

# 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

<b>1</b>	<b>Palm 6.10</b>	<b>3</b>
<b>2</b>	<b>Palm 6.16</b>	<b>3</b>
<b>3</b>	<b>Chapra 15.12</b>	<b>3</b>
<b>4</b>	<b>Chapra 14.12</b>	<b>3</b>
<b>5</b>	<b>Chapra 14.14</b>	<b>3</b>
<b>6</b>	<b>Chapra 15.7</b>	<b>4</b>
<b>7</b>	<b>Chapra 15.10</b>	<b>4</b>
<b>8</b>	<b>Chapra 15.10 Alternate</b>	<b>4</b>
<b>9</b>	<b>Chapra 14.11 (Linearized)</b>	<b>4</b>
<b>10</b>	<b>Chapra 14.11 (Nonlinear Regression)</b>	<b>4</b>
<b>A</b>	<b>Codes and Output</b>	<b>5</b>
A.1	ONE.m . . . . .	5
A.2	TWO.m . . . . .	6
A.3	THREE.m . . . . .	7
A.4	FOUR.m . . . . .	8
A.5	FIVE.m . . . . .	9
A.6	SIX.m . . . . .	11
A.7	SEVEN.m . . . . .	13
A.8	EIGHT.m . . . . .	15
A.9	NINE.m . . . . .	17
A.10	TEN.m . . . . .	20
<b>B</b>	<b>Figures</b>	<b>23</b>

## List of Figures

1	Distance Velocity Plot . . . . .	23
2	Bacteria Plot . . . . .	23
3	Oxygen Concentration Surface . . . . .	23
4	Population vs Time First Model . . . . .	24

5	Decay of A,B,C First Model . . . . .	24
6	Population vs Time Second Model . . . . .	24
7	Decay of A,B,C Second Model . . . . .	25
8	Untransformed and Transformed Plots for Linearized Metabolism Models . . . . .	26
9	Untransformed and Transformed Plots for Unlinearized Metabolism Models . . . . .	27

## 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 :  $y = 5.75 \ln(x) + 9.91$ . 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 extrapolation 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 scientifically 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

```
1  % [ONE.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % I have adhered to all the tenets of the
5  % Duke Community Standard in creating this code.
6  % Signed: [cy111]
7
8  clear
9  % V data
10 v = 20:10:70;
11 % d data
12 d = [45 80 130 185 250 330];
13 plot(v,d,'ko')
14 hold on
15 % Linear a
16 x=linspace(20,70,100)
17 p1 = polyfit(v,d,1);
18 y1=polyval(p1,x)
19 y11=polyval(p1,v)
20 plot(x,y1,'b-')
21 est1=polyval(p1,[65 100])
22 hold on
23
24 % Quadratic a
25 p2 = polyfit(v,d,2);
26 y2=polyval(p2,x)
27 y22=polyval(p2,v)
28 plot(x,y2,'r--')
29 est2=polyval(p2,[65 100])
30 hold on
31 % Cubic a
32 p3 = polyfit(v,d,3);
33 y3=polyval(p3,x)
34 y33=polyval(p3,v)
35 plot(x,y3,'c-.')
36 legend('data points','linear','quadratic','cubic','location','northwest')
37 xlabel('v (mi/hr)')
38 ylabel('d (ft)')
39 title('Distance vs Velocity Plot(cy111)')
40 print -depsc DistanceVelocityPlot
41 est3=polyval(p3,[65 100])
42 St = sum(( d - mean(d) ).^2)
43
44 % Compute sum of the squares of the estimate residuals
45 Sr1 = sum(( d - y11 ).^2)
46 Sr2 = sum(( d - y22 ).^2)
47 Sr3 = sum(( d - y33 ).^2)
48
49 % Compute the coefficient of determination
50 r21 = (St - Sr1) / St
51 r22 = (St - Sr2) / St
52 r23 = (St - Sr3) / St
```

## A.2 TWO.m

```
1  % [TWO.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: General Linear Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9
10 clear
11 %% Rename and create model data
12 xx=1:1:10
13 yy=[10 14 16 18 19 20 21 22 23 23]
14 lnx=log(xx)
15 x=lnx(:)
16 y=yy(:)
17
18
19 %% Define model equation and A matrix
20
21     yeqn = @(coefs, x) coefs(1)*x.^1 + coefs(2)*x.^0;
22     A     = [x.^1 x.^0];
23
24
25 %% Determine the function coefficients
26 MyCoefs = A\y
27
28 %% Generate estimates and model
29 yhat     = yeqn(MyCoefs, x);
30 x1=log(2.5)
31 x2=log(11)
32 est = yeqn(MyCoefs, [x1 x2])
33
34 %% Calculate statistics
35 % Compute sum of the squares of the data residuals
36 St = sum(( y - mean(y) ).^2)
37
38 % Compute sum of the squares of the estimate residuals
39 Sr = sum(( y - yhat ).^2)
40
41 % Compute the coefficient of determination
42 r2 = (St - Sr) / St
```

### A.3 THREE.m

```
1  % [THREE.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: General Linear Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9
10 %% Initialize the workspace
11 clear; format short e
12
13
14 %% Load and manipulate the data
15
16 x=1:1:5
17 y=[2.2 2.8 3.6 4.5 5.5]
18 x=x'
19 y=y'
20
21
22
23 %% Define model equation and A matrix
24
25     yeqn = @(coefs, x) coefs(1)*x.^0 + coefs(2)*x.^1 + coefs(3)*x.^-1;
26     A     =                [x.^0   x.^1   x.^-1];
27
28
29 %% Determine the function coefficients
30 MyCoefs = A\y
31
32 %% Generate estimates and model
33 yhat     = yeqn(MyCoefs, x);
34 x1=1.5
35 x2=4.5
36 est = yeqn(MyCoefs, [x1 x2])
37
38
39 %% Calculate statistics
40 % Compute sum of the squares of the data residuals
41 St = sum(( y - mean(y) ).^2)
42
43 % Compute sum of the squares of the estimate residuals
44 Sr = sum(( y - yhat ).^2)
45
46 % Compute the coefficient of determination
47 r2 = (St - Sr) / St
```

## A.4 FOUR.m

```
1  % [FOUR.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: Linearized Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9
10 %% Initialize the workspace
11 clear; format short e
12
13 %% Rename and create model data
14 W=[70 75 77 80 82 84 87 90]
15 A=[2.10 2.12 2.15 2.20 2.22 2.23 2.26 2.30]
16 lnW=log(W)
17 lnA=log(A)
18 x=lnW'
19 y=lnA'
20
21 %% Define model equation and A matrix
22
23      yeqn = @(coefs, x) coefs(1)*x.^1 + coefs(2)*x.^0;
24      A      =              [x.^1          x.^0];
25
26 %% Determine the function coefficients
27 MyCoefs = A\y
28
29 %% Generate estimates and model
30 yhat    = yeqn(MyCoefs, x);
31
32 %% Calculate statistics
33 % Compute sum of the squares of the data residuals
34 St = sum(( y - mean(y) ).^2)
35
36
37 % Compute sum of the squares of the estimate residuals
38 Sr = sum(( y - yhat ).^2)
39
40 % Compute the coefficient of determination
41 r2 = (St - Sr) / St
42
43 finalb = MyCoefs(1)
44 finala = exp(MyCoefs(2))
45
46 est=exp(yeqn(MyCoefs, log(95)))
```



## A.5 FIVE.m

```
1  % [FIVE.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: Linearized Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9
10 %% Initialize the workspace
11 clear; format short e
12
13 %% Rename and create model data
14 c = [0.5 0.8 1.5 2.5 4]
15 k = [1.1 2.5 5.3 7.6 8.9]
16 Divk=1./k
17 Divc=1./(c.^2)
18 x=Divc'
19 y=Divk'
20 xmodel=linspace(min(x),max(x),100)
21
22 %% Define model equation and A matrix
23
24         yeqn = @(coefs, x) coefs(1)*x.^1 + coefs(2)*x.^0;
25
26         A      =                [x.^1          x.^0];
27
28
29 %% Determine the function coefficients
30 MyCoefs = A\y
31
32 %% Generate estimates and model
33 yhat    = yeqn(MyCoefs, x);
34
35
36
37
38 %% Calculate statistics
39 % Compute sum of the squares of the data residuals
40 St = sum(( y - mean(y) ).^2)
41
42 % Compute sum of the squares of the estimate residuals
43 Sr = sum(( y - yhat ).^2)
44
45 % Compute the coefficient of determination
46 r2 = (St - Sr) / St
47
48 kmax = 1/MyCoefs(2)
49 cs = MyCoefs(1)*kmax
50
51 ymodel=(kmax.*xmodel.^2)./(cs+xmodel.^2)
52 yvalues=(kmax.*c.^2)./(cs+c.^2)
53 plot(c,      k,      'ko',...
54      c,      yvalues,      'ks',...
```

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

## A.6 SIX.m

```
1  % [SIX.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: Linearized Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9  clear
10 load('DOCTable.mat')
11
12 %% Initialize the workspace
13 format short e
14 figure(1); clf
15
16 %% Create data set - this takes the place of loading the set
17 %% for this demonstration; otherwise, the end of this step
18 %% should have x1m, x2m, and ym defined as matrices
19 % [x1m, x2m] = meshgrid(0:.1:1, 0:.15:3);
20 %T , c
21 %ym - OC
22
23 %% Rename and create vectors and model data
24 x1v = T(:);
25 x2v = c(:);
26 yv = OC(:);
27 [x1model, x2model] = meshgrid(...
28     linspace(min(x1v), max(x1v), 19),...
29     linspace(min(x2v), max(x2v), 17));
30
31 %% Define model equation and A matrix
32
33     yeqn = @(coefs, x1e, x2e) coefs(1)*x2e.^1 + coefs(2)*x1e.^3 + coefs(3)*x1e.^2 + coefs(4)*x1e
34     A      =                                [x2v.^1                x1v.^3                x1v.^2    x1v.^1    x1v
35
36
37 %% Determine the function coefficients
38 MyCoefs = A\yv
39
40 %% Generate estimates and model
41 yhat = yeqn(MyCoefs, x1v, x2v);
42 ymodel = yeqn(MyCoefs, x1model, x2model);
43
44 %% Calculate statistics
45 St = sum((yv - mean(yv)).^2)
46 Sr = sum((yv - yhat).^2)
47 r2 = (St - Sr) / St
48
49 %% Generate plots
50
51 %figure(1); clf
52 %surf(T, c, OC);
53 %xlabel('x1'); ylabel('x2'); zlabel('y Data')
54 %view(145, 15)
```

```

55 % Model data
56 figure(1)
57 surf(x1model, x2model, ymodel)
58 title('Oxygen Concentration in Water(cyl11)')
59 xlabel('T values(C)'); ylabel('c values(g/L)'); zlabel('OC values(mg/L)')
60 view(145, 15)
61 colormap autumn
62 print -depsc OxygenConcentration
63 Test=12
64 cest=15
65 est = yeqn(MyCoefs, Test, cest)
66 relerr=(est-9.09)/9.09

```

## A.7 SEVEN.m

```
1  % [SEVEN.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: General Multidimensional Linear Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9  %% Initialize the workspace
10 clear; format short e
11
12
13 t = [0.5 1 2 3 4 5 6 7 9]
14 p = [6 4.4 3.2 2.7 2 1.9 1.7 1.4 1.1]
15 x=t'
16 y=p'
17
18
19 %% Rename and create model data
20
21 xmodel = linspace(min(x), max(x), 100);
22
23 %% Define model equation and A matrix
24
25     yeqn = @(coefs, x) coefs(1).*exp(-1.5.*x) + coefs(2).*exp(-0.3.*x) + coefs(3).*exp(-0.05.*x)
26     A     =                [exp(-1.5.*x)    exp(-0.3.*x)    exp(-0.05.*x)];
27
28 %% Determine the function coefficients
29 MyCoefs = A\y
30
31 %% Generate estimates and model
32 yhat  = yeqn(MyCoefs, x);
33 ymodel = yeqn(MyCoefs, xmodel);
34
35 %% Calculate statistics
36 % Compute sum of the squares of the data residuals
37 St = sum(( y - mean(y) ).^2)
38
39 % Compute sum of the squares of the estimate residuals
40 Sr = sum(( y - yhat ).^2)
41
42 % Compute the coefficient of determination
43 r2 = (St - Sr) / St
44
45 %% Generate plots
46 figure(1)
47 plot(x,      y,      'ko',...
48      x,      yhat,   'ks',...
49      xmodel, ymodel, 'k-');
50 xlabel('Time(s)')
51 ylabel('Population')
52 title('Population vs. Time First Model(cy111)')
53 legend('Data', 'Estimates', 'Model')
54 print -depsc Decay11
```

```

55
56 figure(2)
57 tmodel=linspace(0,9,100)
58
59 plot(tmodel,MyCoefs(1).*exp(-1.5.*tmodel),'r--')
60 hold on
61 plot(tmodel,MyCoefs(2).*exp(-0.3.*tmodel),'b-')
62 hold on
63 plot(tmodel,MyCoefs(3).*exp(-0.05.*tmodel),'g-.'')
64 title('Decay of A,B,C First Model(cy111)')
65 legend('Decay A', 'Decay B', 'Decay C')
66 print -depsc Decay12

```

## A.8 EIGHT.m

```
1  % [EIGHT.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: General Linear Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9  %% Initialize the workspace
10 clear; format short e
11
12
13 t = [0.5 1 2 3 4 5 6 7 9]
14 p = [6 4.4 3.2 2.7 2 1.9 1.7 1.4 1.1]
15 x=t'
16 y=p'
17
18
19 %% Rename and create model data
20
21 xmodel = linspace(min(x), max(x), 100);
22
23 %% Define model equation and A matrix
24
25         yeqn = @(coefs, x) coefs(1).*exp(-1.5.*x) + coefs(2).*exp(-0.3.*x) + coefs(3).*exp(-0.2.*x);
26         A     =                [exp(-1.5.*x)    exp(-0.3.*x)    exp(-0.2.*x)];
27
28 %% Determine the function coefficients
29 MyCoefs = A\y
30
31 %% Generate estimates and model
32 yhat  = yeqn(MyCoefs, x);
33 ymodel = yeqn(MyCoefs, xmodel);
34
35 %% Calculate statistics
36 % Compute sum of the squares of the data residuals
37 St = sum(( y - mean(y) ).^2)
38
39 % Compute sum of the squares of the estimate residuals
40 Sr = sum(( y - yhat ).^2)
41
42 % Compute the coefficient of determination
43 r2 = (St - Sr) / St
44
45 %% Generate plots
46 figure(1)
47 plot(x,      y,      'ko',...
48      x,      yhat,   'ks',...
49      xmodel, ymodel, 'k-');
50 xlabel('Time(s)')
51 ylabel('Population')
52 title('Population vs. Time Second Model(cy111)')
53 legend('Data', 'Estimates', 'Model')
54 print -depsc Decay21
```

```

55
56 figure(2)
57 tmodel=linspace(0,9,100)
58 plot(tmodel,MyCoefs(1).*exp(-1.5.*tmodel),'r--')
59 hold on
60 plot(tmodel,MyCoefs(2).*exp(-0.3.*tmodel),'b-')
61 hold on
62 plot(tmodel,MyCoefs(3).*exp(-0.2.*tmodel),'g-.'')
63 title('Decay of A,B,C Second Model(cy111)')
64 legend('Decay A', 'Decay B', 'Decay C')
65 print -depsc Decay22

```



## A.9 NINE.m

```
1  % [NINE.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: Linearized Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9  %% Initialize the workspace
10 clear; format short e
11
12
13 %% Load and manipulate the data
14
15 Mass=[400 70 45 2 0.3 0.16]
16 Metabolism=[270 82 50 4.8 1.45 0.97]
17
18 %% Rename and create model data
19 x = Mass'
20 y = Metabolism'
21 xmodel = linspace(min(x), max(x), 100);
22
23 %% Define the model equation; transform the variables; find the linearized fit; transform back
24
25 %model = 'exponential'
26 %switch model
27 %    case 'exponential'
28 %        yeqnE = @(coefs, x) coefs(1).*exp(coefs(2).*x);
29 %        xiE = x;
30 %        etaE = log(y);
31 %        PE = polyfit(xiE, etaE, 1);
32 %        MyCoefsE(1) = exp(PE(2));
33 %        MyCoefsE(2) = PE(1)
34 %    case 'power law'
35 %        yeqnP = @(coefs, x) coefs(1).*x.^coefs(2);
36 %        xiP = log10(x);
37 %        etaP = log10(y);
38 %        PP = polyfit(xiP, etaP, 1);
39 %        MyCoefsP(1) = 10^(PP(2));
40 %        MyCoefsP(2) = PP(1)
41 %    case 'sat growth'
42 %        yeqnS = @(coefs, x) coefs(1).*x./(coefs(2)+x);
43 %        xiS = 1./x;
44 %        etaS = 1./y;
45 %        PS = polyfit(xiS, etaS, 1);
46 %        MyCoefsS(1) = 1/PS(2);
47 %        MyCoefsS(2) = PS(1)/PS(2)
48
49
50 %% Generate estimates and model
51
52 yhatE = yeqnE(MyCoefsE, x);
53 ymodelE = yeqnE(MyCoefsE, xmodel);
54
```

```

55 yhatP = yeqnP(MyCoefsP, x);
56 ymodelP = yeqnP(MyCoefsP, xmodel);
57
58 yhatS = yeqnS(MyCoefsS, x);
59 ymodelS = yeqnS(MyCoefsS, xmodel);
60
61 %% Calculate statistics
62 % Compute sum of the squares of the data residuals
63 St = sum(( y - mean(y) ).^2)
64
65 % Compute sum of the squares of the estimate residuals
66 SrE = sum(( y - yhatE ).^2)
67 SrP = sum(( y - yhatP ).^2)
68 SrS = sum(( y - yhatS ).^2)
69
70 % Compute the coefficient of determination
71 r2E = (St - SrE) / St
72 r2P = (St - SrP) / St
73 r2S = (St - SrS) / St
74
75 %% Generate plots
76 figure(1)
77 plot(x, y, 'ko', xmodel, ymodelE, 'k-')
78 title('Metabolism as a Function of Mass(Exponential) (cy111)')
79 xlabel('Mass(kg)')
80 ylabel('Metabolism(watt)')
81 print -depsc M11
82
83
84 figure(2)
85 plot(xmodel, log(ymodelE), x, log(y), 'ko' )
86 xlabel('Mass(kg)')
87 ylabel('log(Metabolism)')
88 title('Linearized Exponential Model (cy111)')
89 print -depsc M12
90
91 figure(3)
92 plot(x, y, 'k-', xmodel, ymodelP, 'k:')
93 xlabel('Mass(kg)')
94 ylabel('Metabolism(watt)')
95 title('Metabolism as a Function of Mass(Power) (cy111)')
96 print -depsc M13
97
98 figure(4)
99 plot(log10(xmodel), log10(ymodelP), log10(x), log10(y), 'ko' )
100 xlabel('log10(Mass)')
101 ylabel('log10(Metabolism)')
102 title('Linearized Power Model (cy111)')
103 print -depsc M14
104
105 figure(5)
106 plot(x, y, 'ko', xmodel, ymodelS, 'k--')
107 xlabel('Mass(kg)')
108 ylabel('Metabolism(watt)')
109 title('Metabolism as a Function of Mass(Saturation) (cy111)')

```

```

110     print -depsc M15
111
112     figure(6)
113     plot(1./xmodel,1./ymodelS, 1./x,      1./y,      'ko' )
114     xlabel('1/Mass(1/kg)')
115     ylabel('1/Metabolism(1/watt)')
116     title('Linearized Saturation Model (cy111)')
117     print -depsc M16
118
119     EstE = yeqnE(MyCoefsE, 200)
120     EstP = yeqnP(MyCoefsP, 200)
121     EstS = yeqnS(MyCoefsS, 200)

```

## A.10 TEN.m

```
1  % [TEN.m]
2  % [Cemal Yagcioglu]
3  % [November 20,2016]
4  % Based on: Nonlinear Regression Example Code
5  % Written by: Michael R. Gustafson II
6  % I have adhered to all the tenets of the
7  % Duke Community Standard in creating this code.
8  % Signed: [cy111]
9  %% Initialize the workspace
10 clear; format short e
11 figure(1); clf
12
13 %% Load and manipulate the data
14 Mass=[400 70 45 2 0.3 0.16]
15 Metabolism=[270 82 50 4.8 1.45 0.97]
16
17 %% Rename and create model data
18 x = Mass'
19 y = Metabolism'
20 xmodel = linspace(min(x), max(x), 100);
21
22     yeqnE = @(coefs, x) coefs(1).*exp(coefs(2).*x);
23     InitGuessE = [5.3509 0.01138]
24
25     yeqnP = @(coefs, x) coefs(1).*x.^coefs(2);
26     InitGuessP = [3.3893 0.72656]
27
28     yeqnS = @(coefs, x) coefs(1).*x./(coefs(2)+x);
29     InitGuessS = [21.011 3.4857]
30
31 %% Determine the function coefficients
32 fSSRE = @(coefs, x, y) sum(( y - yeqnE(coefs, x) ).^2)
33 [MyCoefsE, SrE] = fminsearch(@(MyCoefsDummy) fSSRE(MyCoefsDummy, x, y), InitGuessE)
34
35 fSSRP = @(coefs, x, y) sum(( y - yeqnP(coefs, x) ).^2)
36 [MyCoefsP, SrP] = fminsearch(@(MyCoefsDummy) fSSRP(MyCoefsDummy, x, y), InitGuessP)
37
38 fSSRS = @(coefs, x, y) sum(( y - yeqnS(coefs, x) ).^2)
39 [MyCoefsS, SrS] = fminsearch(@(MyCoefsDummy) fSSRS(MyCoefsDummy, x, y), InitGuessS)
40
41 %% Generate estimates and model
42 yhatE = yeqnE(MyCoefsE, x);
43 ymodelE = yeqnE(MyCoefsE, xmodel);
44
45 yhatP = yeqnP(MyCoefsP, x);
46 ymodelP = yeqnP(MyCoefsP, xmodel);
47
48 yhatS = yeqnS(MyCoefsS, x);
49 ymodelS = yeqnS(MyCoefsS, xmodel);
50
51
52 %% Calculate statistics
53 % Compute sum of the squares of the data residuals
54 St = sum(( y - mean(y) ).^2)
```

```

55
56 % Compute sum of the squares of the estimate residuals
57 SrE = sum(( y - yhatE ).^2)
58 SrP = sum(( y - yhatP ).^2)
59 SrS = sum(( y - yhatS ).^2)
60
61 % Compute the coefficient of determination
62 r2E = (St - SrE) / St
63 r2P = (St - SrP) / St
64 r2S = (St - SrS) / St
65
66
67 %% Generate plots
68 figure(1)
69 plot(x, y, 'ko', xmodel, ymodelE, 'k-')
70 title('Nonlinear Metabolism as a Function of Mass(Exponential) (cy111)')
71 xlabel('Mass(kg)')
72 ylabel('Metabolism(watt)')
73 print -depsc M21
74
75
76 figure(2)
77 plot(xmodel, log(ymodelE), x, log(y), 'ko')
78 xlabel('Mass(kg)')
79 ylabel('log(Metabolism)')
80 title('Nonlinear Transformation Linearized Exponential Model (cy111)')
81 print -depsc M22
82
83 figure(3)
84 plot(x, y, 'k-', xmodel, ymodelP, 'k:')
85 xlabel('Mass(kg)')
86 ylabel('Metabolism(watt)')
87 title('Nonlinear Metabolism as a Function of Mass(Power) (cy111)')
88 print -depsc M23
89
90 figure(4)
91 plot(log10(xmodel), log10(ymodelP), log10(x), log10(y), 'ko')
92 xlabel('log10(Mass)')
93 ylabel('log10(Metabolism)')
94 title('Nonlinear Transformation Power Model (cy111)')
95 print -depsc M24
96 figure(5)
97 plot(x, y, 'ko', xmodel, ymodelS, 'k--')
98 xlabel('Mass(kg)')
99 ylabel('Metabolism(watt)')
100 title('Nonlinear Metabolism as a Function of Mass(Saturation) (cy111)')
101 print -depsc M25
102 figure(6)
103 plot(1./xmodel, 1./ymodelS, 1./x, 1./y, 'ko')
104 xlabel('1/Mass(1/kg)')
105 ylabel('1/Metabolism(1/watt)')
106 title('Nonlinear Transformation Saturation Model (cy111)')
107 print -depsc M26
108
109

```

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

B Figures

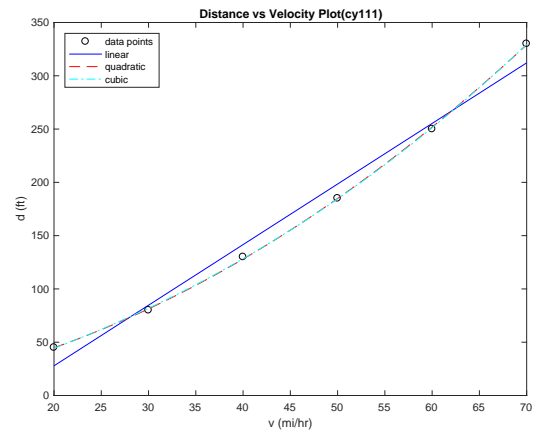


Figure 1: Distance Velocity Plot

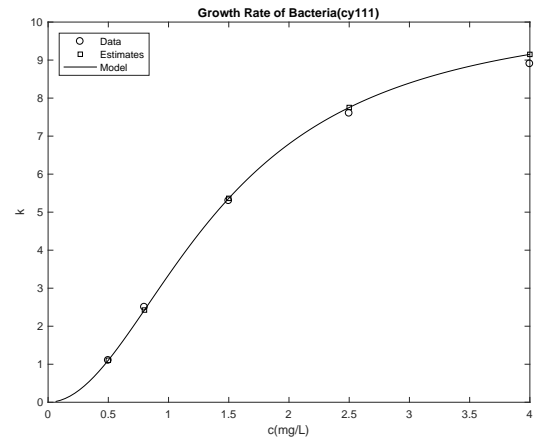


Figure 2: Bacteria Plot

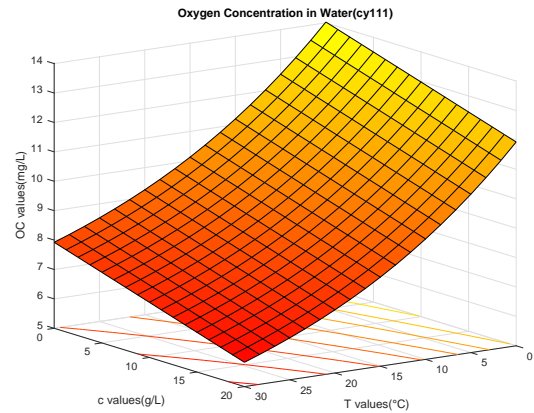


Figure 3: Oxygen Concentration Surface

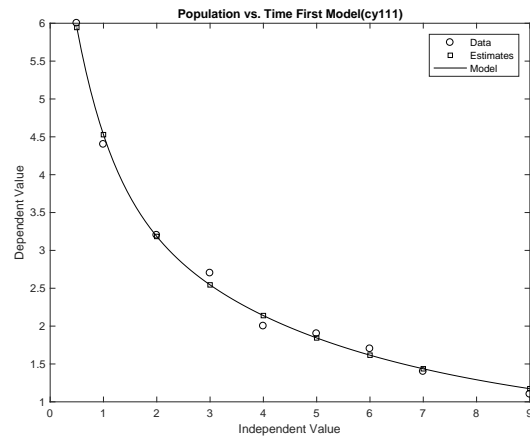


Figure 4: Population vs Time First Model

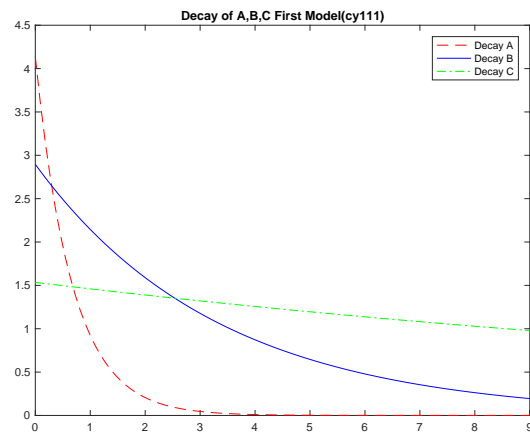


Figure 5: Decay of A,B,C First Model

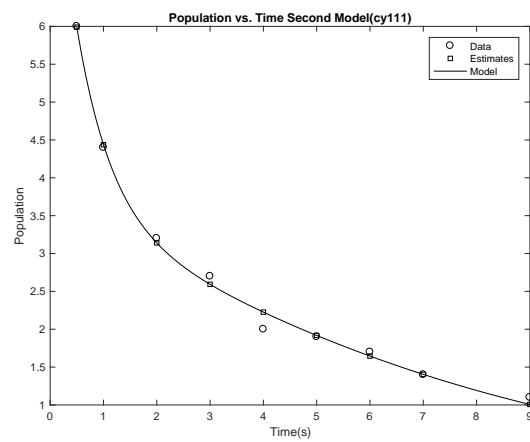


Figure 6: Population vs Time Second Model



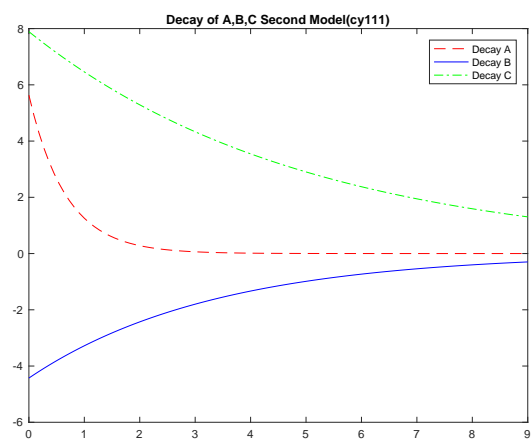


Figure 7: Decay of A,B,C Second Model

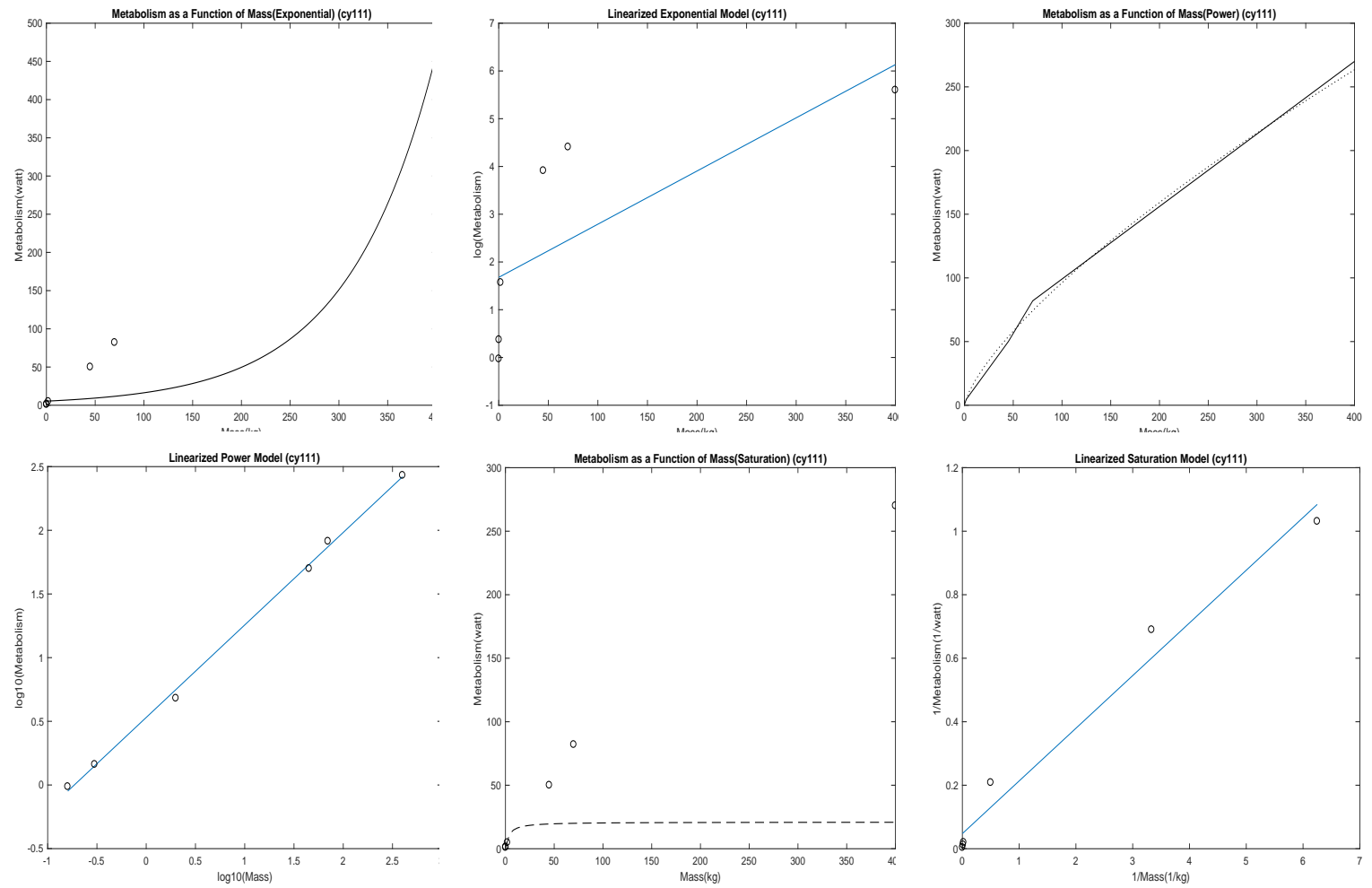


Figure 8: Untransformed and Transformed Plots for Linearized Metabolism Models

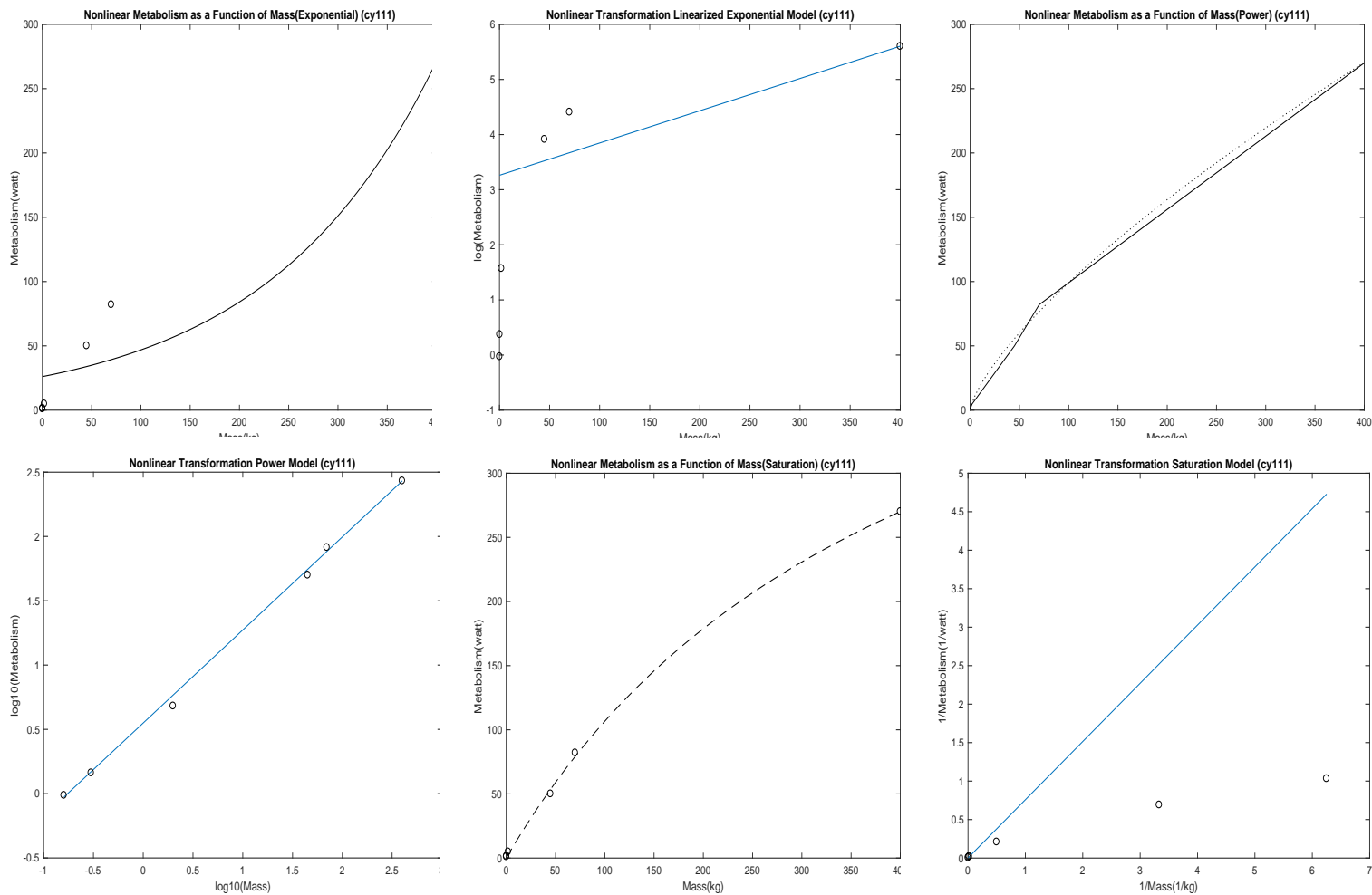


Figure 9: Untransformed and Transformed Plots for Unlinearized Metabolism Models