vehicle are given in the following table, where the distance is measured in nautical miles (nmi). Assuming the relative speed v is constant, the range as function of time is given by $r = -vt + r_0$ where r_0 is the initial range at $t = 0$. Estimate the speed v and when the range will be zero.

Time, t (min)	0	2	4	6	8	10
Range, r (nmi)	3.8	3.5	2.7	2.1	1.2	0.7

The speed v = the slope = 0.3286 nmi/min = 0.3286 nmi/min x 60min/hr = 19.7143 nmi/hr

```
>> t=0:2:10;
>> r=[3.8,3.5,2.7,2.1,1.2,0.7];
>> %First order curve fit.
>> p=polyfit(t,r,1)
p =
   -0.3286    3.9762
>> %Create plotting variable.
>> rp=p(1)*t+p(2);
>> plot(t,r,'o',t,rp),xlabel('t(min)'),ylabel('r(nmi)')
>> %Speed calculation.
>> v=-p(1)*60%Speed in knots(nmi/hr)
v =
    19.7143
```



From the regression model r = -0.3286 t + 3.9762 when r = 0 then t = 12.1 min

Example: Fitting an exponential function

The temperature of coffee cooling in a mug at room temperature (68°F) was measured at various times. The data follow.

Time t (sec)	Temperature T (°F)
0	145
620	130
2266	103
3482	90

Develop an exponential function model of the coffee's temperature as a function of time, and use the model to estimate how long it took the temperature to reach 120°F.

```
% Enter the data.
time = [0,620,2266,3482];
temp = [145,130,103,90];
% Subtract the room temperature.
temp = temp - 68;
% Plot the data on rectilinear scales.
subplot(2,2,1)
plot(time,temp,time,temp,'o'),xlabel('Time (sec)'),...
    ylabel('Relative Temperature (deg F)')
%
% Plot the data on semilog scales.
subplot(2,2,2)
semilogy(time,temp,time,temp,'o'),xlabel('Time (sec)'),...
    ylabel('Relative Temperature (deg F)')
```

The data can be described with the Exponential function: $T = 68 + b(10)^{mt}$

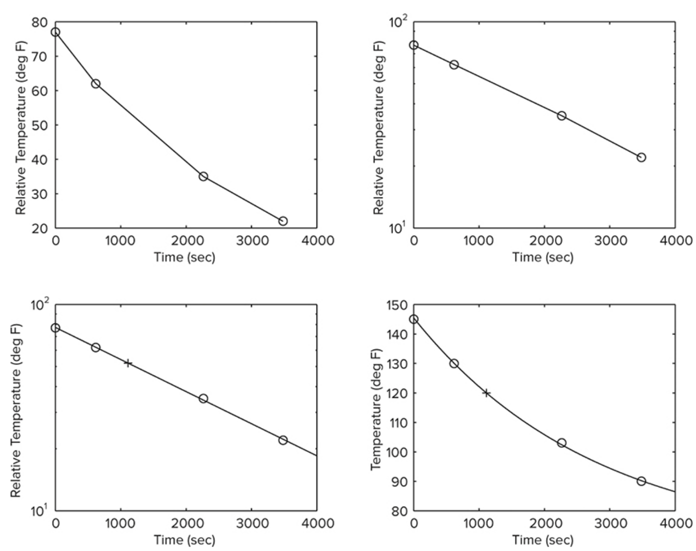
```
% Fit a straight line to the transformed data.
p = polyfit(time,log10(temp),1);
m = p(1)
b = 10^p(2)
```

The computed values are m = -1.5557x10-4 and b = 77.4469.
Thus, the derived Model is $T = 68 + b(10)^{mt}$
To estimate how long it will take for the coffee to cool to 120ºF. We must solve the equation: $120 = 68 + b(10)^{mt}$ for t.
The solution is $t = \left[\log_{10}(120-68) - \log_{10}(b)\right]/m$

```
% Compute the time to reach 120 degrees.
t_120 = (log10(120-68)-log10(b))/m
% Show derived curve and estimated point on semilog scales.
t = 0:10:4000;
T = 68+b*10.^(m*t);
subplot(2,2,3)
semilogy(t,T-68,time,temp,'o',t_120,120-68,'+'),
xlabel('Time (sec)'),...
   ylabel('Relative Temperature (deg F)')
%
% Show derived curve and estimated point on linear scales.
subplot(2,2,4)
plot(t,T,time,temp+68,'o',t_120,120,'+'),xlabel('Time (sec)'),...
   ylabel('Temperature (deg F)')
```

The computed value of t_120 is 1113. Thus, the time to reach 120oG is 1112 sec.

Fitting an Exponential Function: Temperature of a Cooling Cup of Coffee



Question 1

Q1-The distance a spring stretches from its "free length" is a function of how much tension force is applied to it. The following table gives the spring length y that the given applied force f produced in a particular spring. The spring's free length is 4.7 in. Find a functional relation between f and x, the extension from the free length, $f = kx + b$. The spring constant k is:

Force f (lb)	Spring length y (in.)
0	4.7
0.94	7.2
2.30	10.6
3.28	12.9

a) 0.1
b) 0.2
c) 0.3
d) 0.4 (correct)

```
>> f=[0,0.94,2.3,3.28];
>> y=[4.7,7.2,10.6,12.9];
>> x=y-4.7;
>> p=polyfit(x,f,1)
p =
   0.3998   -0.0294
>> p(1)
ans =
    0.3998
```


Question 2

Q2. The population data for a certain country are as follows:

Year	2012	2013	2014	2015	2016	2017
Population (millions)	10	10.9	11.7	12.6	13.8	14.9

Obtain an exponential function that describes these data. How many years after 2012 will the population be double its 2012 size (20 millions)?

```
>> x=[2012,2013,2014,2015,2016,2017];
>> x=x-2012;
>> pop=[10,10.9,11.7,12.6,13.8,14.9];
>> p=polyfit(x,log10(pop),1);
>> p
p =

    0.0344    1.0004
>> x=(1/0.0344)*log10(20/(10^1.0004))

x =

    8.7392
```

$$10^{1.0004}(10)^{0.0344x} = 20$$

$$0.0344x = \log\left(\frac{20}{10^{1.0004}}\right)$$

$$x = \frac{1}{0.0344} \times \log\left(\frac{20}{10^{1.0004}}\right)$$

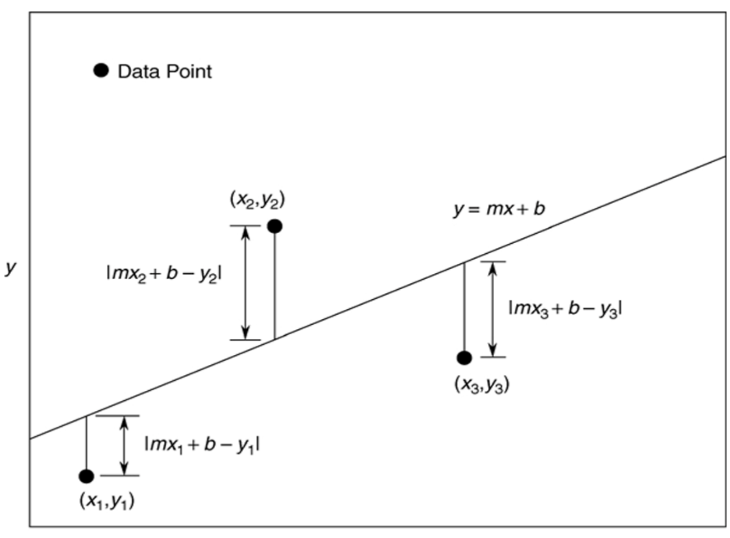
The Least Squares Criterion

The Least Squares Criterion: used to fit a function $f(x)$. It minimizes the sum of the squares of the residuals, J. J is defined as

$$J = \sum_{i=1}^{m} \left[f(x_i) - y_i \right]^2$$

We can use this criterion to compare the quality of the curve fit for two or more functions used to describe the same data. The function that gives the smallest J value gives the best fit.

Illustration of the Least Squares Criterion



Access the text alternative for slide images.

The `polyfit` Function is Based on the Least Squares Method

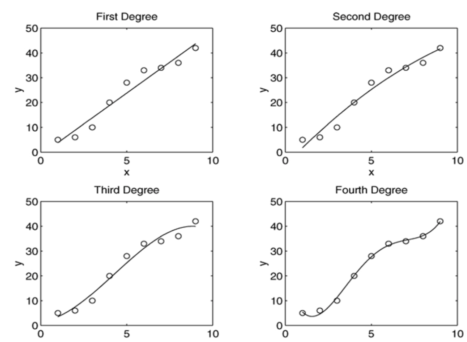
Its syntax is

`p = polyfit(x,y,n)` — Fits a polynomial of degree n to data described by the vectors x and y, where x is the independent variable. Returns a row vector p of length $n + 1$ that contains the polynomial coefficients in order of descending powers.

Regression Using Polynomials of First through Fourth Degree

Consider the data set where $x = 1, 2, 3, \ldots, 9$ and $y = 5, 6, 10, 20, 28, 33, 34, 36, 42$. Fit polynomials of first through fourth degree to this data and compare the results.

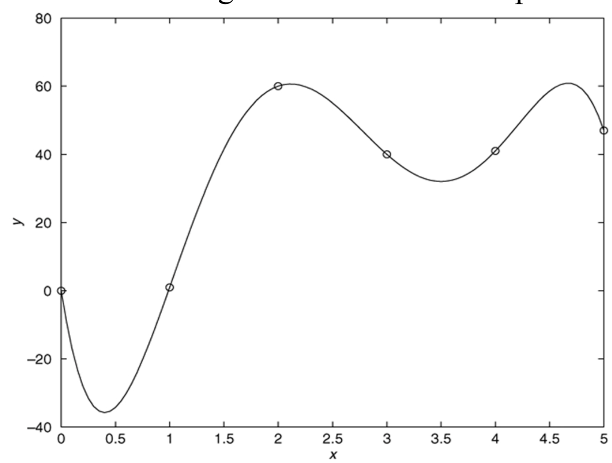
```
>> x=1:9;
>> y=[5,6,10,20,28,33,34,36,42];
>> for k=1:4
coeff=polyfit(x,y,k);
J(k)=sum((polyval(coeff,x)-y).^2);
subplot(2,2,k);
plot(x,y,'o');
hold on
plot(x,polyval(coeff,x));
end
>> J
J =
   71.5389    56.6727    41.8838     4.6566
```



Fourth order is the best option with a minimized error!

Beware of Using Polynomials of High Degree

This fifth-degree polynomial passes through all six data points but exhibits large excursions between points.



Assessing the Quality of a Curve Fit

Denote the sum of the squares of the deviation of the y values from their mean y by S, which can be computed from

$$S = \sum_{i=1}^{m} (y_i - \bar{y})^2 \qquad (6.2\text{-}2)$$

This formula can be used to compute another measure of the quality of the curve fit, the *coefficient of determination,* also known as the *r-squared value.* It is defined as

$$J = \sum_{i=1}^{m} \left[f(x_i) - y_i \right]^2 \qquad S = \sum_{i=1}^{m} (y_i - \bar{y})^2 \qquad r^2 = 1 - \frac{J}{S}$$

The value of S indicates how much the data is spread around the mean, and the value of J indicates how much of the data spread is unaccounted for by the model.

Thus, the ratio J/S indicates the fractional variation unaccounted for by the model.

Assessing the Quality of a Curve Fit 2

For a perfect fit, $J = 0$ and thus $r^2 = 1$. Thus, the closer r^2 is to 1, the better the fit. The largest r^2 can be is 1.

It is possible for J to be larger than S, and thus it is possible for r^2 to be negative. Such cases, however, are indicative of a very poor model that should not be used.

As a rule of thumb, a good fit accounts for at least 99 percent of the data variation. This value corresponds to $r^2 \geq 0.99$.

Scaling the Data

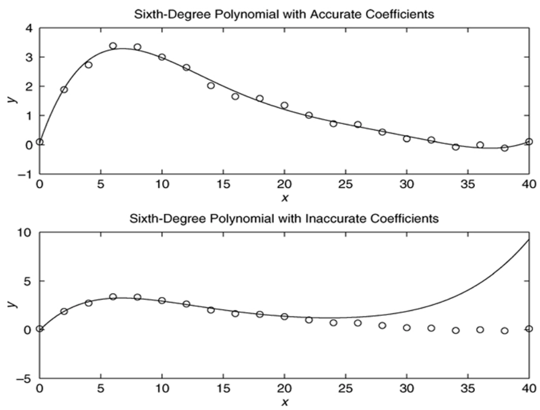
The effect of computational errors in computing the coefficients can be lessened by properly scaling the x values. You can scale the data yourself before using `polyfit`. Some common scaling methods are

1. Subtract the minimum x value or the mean x value from the x data, if the range of x is small, or
2. Divide the x values by the maximum value or the mean value, if the range is large.

Effect of Coefficient Accuracy on a Sixth-Degree Polynomial

Top graph: 14 decimal-place accuracy

Bottom graph: 8 decimal-place accuracy



- High-degree polynomials can produce large errors if their coefficients are not represented with a large number of significant figures.

Example: Scaling the data

The following data give the number of vehicles (in millions) crossing a bridge each year for 10 years. Fit a cubic polynomial to the data and use the fit to estimate the flow in the year 2010.

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Vehicle flow(millions)	2.1	3.4	4.5	5.3	6.2	6.6	6.8	7	7.4	7.8

```
>> Year=2000:2009;
>> Veh_Flow=[2.1,3.4,4.5,5.3,6.2,6.6,6.8,7,7.4,7.8];
>> p=polyfit(Year,Veh_Flow,3);
Warning: Polynomial is badly conditioned. Add points with
distinct X values, reduce the degree of the polynomial, or
try centering and scaling as described in HELP POLYFIT.
> In polyfit (line 72)
```

No meaningful solution/formulation, so it may need scaling!

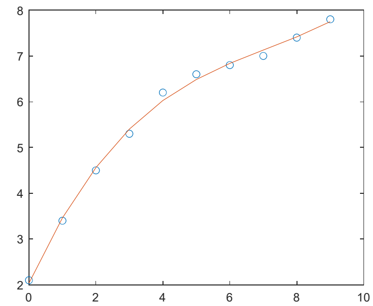
Example: Scaling the data

Correct solution using scaling!

```
>> Year=2000:2009;
>> Veh_Flow=[2.1,3.4,4.5,5.3,6.2,6.6,6.8,7,7.4,7.8];
>> %Scaling the "Year" as a new "x"
>> x=Year-2000;
>> %Calling "Veh_Flow" as "y" for simplicity
>> y=Veh_Flow;
>> p=polyfit(x,y,3);
>> J=sum((polyval(p,x)-y).^2);
>> S=sum((y-mean(y)).^2);
>> r2=1-J/S;
>> plot(x,y,'o',x,polyval(p,x))
>> r2
r2 =
```

Acceptable r^2

```
    0.9972
>> p
p =

    0.0087   -0.1851    1.5991    2.0362
```



$$f(t) = 0.0087(t-2000)^3 - 0.1851(t-2000)^2 + 1.5991(t-2000) + 2.0362$$

$f(2010)$=8.21

Question 3

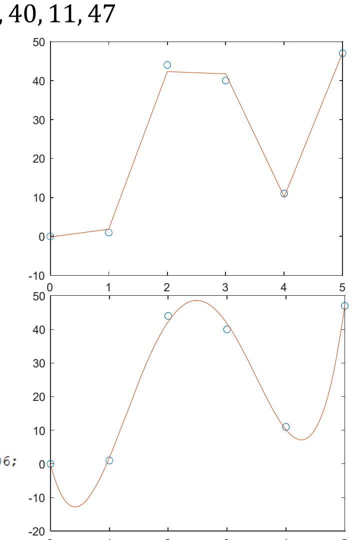
Q3 -The plot of a fourth-degree polynomial for the following data is:

$x = 0, 1, \ldots, 5$ and $y = 0, 1, 44, 40, 11, 47$

```
>> x=0:5;
>> y=[0,1,44,40,11,47];
>> p=polyfit(x,y,4);
>> p
p =

    3.7292  -35.6157   99.9514  -66.0410   -0.1706
>> plot(x,y,'o',x,polyval(p,x))
>> J=sum((polyval(p,x)-y).^2);
>> S=sum((y-mean(y)).^2);
>> r2=1-J/S;
>> r2

r2 =
```

Acceptable r^2

```
    0.9970

>> xp=0:0.01:5;
>> yp=3.7292*(xp.^4)-35.6157*(xp.^3)+99.9514*(xp.^2)-66.041.*(xp)-0.1706;
>> plot(x,y,'o',xp,yp)
```



Question 4

Q4- The U.S. census data from 1790 to 1990 is stored in the file census. dat, which is supplied with MATLAB. Type load census to load this file. The first column, cdate, contains the years, and the second column, pop, contains the population in millions. First try to fit a cubic polynomial to the data. If you get a warning message, scale the data by subtracting 1790 from the years, and fit a cubic. The coefficient of determination (r-squared value) is:

```
>> load census
```

Variables - cdate

cdate | pop

21x1 double

	1	2
1	1790	
2	1800	
3	1810	
4	1820	

```
>> load census
>> x=cdate-1790;
>> y=pop;
>> p=polyfit(x,y,3)
p =

    0.0000    0.0054   -0.0022    4.2644

>> pop1965=polyval(p,1965-1790)
pop1965 =

  189.4417
>> J=sum((polyval(p,x)-y).^2);
>> S=sum((y-mean(y)).^2);
```

Example pop in 1965

```
>> r2=1-J/S;
>> r2

r2 =

    0.9988
```

Acceptable r^2

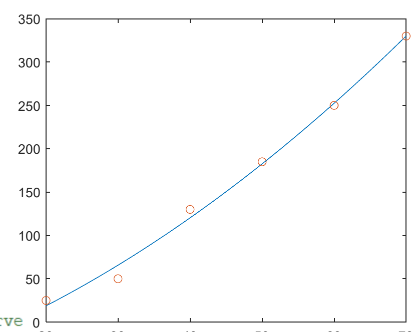
Question 5

Q5- The following data give the stopping distance d as a function of initial speed v, for a certain car model. Find a quadratic polynomial that fits the data. Determine the quality of the curve fit by computing r^2.

$v\ (mi/hr)$	20	30	40	50	60	70
$d\ (ft)$	25	50	130	185	250	330

a) 0.9921
b) 0.9943 (correct)
c) 0.9962
d) 0.9998

```
>> x=20:10:70;
>> y=[25,50,130,185,250,330];
>> p=polyfit(x,y,2);
>> J=sum((polyval(p,x)-y).^2);
>> S=sum((y-mean(y)).^2);
>> r2=1-J/S;
>> r2
r2 =

    0.9943
>> %Ploting the data and fitted curve
>> xp=20:0.01:70;
>> yp=polyval(p,xp);
>> plot(xp,yp,x,y,'o'),axis([20 70 0 350])
```



Question 6

Q6-The following function is linear in the parameters a_1 and a_2.

$$y(x) = a_2 + a_1 \ln(x).$$

Use least-squares regression with the following data to estimate the values of a_1 and a_2.

x	1	2	3	4	5	6	7	8	9	10
y	10	14	16	18	19	20	21	22	23	23

a) $a_1 = 14.4353$ $a_2 = 2.4543$
b) $a_1 = 7.34764$ $a_2 = 6.6895$
c) $a_1 = 5.7518$ $a_2 = 9.9123$ (correct)
d) $a_1 = 3.4344$ $a_2 = 11.472$

```
>> x=1:10;
>> y=[10,14,16,18,19,20,21,22,23,23];
>> %Defining Ln(x)as a variable
>> logx=log(x);%log(x)=Ln(x)
>> coeff=polyfit(logx,y,1);%y=a1+a2(logx)is first order.
>> coeff
coeff =

    5.7518    9.9123
```

```
>> %What is the function value in x=3.5?
>> value=coeff(1)*log(3.5)+coeff(2)

value =

   17.1179
```


Multiple linear regression

Suppose that y is a linear function of two or more variables $x1, x2, \ldots,$

for example, $y = a_0 + a_1 x_1 + a_2 x_2$.

How to find the coefficient values a_0, a_1, and a_2 to fit a set of data (y, x_1, x_2) in the least-squares sense?

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} \\ 1 & x_{12} & x_{22} \\ 1 & x_{13} & x_{23} \\ \cdots & \cdots & \cdots \\ 1 & x_{1n} & x_{2n} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_n \end{bmatrix}$$

The solution for the coefficients is given by **a** = **X\y**.

Example: Multiple linear regression

Obtain a linear model $y = a_0 + a_1x_1 + a_2x_2$ to describe the relationship.

Breaking strength (kN)	% of element 1	% of element 2
y	x_1	x_2
7.1	0	5
19.2	1	7
31	2	8
45	3	11

```
>> x1=[0;1;2;3];
>> x2=[5;7;8;11];
>> %Making matrix X from x1 and x2
>> firstcolumn=ones(size(x1))
firstcolumn =

     1
     1
     1
     1
>> X=[firstcolumn,x1,x2]
X =

     1     0     5
     1     1     7
     1     2     8
     1     3    11
>> y=[7.1;19.2;31;45];
>> a=X\y
```

```
a =

    0.8000
   10.2429
    1.2143
```


Linear-in-parameters regression

Sometimes we want to fit an expression that is neither a polynomial nor a function that can be converted to linear form by a logarithmic or other transformation.

In some cases, we can still do a least-squares fit if the function is a linear expression in terms of its parameters.

Example: The following data gives the output voltage of a certain device as a function of time. Obtain a function that describes this data.

$$v(t) = a_1 + a_2e^{-t/T} \text{ (first-order model)}$$

The first-order model written for each of the n data points results in n equations, which can be expressed as follows:

$$\begin{bmatrix} 1 & e^{-t_1} \\ 1 & e^{-t_2} \\ \cdots & \cdots \\ 1 & e^{-t_n} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix}$$

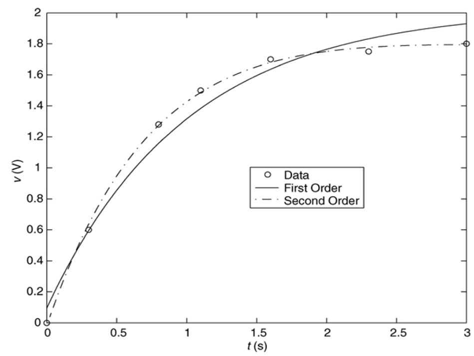
Xa = y′

Example: Linear-in-parameters regression

$$y(t) = a_1 + a_2 e^{-t} \text{ (first-order model)}$$

```
>> t=[0,0.3,0.8,1.1,1.6,2.3,3];
>> y=[0,0.6,1.28,1.5,1.7,1.75,1.8];
>> X=[ones(size(t));exp(-t)]';%To generate X matrix (last slide)
>> a=X\y'

a =

    2.0258
   -1.9307

>> plot(t,y,'o')
>> hold on
>> yp=a(1)+a(2).*exp(-t);
>> plot(t,yp)
```

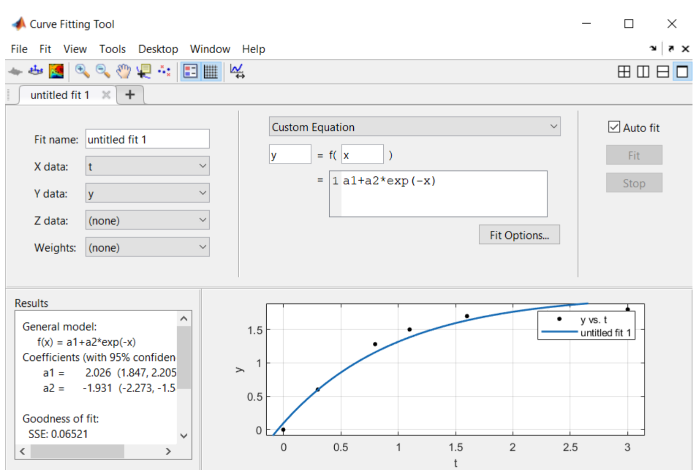
Data for t and y



Cftool

```
>> t=[0,0.3,0.8,1.1,1.6,2.3,3];
>> y=[0,0.6,1.28,1.5,1.7,1.75,1.8];
>> cftool
```

$$y(t) = a_1 + a_2 e^{-t} \text{ (first-order model)}$$



Question 7

Q7- Obtain a linear model $y = a_0 + a_1 x_1 + a_2 x_2$ for the following data to describe the relationship. The maximum error is

y	*x1*	*x2*
3.8	7.5	6
5.6	6	9
6	13.5	10.5
5	16.5	18
5.8	19.5	21
5.6	21	25.5

a) Max_error = 1.0847 (correct)
b) Max_error = 2.1249
c) Max_error = 3.8926
d) Max_error = 4.3285

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} \\ 1 & x_{12} & x_{22} \\ 1 & x_{13} & x_{23} \\ \cdots & \cdots & \cdots \\ 1 & x_{1n} & x_{2n} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_n \end{bmatrix}$$

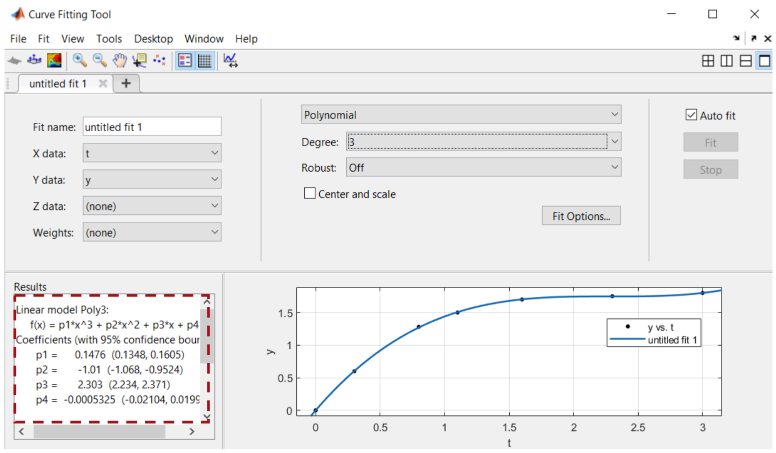
Question

```
>> y=[3.8,5.6,6,5,5.8,5.6]';
>> x1=[7.5,6,13.5,16.5,19.5,21]';
>> x2=[6,9,10.5,18,21,25.5]';
>> X=[ones(size(x1)),x1,x2];
>> a=X\y

a =

    4.5242
    0.0264
    0.0271

>> yp=X*a;
>> Max_percent_error=100*max(abs((yp-y)./y))

Max_percent_error =

   28.5455

>> Max_error=max(abs(yp-y))

Max_error =

    1.0847
```

Question 8

Q8- Using cftool toolbox, find a polynomial function that fits following t and y data.

```
>> t=[0,0.3,0.8,1.1,1.6,2.3,3];
>> y=[0,0.6,1.28,1.5,1.7,1.75,1.8];
```



Using Residuals

The following table gives data on the growth of a certain bacteria population with time. Fit an equation to these data.

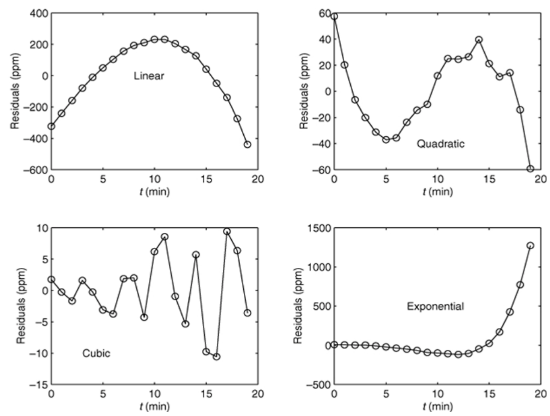
```
% Time data
x = 0:19;
% Population data
y = [6,13,23,33,54,83,118,156,210,282,...
    350,440,557,685,815,990,1170,1350,1575,1830];
% Linear fit
p1 = polyfit(x,y,1);
% Quadratic fit
p2 = polyfit(x,y,2);
% Cubic fit
p3 = polyfit(x,y,3);
% Exponential fit
p4 = polyfit(x,log10(y),1);
% Residuals
res1 = polyval(p1,x) - y;
res2 = polyval(p2,x) - y;
res3 = polyval(p3,x) - y;
res4 = 10.^polyval(p4,x) - y;
```

Time (min)	Bacteria (ppm)
0	6
1	13
2	23
3	33
4	54
5	83
6	118
7	156
8	210
9	282
10	350
11	440
12	557
13	685
14	815
15	990
16	1170
17	1350
18	1575
19	1830

Using Residuals

Residual plots of four models

Residuals are differences between the one-step-predicted output from the model and the measured output from the validation data set. Thus, residuals represent the portion of the validation data not explained by the model.



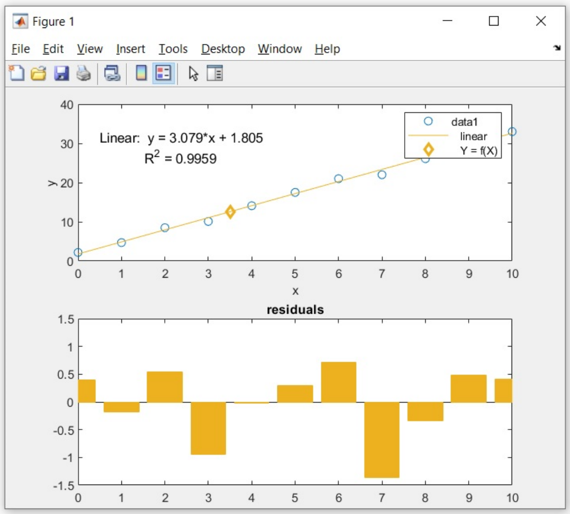
The cubic is the best fit of the four models considered. Its coefficient of determination is $r^2 = 0.9999$. The model is $y = 0.1916t^3 + 1.2082t^2 + 3.607t + 7.7307$

Basic Fitting Interface

MATLAB supports curve fitting through the Basic Fitting interface. Using this interface, you can quickly perform basic curve fitting tasks within the same easy-to-use environment. The interface is designed so that you can:

- Fit data using a cubic spline or a polynomial up to degree 10.
- Plot multiple fits simultaneously for a given data set.
- Plot the residuals.
- Examine the numerical results of a fit.
- Interpolate or extrapolate a fit.
- Annotate the plot with the numerical fit results and the norm of residuals.
- Save the fit and evaluated results to the MATLAB workspace.

A Figure Produced by the Basic Fitting Interface



Source: MATLAB

Because learning changes everything.®

www.mheducation.com