EGR 103L - Fall 2016

Laboratory 8 - Intermediate Curve Fitting

CEMAL YAGCIOGLU
Lab Section 5D, Wednesday 11.45AM - 2.35PM
30 OCTOBER, 2016

I have adhered to the Duke Community Standard in completing this assignment. I understand that a violation of the Standard can result in failure of this assignment, failure of this course, and/or suspension from Duke University.

Contents

List of Figures

1 Palm 6.10

St value is equal to 57550.

Order	Equation ($d = f(v)$)
1	$(5.6857e + 00) + (-8.5857e + 01)x$
2	$(5.0893e - 02) + (1.1054e + 00)x(2.3571e + 00)x^2$
3	$(1.3889e - 04) + (3.2143e - 02)x + (1.8790e + 00)x^2 + (-7.1429e + 00)x^3$

Order	S_r	r^2	$d(65)$ (ft)	$d(100)$ (ft)
1	9.7714e+02	9.8302e-01	2.8371e+02	4.8271e+02
2	1.0179e+01	9.9982e-01	2.8923e+02	6.2182e+02
3	8.9286e+00	9.9984e-01	2.8894e+02	6.4107e+02

Although r squared value increases with higher number of terms, 2nd degree polynomial and 3rd degree polynomial are both good representations of the given data. The increasing r squared value when passing from second degree to third degree is due to side effect of mathematics and does not add a scientifically significant term.

2 Palm 6.16

Model is : $y = 5.75ln(x) + 9.91$. Thus a1=5.75 and a2=9.91.
St, Sr and r2 values are respectively 160, 0.4145, 0.997.
Estimates for x=2.5 and x=11 are respectively 15.2, 23.7.

3 Chapra 15.12

Model is : $y = 3.7450e - 01 + (9.8644e - 01)x + (8.4564e - 01)x^{-1}$
St, Sr and r2 values are respectively 6.9480e+00, 2.7651e-03, 9.9960e-01.
Estimates for x=1.5 and x=4.5 are respectively 2.4179e+00, 5.0014e+00.

4 Chapra 14.12

Model is : $W = (4.1489e - 01) * A^{3.7991e-01}$
St, Sr and r2 values are respectively 7.0102e-03, 2.0240e-04, 9.7113e-01.
Estimate for W=95 is 2.3404e+00(meters squared).

5 Chapra 14.14

Model is : $k = ((1.0345e + 01) * c^2)/(2.0897e + 00 + c^2)$
St, St and r2 values are respectively 4.4527e-01, 1.9178e-04, 9.9957e-01.

6 Chapra 15.7

Model is : $OC = (-1.0493e-01)c + (-4.3704e-05)T^3 + (5.7444e-03)T^2 + (-3.3642e-01)T + 1.4027e+01$
St, Sr and r2 values are respectively 1.0400e+02, 1.2784e+00, 9.8771e-01.
Estimate for T=12 and c=15 is 9.1678e+00.
For this model, r2 value is high, indicating a good mathematical fit. Also relative error for given point is small, which tells that exptrapolation can be accurately used meaning its a good mathematical fit.
Relative error is equal to 8.5605e-03.

7 Chapra 15.10

Model is : $p(t) = (4.1375e+00)e^{-1.5t} + (2.8959e+00)e^{-0.3t} + (1.5349e+00)e^{-0.05t}$
St, Sr and r2 values are respectively 2.0409e+01, 8.0348e-02, 9.9606e-01.
In this model, r2 value is high indicating a mathetically fit model. Also, as expected for decay, all graphs show decrease. Thus its a mathematically good model.

8 Chapra 15.10 Alternate

Model is : $p(t) = (5.6328e+00)e^{-1.5t} + (-4.4272e+00)e^{-0.3t} + (7.8906e+00)e^{-0.2t}$
St, Sr and r2 values are respectively 2.0409e+01, 7.9611e-02, 9.9610e-01.
This model has a higher r2 value, however Decay B starts from a negative value which is not scientificially accurate as a population cannot be negative. Thus, it is not a scientifically accurate model.

9 Chapra 14.11 (Linearized)

St is equal to 5.4240e+04.

Fit	Equation	S_r	r^2	Tiger M. (watts)
Exp.	$M = (5.3509e+00)\ e^{((1.1138e-02)\ m)}$	4.2972e+04	2.0773e-01	4.9640e+01
Power	$M = (3.3893e+00)\ m^{(7.2656e-01)}$	1.1874e+02	9.9781e-01	1.5920e+02
Sat. G.	$M = (2.1011e+01)\frac{m}{(3.4857e+00)+m}$	6.6866e+04	-2.3279e-01	2.0651e+01

10 Chapra 14.11 (Nonlinear Regression)

St is equal to 5.4240e+04.

Fit	Equation	S_r	r^2	Tiger M. (watts)
Exp.	$M = (2.6088e+01)\ e^{((5.8543e-03)\ m)}$	3.7901e+03	9.3012e-01	8.4128e+01
Power	$M = (3.5436e+00)\ m^{(7.2345e-01)}$	6.2197e+01	9.9885e-01	1.6373e+02
Sat. G.	$M = (5.5198e+02)\frac{m}{(4.1748e+02)+m}$	2.7645e+01	9.9949e-01	1.7878e+02

A Codes and Output

A.1 ONE.m

```
% [ONE.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

clear
% V data
v = 20:10:70;
% d data
d = [45 80 130 185 250 330];
plot(v,d,'ko')
hold on
% Linear  a
x=linspace(20,70,100)
p1 = polyfit(v,d,1);
y1=polyval(p1,x)
y11=polyval(p1,v)
plot(x,y1,'b-')
est1=polyval(p1,[65 100])
hold on

% Quadratic  a
p2 = polyfit(v,d,2);
y2=polyval(p2,x)
y22=polyval(p2,v)
plot(x,y2,'r--')
est2=polyval(p2,[65 100])
hold on
% Cubic  a
p3 = polyfit(v,d,3);
y3=polyval(p3,x)
y33=polyval(p3,v)
plot(x,y3,'c-.')
legend('data points','linear','quadratic','cubic','location','northwest')
xlabel('v (mi/hr)')
ylabel('d (ft)')
title('Distance vs Velocity Plot(cy111)')
print -depsc DistanceVelocityPlot
est3=polyval(p3,[65 100])
St = sum(( d - mean(d) ).^2)

% Compute sum of the squares of the estimate residuals
Sr1 = sum(( d - y11 ).^2)
Sr2 = sum(( d - y22 ).^2)
Sr3 = sum(( d - y33 ).^2)

% Compute the coefficient of determination
r21 = (St - Sr1) / St
r22 = (St - Sr2) / St
r23 = (St - Sr3) / St
```

A.2 TWO.m

```
% [TWO.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: General Linear Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

clear
%% Rename and create model data
xx=1:1:10
yy=[10 14 16 18 19 20 21 22 23 23]
lnx=log(xx)
x=lnx(:)
y=yy(:)


%% Define model equation and A matrix

        yeqn = @(coefs, x) coefs(1)*x.^1 + coefs(2)*x.^0;
        A    = [x.^1 x.^0];


%% Determine the function coefficients
MyCoefs = A\y

%% Generate estimates and model
yhat   = yeqn(MyCoefs, x);
x1=log(2.5)
x2=log(11)
est = yeqn(MyCoefs, [x1  x2])

%% Calculate statistics
% Compute sum of the squares of the data residuals
St = sum(( y - mean(y) ).^2)

% Compute sum of the squares of the estimate residuals
Sr = sum(( y - yhat ).^2)

% Compute the coefficient of determination
r2 = (St - Sr) / St
```

A.3 THREE.m

```
% [THREE.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: General Linear Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

%% Initialize the workspace
clear; format short e


%% Load and manipulate the data

x=1:1:5
y=[2.2 2.8 3.6 4.5 5.5]
x=x'
y=y'



%% Define model equation and A matrix

        yeqn = @(coefs, x) coefs(1)*x.^0 + coefs(2)*x.^1 + coefs(3)*x.^-1;
        A    =                     [x.^0   x.^1    x.^-1];


%% Determine the function coefficients
MyCoefs = A\y

%% Generate estimates and model
yhat   = yeqn(MyCoefs, x);
x1=1.5
x2=4.5
est = yeqn(MyCoefs, [x1 x2])


%% Calculate statistics
% Compute sum of the squares of the data residuals
St = sum(( y - mean(y) ).^2)

% Compute sum of the squares of the estimate residuals
Sr = sum(( y - yhat ).^2)

% Compute the coefficient of determination
r2 = (St - Sr) / St
```

A.4 FOUR.m

```
% [FOUR.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: Linearized Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

%% Initialize the workspace
clear; format short e

%% Rename and create model data
W=[70 75 77 80 82 84 87 90]
A=[2.10 2.12 2.15 2.20 2.22 2.23 2.26 2.30]
lnW=log(W)
lnA=log(A)
x=lnW'
y=lnA'

%% Define model equation and A matrix

        yeqn = @(coefs, x) coefs(1)*x.^1 + coefs(2)*x.^0;
        A    =                     [x.^1            x.^0];

%% Determine the function coefficients
MyCoefs = A\y

%% Generate estimates and model
yhat   = yeqn(MyCoefs, x);

%% Calculate statistics
% Compute sum of the squares of the data residuals
St = sum(( y - mean(y) ).^2)


% Compute sum of the squares of the estimate residuals
Sr = sum(( y - yhat ).^2)

% Compute the coefficient of determination
r2 = (St - Sr) / St

finalb = MyCoefs(1)
finala = exp(MyCoefs(2))

est=exp(yeqn(MyCoefs, log(95)))
```

A.5 FIVE.m

```
% [FIVE.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: Linearized Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

%% Initialize the workspace
clear; format short e

%% Rename and create model data
c = [0.5 0.8 1.5 2.5 4]
k = [1.1 2.5 5.3 7.6 8.9]
Divk=1./k
Divc=1./(c.^2)
x=Divc'
y=Divk'
xmodel=linspace(min(x),max(x),100)

%% Define model equation and A matrix

        yeqn = @(coefs, x) coefs(1)*x.^1 + coefs(2)*x.^0;

        A    =                    [x.^1            x.^0];


%% Determine the function coefficients
MyCoefs = A\y

%% Generate estimates and model
yhat   = yeqn(MyCoefs, x);




%% Calculate statistics
% Compute sum of the squares of the data residuals
St = sum(( y - mean(y) ).^2)

% Compute sum of the squares of the estimate residuals
Sr = sum(( y - yhat ).^2)

% Compute the coefficient of determination
r2 = (St - Sr) / St

kmax = 1/MyCoefs(2)
cs = MyCoefs(1)*kmax

ymodel=(kmax.*xmodel.^2)./(cs+xmodel.^2)
yvalues=(kmax.*c.^2)./(cs+c.^2)
plot(c,      k,      'ko',...
     c,      yvalues,   'ks',...
```

```
        xmodel, ymodel, 'k-');
 xlabel('c(mg/L)')
 ylabel('k')
 title('Growth Rate of Bacteria(cy111)')
 legend('Data', 'Estimates', 'Model', 'location', 'northwest')
print -depsc BacteriaPlot
```

A.6 SIX.m

```
% [SIX.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: Linearized Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]
clear
load('DOCtable.mat')

%% Initialize the workspace
format short e
figure(1); clf

%%% Create data set - this takes the place of loading the set
%%% for this demonstration; otherwise, the end of this step
%%% should have x1m, x2m, and ym defined as matrices
%[x1m, x2m] = meshgrid(0:.1:1, 0:.15:3);
%T , c
%ym - OC

%% Rename and create vectors and model data
x1v = T(:);
x2v = c(:);
yv  = OC(:);
[x1model, x2model] = meshgrid(...
    linspace(min(x1v), max(x1v), 19),...
    linspace(min(x2v), max(x2v), 17));

%% Define model equation and A matrix

        yeqn = @(coefs, x1e, x2e) coefs(1)*x2e.^1 + coefs(2)*x1e.^3 + coefs(3)*x1e.^2 + coefs(4)*x1e
        A    =                            [x2v.^1            x1v.^3            x1v.^2  x1v.^1    x1v


%% Determine the function coefficients
MyCoefs = A\yv

%% Generate estimates and model
yhat   = yeqn(MyCoefs, x1v, x2v);
ymodel = yeqn(MyCoefs, x1model, x2model);

%% Calculate statistics
St = sum((yv - mean(yv)).^2)
Sr = sum((yv - yhat).^2)
r2 = (St - Sr) / St

%% Generate plots

%figure(1); clf
%surfc(T, c, OC);
%xlabel('x1'); ylabel('x2'); zlabel('y Data')
%view(145, 15)
```

```
% Model data
figure(1)
surfc(x1model, x2model, ymodel)
title('Oxygen Concentration in Water(cy111)')
xlabel('T values(C)'); ylabel('c values(g/L)'); zlabel('OC values(mg/L)')
view(145, 15)
colormap autumn
print -depsc OxygenConcentration
Test=12
cest=15
est = yeqn(MyCoefs, Test, cest)
relerr=(est-9.09)/9.09
```

A.7 SEVEN.m

```
% [SEVEN.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: General Multidimensional Linear Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]
%% Initialize the workspace
clear; format short e


t = [0.5 1 2 3 4 5 6 7 9]
p = [6 4.4 3.2 2.7 2 1.9 1.7 1.4 1.1]
x=t'
y=p'


%% Rename and create model data

xmodel = linspace(min(x), max(x), 100);

%% Define model equation and A matrix

        yeqn = @(coefs, x) coefs(1).*exp(-1.5.*x) + coefs(2).*exp(-0.3.*x) + coefs(3).*exp(-0.05.*x)
        A    =                     [exp(-1.5.*x)    exp(-0.3.*x)    exp(-0.05.*x)];

%% Determine the function coefficients
MyCoefs = A\y

%% Generate estimates and model
yhat   = yeqn(MyCoefs, x);
ymodel = yeqn(MyCoefs, xmodel);

%% Calculate statistics
% Compute sum of the squares of the data residuals
St = sum(( y - mean(y) ).^2)

% Compute sum of the squares of the estimate residuals
Sr = sum(( y - yhat ).^2)

% Compute the coefficient of determination
r2 = (St - Sr) / St

%% Generate plots
figure(1)
plot(x,      y,      'ko',...
     x,      yhat,   'ks',...
     xmodel, ymodel, 'k-');
 xlabel('Time(s)')
 ylabel('Population')
 title('Population vs. Time First Model(cy111)')
 legend('Data', 'Estimates', 'Model')
 print -depsc Decay11
```

```
figure(2)
tmodel=linspace(0,9,100)

plot(tmodel,MyCoefs(1).*exp(-1.5.*tmodel),'r--')
hold on
plot(tmodel,MyCoefs(2).*exp(-0.3.*tmodel),'b-')
hold on
plot(tmodel,MyCoefs(3).*exp(-0.05.*tmodel),'g-.')
 title('Decay of A,B,C First Model(cy111)')
legend('Decay A', 'Decay B', 'Decay C')
print -depsc Decay12
```

A.8 EIGHT.m

```
% [EIGHT.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: General Linear Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]
%% Initialize the workspace
clear; format short e


t = [0.5 1 2 3 4 5 6 7 9]
p = [6 4.4 3.2 2.7 2 1.9 1.7 1.4 1.1]
x=t'
y=p'


%% Rename and create model data

xmodel = linspace(min(x), max(x), 100);

%% Define model equation and A matrix

        yeqn = @(coefs, x) coefs(1).*exp(-1.5.*x) + coefs(2).*exp(-0.3.*x) + coefs(3).*exp(-0.2.*x);
        A    =                      [exp(-1.5.*x)    exp(-0.3.*x)    exp(-0.2.*x)];

%% Determine the function coefficients
MyCoefs = A\y

%% Generate estimates and model
yhat   = yeqn(MyCoefs, x);
ymodel = yeqn(MyCoefs, xmodel);

%% Calculate statistics
% Compute sum of the squares of the data residuals
St = sum(( y - mean(y) ).^2)

% Compute sum of the squares of the estimate residuals
Sr = sum(( y - yhat ).^2)

% Compute the coefficient of determination
r2 = (St - Sr) / St

%% Generate plots
figure(1)
plot(x,      y,      'ko',...
     x,      yhat,   'ks',...
     xmodel, ymodel, 'k-');
xlabel('Time(s)')
 ylabel('Population')
 title('Population vs. Time Second Model(cy111)')
 legend('Data', 'Estimates', 'Model')
 print -depsc Decay21
```

```
figure(2)
tmodel=linspace(0,9,100)
plot(tmodel,MyCoefs(1).*exp(-1.5.*tmodel),'r--')
hold on
plot(tmodel,MyCoefs(2).*exp(-0.3.*tmodel),'b-')
hold on
plot(tmodel,MyCoefs(3).*exp(-0.2.*tmodel),'g-.')
title('Decay of A,B,C Second Model(cy111)')
legend('Decay A', 'Decay B', 'Decay C')
print -depsc Decay22
```

A.9 NINE.m

```
% [NINE.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: Linearized Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]
%% Initialize the workspace
clear; format short e


%% Load and manipulate the data

Mass=[400 70 45 2 0.3 0.16]
Metabolism=[270 82 50 4.8 1.45 0.97]

%% Rename and create model data
x = Mass'
y = Metabolism'
xmodel = linspace(min(x), max(x), 100);

%% Define the model equation; transform the variables; find the linearized fit; transform back

%model = 'exponential'
%switch model
%    case 'exponential'
        yeqnE = @(coefs, x) coefs(1).*exp(coefs(2).*x);
        xiE  = x;
        etaE = log(y);
        PE = polyfit(xiE, etaE, 1);
        MyCoefsE(1) = exp(PE(2));
        MyCoefsE(2) = PE(1)
%    case 'power law'
        yeqnP = @(coefs, x) coefs(1).*x.^coefs(2);
        xiP  = log10(x);
        etaP = log10(y);
        PP = polyfit(xiP, etaP, 1);
        MyCoefsP(1) = 10^(PP(2));
        MyCoefsP(2) = PP(1)
 %   case 'sat growth'
        yeqnS = @(coefs, x) coefs(1).*x./(coefs(2)+x);
        xiS  = 1./x;
        etaS = 1./y;
        PS = polyfit(xiS, etaS, 1);
        MyCoefsS(1) =    1/PS(2);
        MyCoefsS(2) = PS(1)/PS(2)


%% Generate estimates and model

yhatE   = yeqnE(MyCoefsE, x);
ymodelE = yeqnE(MyCoefsE, xmodel);

```

```
yhatP   = yeqnP(MyCoefsP, x);
ymodelP = yeqnP(MyCoefsP, xmodel);

yhatS   = yeqnS(MyCoefsS, x);
ymodelS = yeqnS(MyCoefsS, xmodel);

%% Calculate statistics
% Compute sum of the squares of the data residuals
St = sum(( y - mean(y) ).^2)

% Compute sum of the squares of the estimate residuals
SrE = sum(( y - yhatE ).^2)
SrP = sum(( y - yhatP ).^2)
SrS = sum(( y - yhatS ).^2)

% Compute the coefficient of determination
r2E = (St - SrE) / St
r2P = (St - SrP) / St
r2S = (St - SrS) / St

%% Generate plots
figure(1)
plot(x,      y,      'ko',xmodel, ymodelE, 'k-')
title('Metabolism as a Function of Mass(Exponential) (cy111)')'
 xlabel('Mass(kg)')
 ylabel('Metabolism(watt)')
 print -depsc M11


figure(2)
plot(xmodel,log(ymodelE), x,      log(y),      'ko' )
 xlabel('Mass(kg)')
 ylabel('log(Metabolism)')
 title('Linearized Exponential Model (cy111)')
 print -depsc M12

figure(3)
plot(x,      y,      'k-',xmodel, ymodelP, 'k:')
 xlabel('Mass(kg)')
 ylabel('Metabolism(watt)')
 title('Metabolism as a Function of Mass(Power) (cy111)')
 print -depsc M13

figure(4)
plot(log10(xmodel),log10(ymodelP), log10(x),      log10(y),      'ko' )
 xlabel('log10(Mass)')
 ylabel('log10(Metabolism)')
 title('Linearized Power Model (cy111)')
 print -depsc M14

figure(5)
plot(x,      y,      'ko',xmodel, ymodelS, 'k--')
 xlabel('Mass(kg)')
 ylabel('Metabolism(watt)')
 title('Metabolism as a Function of Mass(Saturation) (cy111)')
```

```
 print -depsc M15

figure(6)
plot(1./xmodel,1./ymodelS, 1./x,      1./y,      'ko' )
 xlabel('1/Mass(1/kg)')
 ylabel('1/Metabolism(1/watt)')
 title('Linearized Saturation Model (cy111)')
print -depsc M16

EstE = yeqnE(MyCoefsE, 200)
EstP = yeqnP(MyCoefsP, 200)
EstS = yeqnS(MyCoefsS, 200)
```

A.10 TEN.m

```
% [TEN.m]
% [Cemal Yagcioglu]
% [November 20,2016]
% Based on: Nonlinear Regression Example Code
% Written by: Michael R. Gustafson II
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]
%% Initialize the workspace
clear; format short e
figure(1); clf

%% Load and manipulate the data
Mass=[400 70 45 2 0.3 0.16]
Metabolism=[270 82 50 4.8 1.45 0.97]

%% Rename and create model data
x = Mass'
y = Metabolism'
xmodel = linspace(min(x), max(x), 100);

        yeqnE = @(coefs, x) coefs(1).*exp(coefs(2).*x);
        InitGuessE = [5.3509 0.01138]

        yeqnP = @(coefs, x) coefs(1).*x.^coefs(2);
        InitGuessP = [3.3893 0.72656]

        yeqnS = @(coefs, x) coefs(1).*x./(coefs(2)+x);
        InitGuessS = [21.011 3.4857]

%% Determine the function coefficients
fSSRE = @(coefs, x, y) sum(( y - yeqnE(coefs, x) ).^2)
[MyCoefsE, SrE] = fminsearch(@(MyCoefsDummy) fSSRE(MyCoefsDummy, x, y), InitGuessE)

fSSRP = @(coefs, x, y) sum(( y - yeqnP(coefs, x) ).^2)
[MyCoefsP, SrP] = fminsearch(@(MyCoefsDummy) fSSRP(MyCoefsDummy, x, y), InitGuessP)

fSSRS = @(coefs, x, y) sum(( y - yeqnS(coefs, x) ).^2)
[MyCoefsS, SrS] = fminsearch(@(MyCoefsDummy) fSSRS(MyCoefsDummy, x, y), InitGuessS)

%% Generate estimates and model
yhatE   = yeqnE(MyCoefsE, x);
ymodelE = yeqnE(MyCoefsE, xmodel);

yhatP   = yeqnP(MyCoefsP, x);
ymodelP = yeqnP(MyCoefsP, xmodel);

yhatS   = yeqnS(MyCoefsS, x);
ymodelS = yeqnS(MyCoefsS, xmodel);


%% Calculate statistics
% Compute sum of the squares of the data residuals
St = sum(( y - mean(y) ).^2)
```

```
% Compute sum of the squares of the estimate residuals
SrE = sum(( y - yhatE ).^2)
SrP = sum(( y - yhatP ).^2)
SrS = sum(( y - yhatS ).^2)

% Compute the coefficient of determination
r2E = (St - SrE) / St
r2P = (St - SrP) / St
r2S = (St - SrS) / St


%% Generate plots
figure(1)
plot(x,      y,      'ko',xmodel, ymodelE, 'k-')
title('Nonlinear Metabolism as a Function of Mass(Exponential) (cy111)')
 xlabel('Mass(kg)')
 ylabel('Metabolism(watt)')
 print -depsc M21


figure(2)
plot(xmodel,log(ymodelE), x,      log(y),      'ko' )
 xlabel('Mass(kg)')
 ylabel('log(Metabolism)')
 title('Nonlinear Transformation Linearized Exponential Model (cy111)')
 print -depsc M22

figure(3)
plot(x,      y,      'k-',xmodel, ymodelP, 'k:')
 xlabel('Mass(kg)')
 ylabel('Metabolism(watt)')
 title('Nonlinear Metabolism as a Function of Mass(Power) (cy111)')
 print -depsc M23

figure(4)
plot(log10(xmodel),log10(ymodelP), log10(x),      log10(y),      'ko' )
 xlabel('log10(Mass)')
 ylabel('log10(Metabolism)')
 title('Nonlinear Transformation Power Model (cy111)')
print -depsc M24
figure(5)
plot(x,      y,      'ko',xmodel, ymodelS, 'k--')
 xlabel('Mass(kg)')
 ylabel('Metabolism(watt)')
 title('Nonlinear Metabolism as a Function of Mass(Saturation) (cy111)')
print -depsc M25
figure(6)
plot(1./xmodel,1./ymodelS, 1./x,      1./y,      'ko' )
 xlabel('1/Mass(1/kg)')
 ylabel('1/Metabolism(1/watt)')
 title('Nonlinear Transformation Saturation Model (cy111)')
 print -depsc M26
```

```
EstE = yeqnE(MyCoefsE, 200)
EstP = yeqnP(MyCoefsP, 200)
EstS = yeqnS(MyCoefsS, 200)
```

B Figures

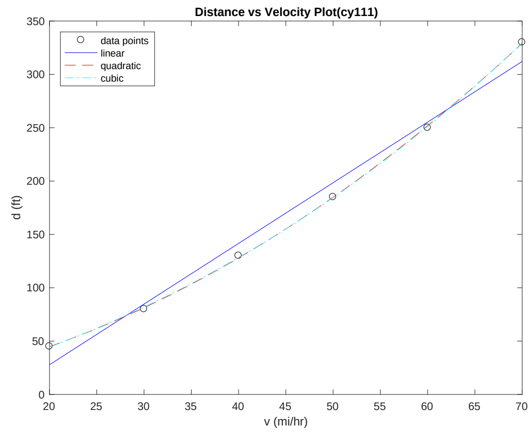

Figure 1: Distance Velocity Plot

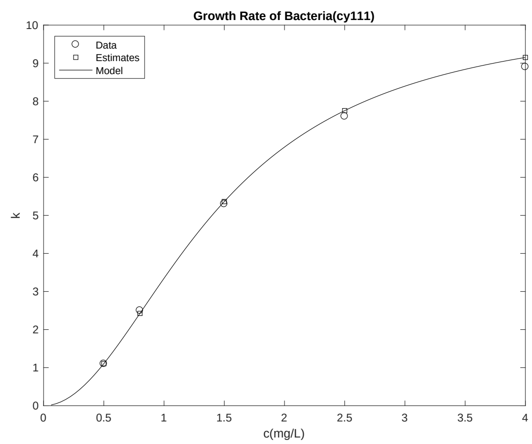

Figure 2: Bacteria Plot

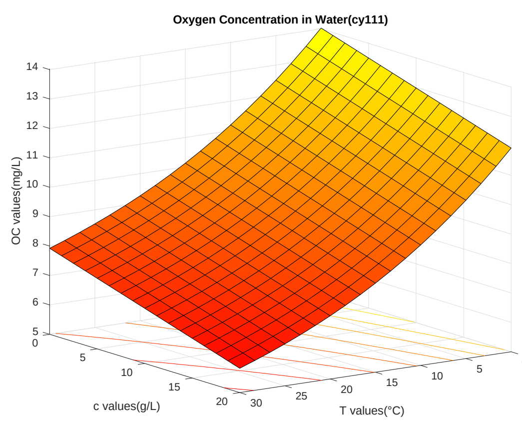

Figure 3: Oxygen Concentration Surface

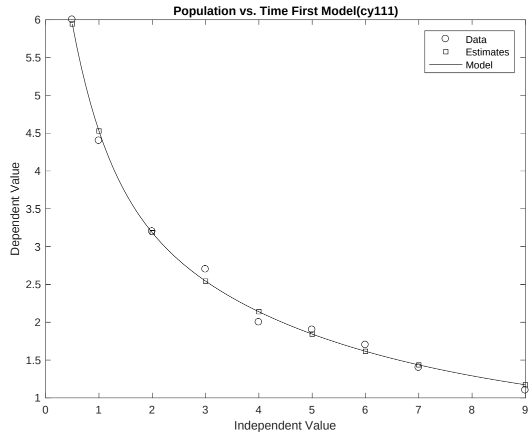

Figure 4: Population vs Time First Model

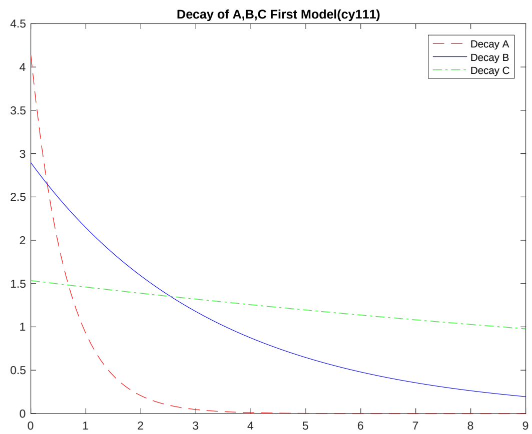

Figure 5: Decay of A,B,C First Model

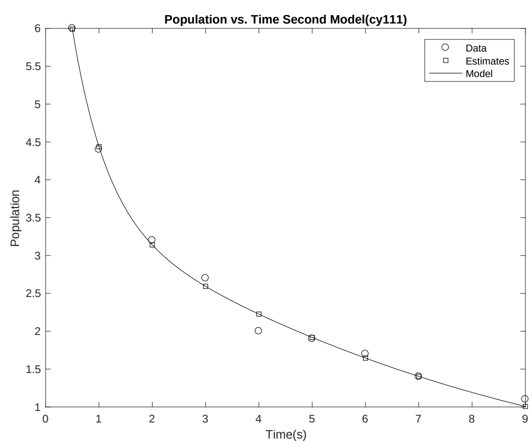

Figure 6: Population vs Time Second Model

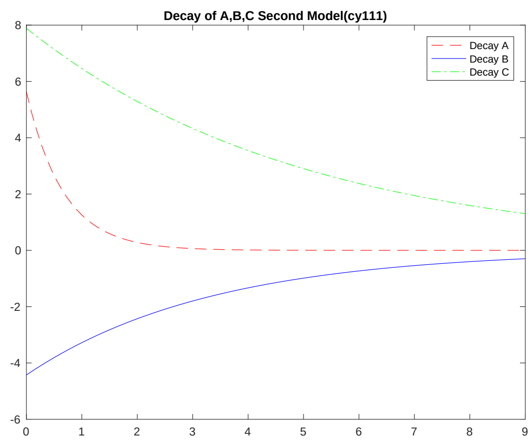

Figure 7: Decay of A,B,C Second Model

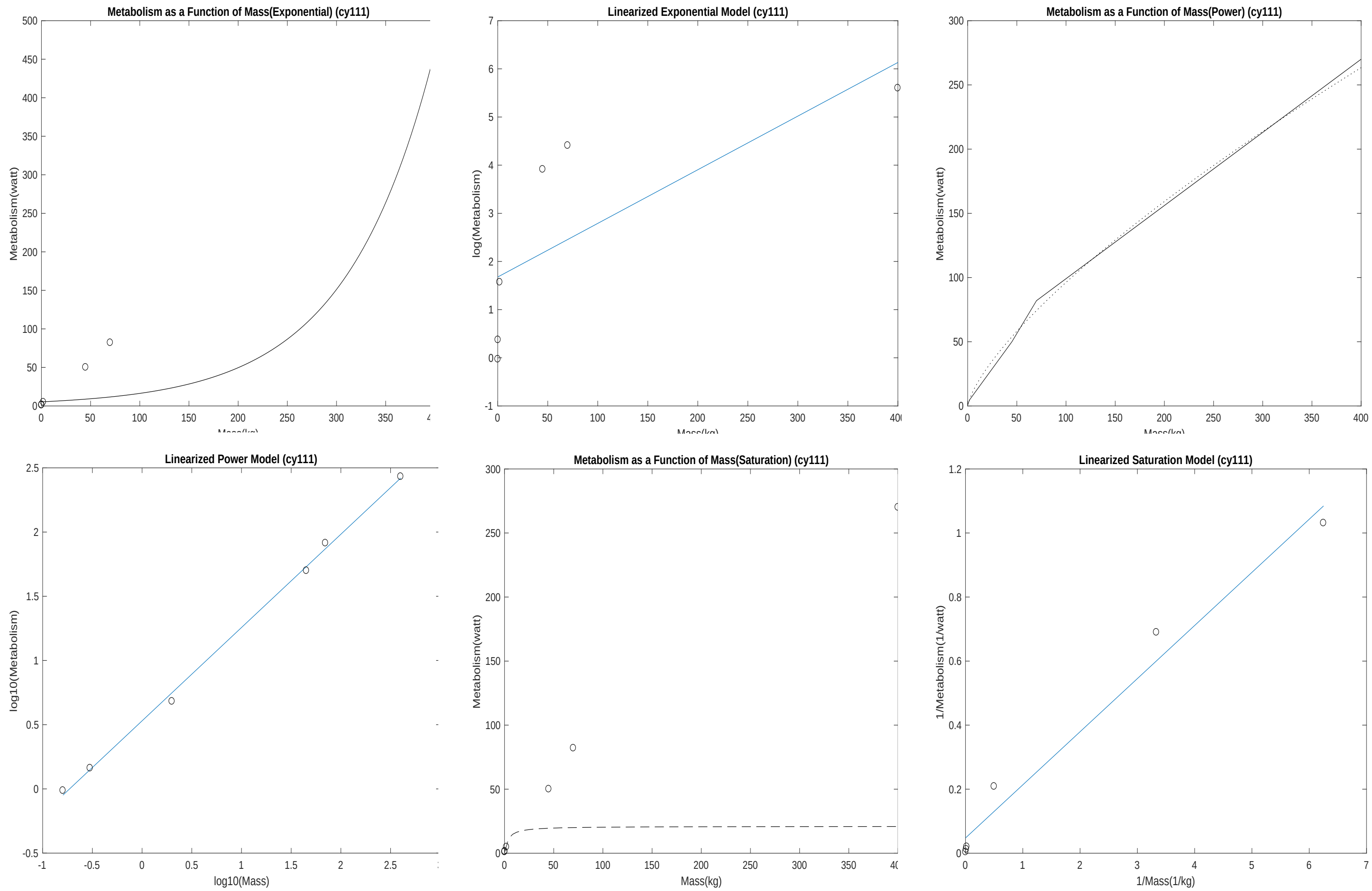

Figure 8: Untransformed and Transformed Plots for Linearized Metabolism Models

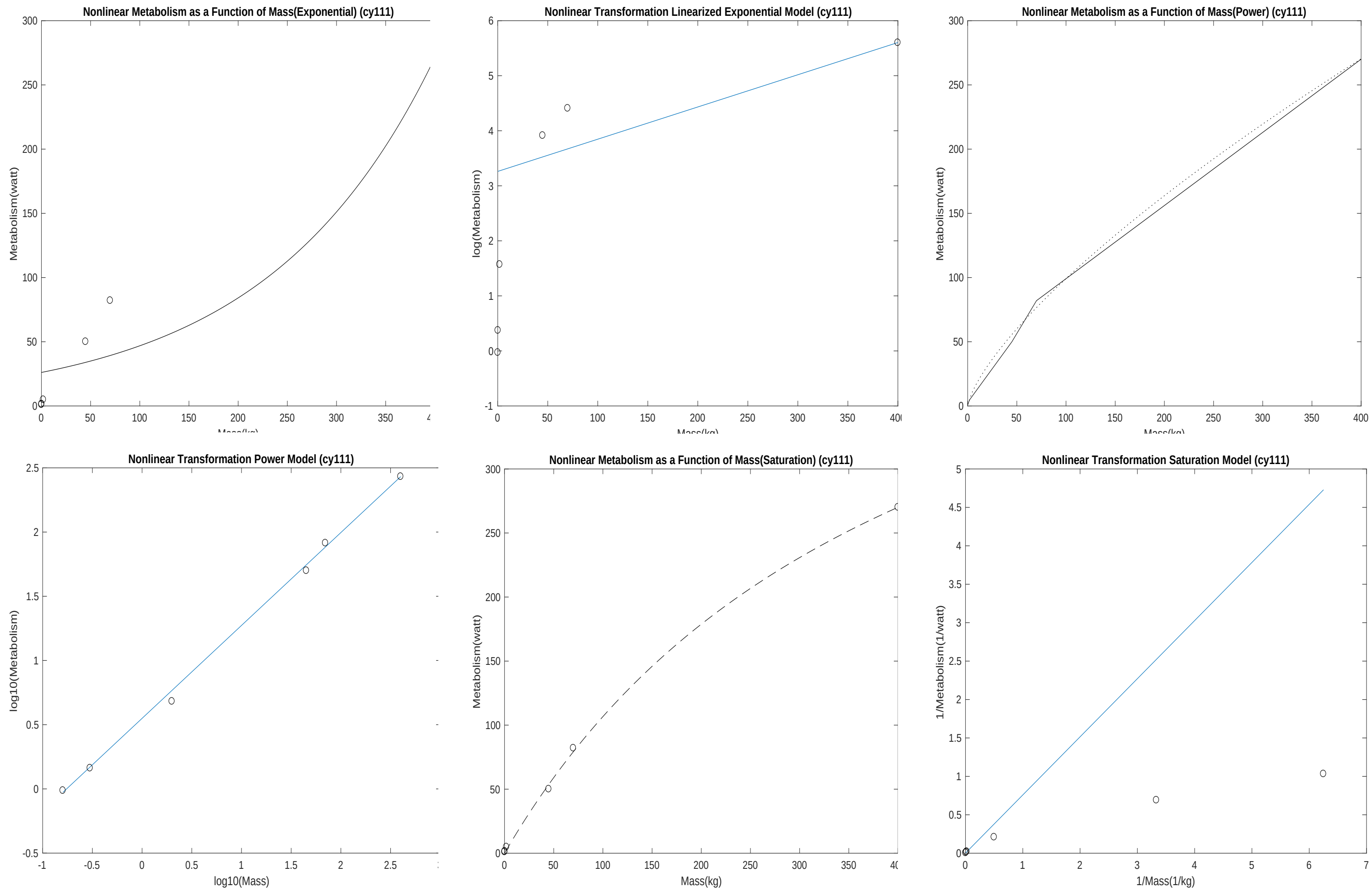

Figure 9: Untransformed and Transformed Plots for Unlinearized Metabolism Models