```
%Non_Linear_Linearized
%This Worksheet demonstrates the use of Matlab to illustrate the procedure
%to regress three different nonlinear models using data linearization.
%
%The desired model can be selected by giving a model number (m):
%m=1: Exponential
%m=2: Power
%m=3: Saturation or Growth
%Select your model and run the program:
              m=1;
%Input arrays X and Y:
X=[0.5,1,3,5,7,9];
Y=[1,0.892,0.708,0.562,0.447,0.355];
clc


disp(sprintf('Nonlinear Regression with Data Linearization'))
disp(sprintf('©2007 Fabian Farelo, Autar Kaw'))
disp(sprintf('University of South Florida'))
disp(sprintf('United States of America'))
disp(sprintf('kaw@eng.usf.edu'))

disp(sprintf('\n\nNOTE: This worksheet demonstrates the use of Matlab to illustrate the procedure'))
disp(sprintf('to regress a given data set to a nonlinear model with linearization of data.'))
%-------------------------------------------------------------------
disp(sprintf('\n\n**************************** Introduction ******************************'))

disp(sprintf('This worksheet illustrates finding the constants of nonlinear regression '))
disp(sprintf('models with data linearization. Three common nonlinear models are illustrated-'))
disp(sprintf('1)Exponential  2) Power  3)Saturation Growth.'))
disp(sprintf('\nGiven n data points (x1,y1), (x2,y2), (x3,y3), ... , (xn,yn), best fit one of the'))
disp(sprintf('following models to the data.'))
disp(sprintf('\nExponential:  y=a*exp(b*x) '))
disp(sprintf('Power:  y = a*x^b'))
disp(sprintf('Growth:  y = (a*x)/(b+x)'))
disp(sprintf('\na and b are the constants of the above regression models.'))
disp(sprintf('\nThe input arrays X and Y are:'))
X
Y
if m==1
%--------- Exponential-----------------
disp(sprintf('\n\n**************************** Exponential  ******************************'))

disp(sprintf('\nIn order to linearize the data of an exponential model, you must first take the natural'))
disp(sprintf('log of both sides.'))
disp(sprintf('The Natural log of y = a*exp(b*x) yields:'))
disp(sprintf('\n           ln(y) = ln(a) +b*x                        (2)'))
disp(sprintf('\nThe following substitutions are then made.'))
disp(sprintf('Let   z = ln(y) , \n      a0= ln(a),  implying a=exp(a0) \nand a1=b '))
disp(sprintf('\nthen'))
disp(sprintf('               z=a0+a1*x                               (3)'))
disp(sprintf('\nThe data z versus x now takes the form of a linear model. The constants a0 and a1'))
disp(sprintf('can be found using the least squares regression method. They are then used'))
disp(sprintf('to determine the original constants of the exponential model a and b, where'))
disp(sprintf(' y =(a*x)/(b+x).'))
```

```
n=length(X);
Z=zeros(1,n);
for i=1:n
   Z(i)=log(Y(i));
end
Z;
xav=sum(X)/n;
zav=sum(Z)/n;
sum(Z);
Sxz=0;
Sxx=0;
for i=1:n
   Sxz=Sxz +X(i)*Z(i)-xav*zav;
   Sxx=Sxx + (X(i))^2-xav^2;
end

Sxx;
Sxz;

a1=Sxz/Sxx
a0=zav-a1*xav
disp(sprintf('Now since a0 and a1 are found, the original constants of the exponential model a and b '))
disp(sprintf('are found as'))
a=exp(a0)
b=a1

disp(sprintf('\nThe model is described as\n                 y=%5g',a))
disp(sprintf('\be^(%5g',b))
disp(sprintf('\b*x)                     (4)'))

xp=(0:0.001:max(X));
yp=zeros(1,length(xp));
for i=1:length(xp)
yp(i)=a.*exp(b*xp(i));
end
plot(xp,yp)
xlabel('x')
ylabel('y=a*exp(b*x)')
hold on
plot(X,Y,'bo','MarkerFaceColor','b')
hold off
end


if m==2
%----------- Power ------------------------
disp(sprintf('\n\n*************************** Power  *****************************'))
disp(sprintf('\nIn order to linearize the data of a power model, you must first take the natural'))
disp(sprintf('log of both sides.'))
disp(sprintf('The Natural log of y = a*x^b yields:'))
disp(sprintf('\n             ln(y) = ln(a) +b*ln(x)                              (2)'))
disp(sprintf('\nThe following substitutions are then made.'))
disp(sprintf('Let   z = ln(y) , \n       w = ln(x) \n       a0= ln(a),  implying a=exp(a0) \nand a1=b '))
disp(sprintf('\nthen'))
disp(sprintf('                 z=a0+a1*w                                       (3)'))
disp(sprintf('\nThe data z versus w now takes the form of a linear model. Least squares linear'))
```

```
disp(sprintf('regression method can be used to solve for the a0 and a1 coefficients which are then used'))
disp(sprintf('to determine the original constants of the power model a and b, where y = a*x^b.'))


   n=length(X);
   Z=zeros(1,n);
for i=1:n
   Z(i)=log(Y(i));
end
w=zeros(1,n);
for i=1:n
   w(i)=log(X(i));
end
wav=sum(w)/n;
zav=sum(Z)/n;
sum(Z);
Swz=0;
Sww=0;
for i=1:n
   Swz=Swz +w(i)*Z(i)-wav*zav;
   Sww=Sww + (w(i))^2-wav^2;
end

Sww;
Swz;

a1=Swz/Sww
a0=zav-a1*wav
disp(sprintf('Now since a0 and a1 are found, the original constants of the model'))
disp(sprintf('are found as'))
a=exp(a0)
b=a1

disp(sprintf('\nThe model is described as\n                y=%5g',a))
disp(sprintf('\b*x^%5g',b))
disp(sprintf('\b                 (5)'))

xp=(0:0.001:max(X));
yp=zeros(1,length(xp));
for i=1:length(xp)
yp(i)=a.*(xp(i)^b);
end
plot(xp,yp)
xlabel('x')
ylabel('y=a*x^b')
hold on
plot(X,Y,'bo','MarkerFaceColor','b')
hold off
end
%--------- Growth -------------------

if m==3
disp(sprintf('\n\n*************************** Growth  *****************************'))
  disp(sprintf('\nIn order to linearize the data of a saturation growth model y = (a*x)/(b+x), you must↙
first take'))
disp(sprintf('the reciprocal of both sides and then rearrange them to yield:'))
```

```
disp(sprintf('\n            1/y = (b/a)*(1/x) + (1/a)                       (2)'))
disp(sprintf('\nThe following substitutions are then made.'))
disp(sprintf('Let  z = 1/y , \n      q = 1/x \n      a0= 1/a,  implying a=1/a0 \nand a1=b/a,   implying b =↙
a1/a0 '))
disp(sprintf('\nthen'))
disp(sprintf('                 z=a0+a1*q                                 (3)'))
disp(sprintf('\nThe data z versus q now takes the form of a linear model. Least squares linear'))
disp(sprintf('regression method is used to solve for the a0 and a1 coefficients which are then used'))
disp(sprintf('to determine the original constants of the growth model a and b, where y = (a*x)/(b+x).'))

   n=length(X);
   Z=zeros(1,n);
   for i=1:n
   Z(i)=1/Y(i);
   end
   w=zeros(1,n);
   for i=1:n
   w(i)=1/X(i);
   end
   wav=sum(w)/n;
   zav=sum(Z)/n;
   sum(Z);
   Swz=0;
   Sww=0;
   for i=1:n
   Swz=Swz +w(i)*Z(i)-wav*zav;
   Sww=Sww + (w(i))^2-wav^2;
   end

   Sww;
   Swz;

   a1=Swz/Sww
   a0=zav-a1*wav

   disp(sprintf('Now since a0 and a1 are found, the original constants of the model a and b'))
   disp(sprintf('are found as'))
   a=1/(a0)
   b=a1/a0

   disp(sprintf('\nThe model is described as\n              y=%5g',a))
   disp(sprintf('\b*x/(%5g',b))
   disp(sprintf('\b + x)                 (5)'))

   xp=(min(X):0.01:max(X));
   yp=zeros(1,length(xp));
   for i=1:length(xp)
   yp(i)=(a.*xp(i))/(b+xp(i));
   end
   plot(xp,yp)
   xlabel('x')
   ylabel('y=ax/(b+x)')
   hold on
   plot(X,Y,'bo','MarkerFaceColor','b')

   hold off
```

```
end
```