Table of Contents

Part 4: Curve Fitting and Modeling

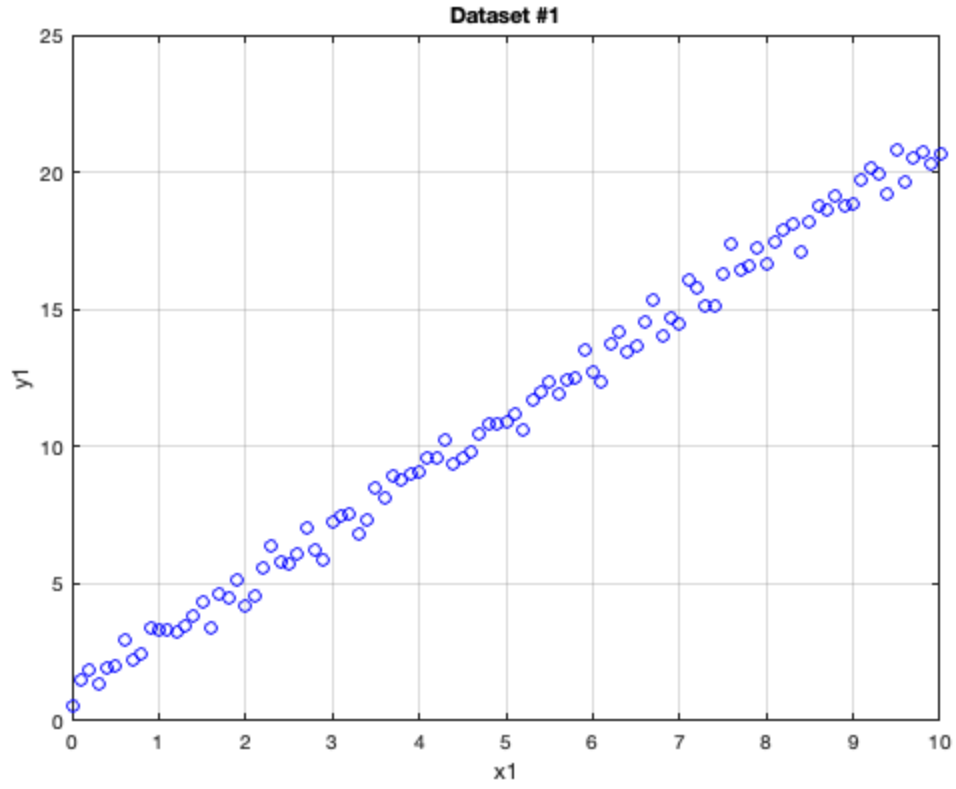
Kan Kanjanapas (Ph.D.)

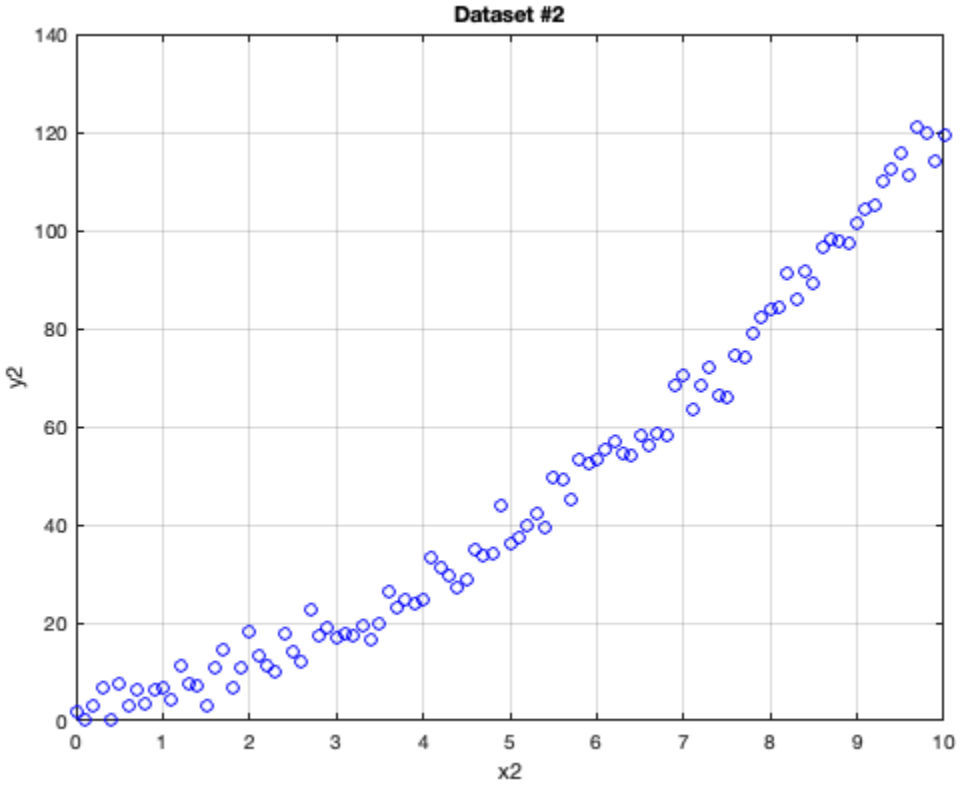
```
clc;
close all;
clear all;
```

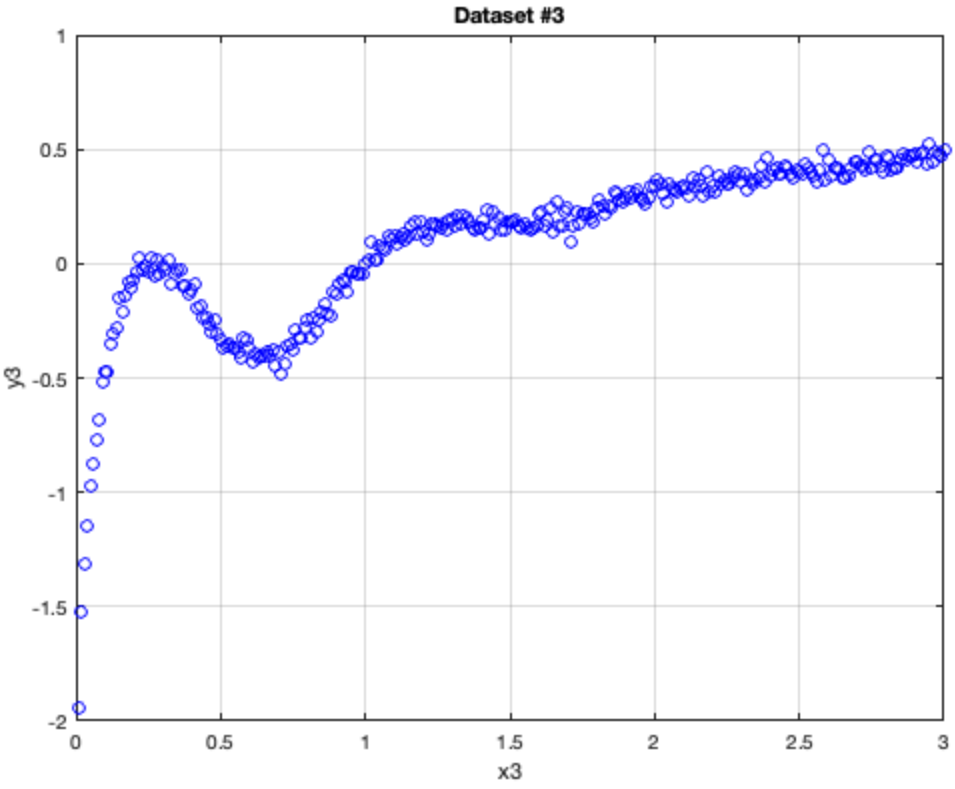
Initialization

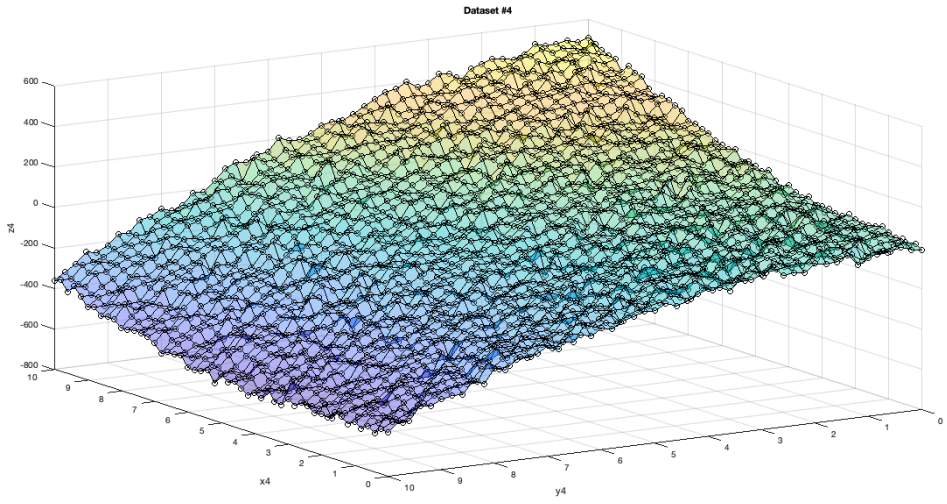
Let's generate data

```
Data_Generation;
```


Dataset #1
y1
x1
0
5
10
15
20
25
0
1
2
3
4
5
6
7
8
9
10


Dataset #2
y2
x2
0
20
40
60
80
100
120
140
0
1
2
3
4
5
6
7
8
9
10



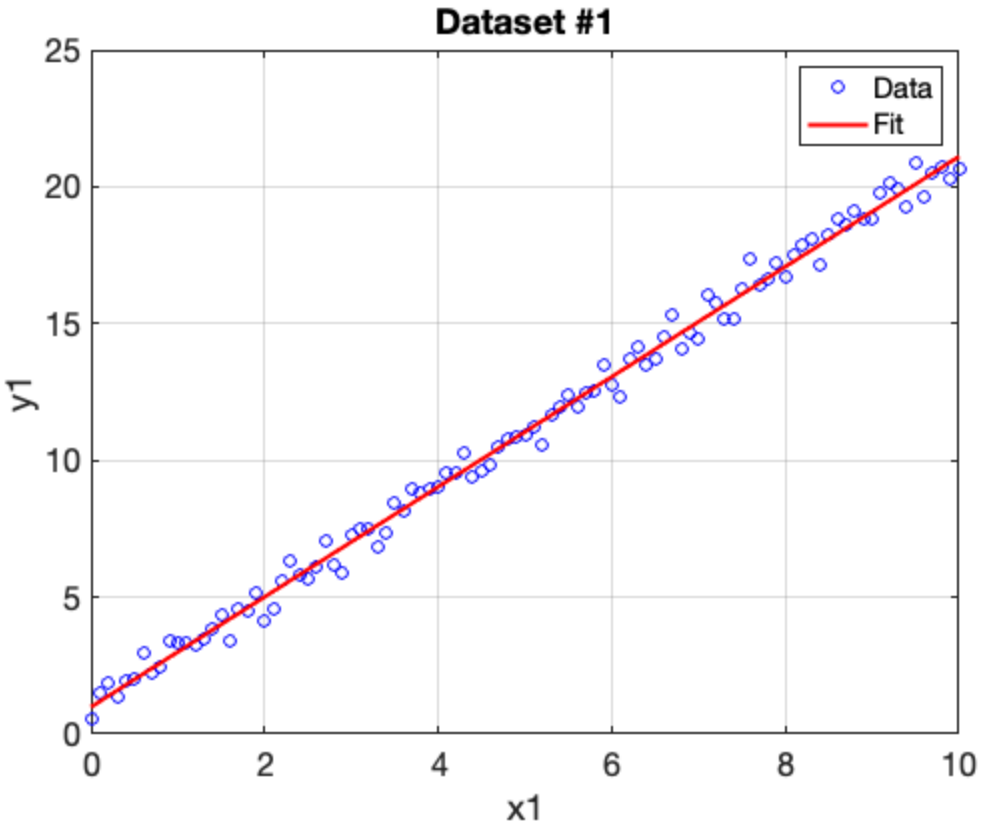


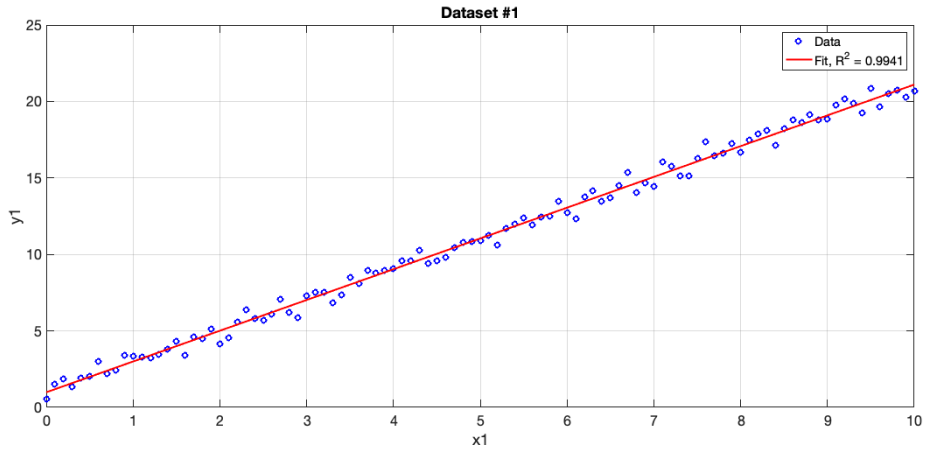
Part 1: Basic Fit

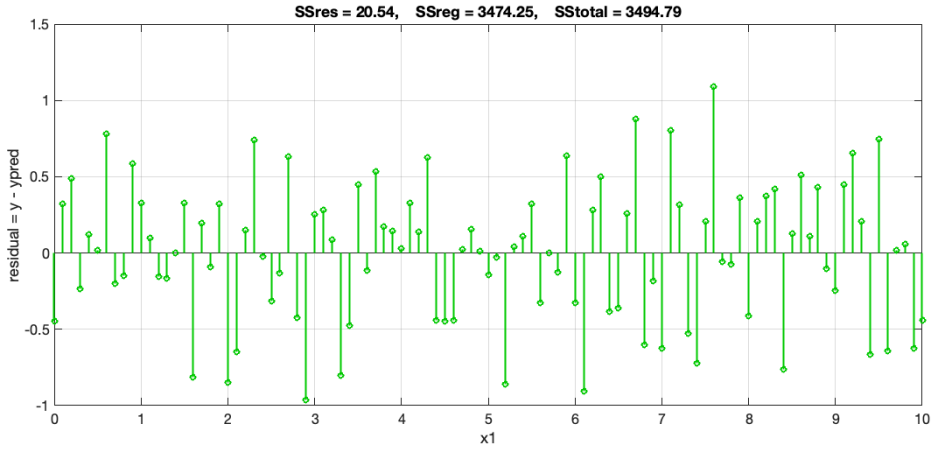
```
x1;
y1;
% figure,plot(x1,y1,'o','Color','b'); xlabel('x1'); ylabel('y1');
 title('Dataset #1'); grid on; set(gca, 'FontSize', 16);
```

```
% Method 1: pinv

% [1 x1][c] = [y1]
% [1 x2][m]   [y2]
%  ...        ...
% [1 xn]      [yn]

A1 = [ ones(length(x1),1)   x1 ];
B1 = y1;

% Solve for coefficients
P = A1\B1;

c1 = P(1,1);
m1 = P(2,1);

% Prediction
y1_pred = m1*x1 + c1;


% Plot to evaluate fit
figure;
plot(x1,y1,'o','Color','b');
hold on;
plot(x1,y1_pred,'Color','r', 'LineWidth', 2);
xlabel('x1');
ylabel('y1');
title('Dataset #1');
h_legend = legend('Data', 'Fit');
set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
grid on;
set(gca, 'FontSize', 16);



% The total sum of squares = n*Variance of data
y1_bar = mean(y1);
SS_total = sum( (y1 - y1_bar).^2 );


% The regression sum of squares:
SS_reg = sum( (y1_pred - y1_bar).^2 );


% The sum of squares of residuals;
res = y1 - y1_pred;
SS_res = sum( (y1 - y1_pred).^2 );


% SS_total = SS_reg + SS_res

% The most general definition of the coefficient of determination is
r_squared_1 = 1 - SS_res/SS_total;
```

```
r_squared_2 = SS_reg/SS_total;


figure;
set(gcf, 'Position', [0 0 2560 2560]/2);

subplot(2,1,1);
plot(x1,y1,'o','Color','b', 'LineWidth', 2);
hold on;
plot(x1,y1_pred,'Color','r', 'LineWidth', 2);
xlabel('x1');
ylabel('y1');
title('Dataset #1');
h_legend = legend('Data', sprintf('Fit, R^2 = %.4f', r_squared_1));
set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
grid on;
set(gca, 'FontSize', 16);

subplot(2,1,2);
stem(x1, res, 'LineWidth', 2, 'Color', [0  0.8  0]);
xlabel('x1');
ylabel('residual = y - ypred');
title(sprintf('SSres = %.2f,    SSreg = %.2f,    SStotal = %.2f',
 SS_res, SS_reg, SS_total));
grid on;
set(gca, 'FontSize', 16);
```







Part 2.1

```
% mdl = fitlm(x1, y1);

fitType = 'poly1';
fitobject_1 = fit(x1, y1, fitType);
fitobject_1.p1;
fitobject_1.p2;

[fitobject_1, gof_1, output_1] = fit(x1, y1, fitType)
fitobject_1
gof_1
output_1



% % % fitobject_1 =
```

```
% % %
% % %      Linear model Poly1:
% % %      fitobject_1(x) = p1*x + p2
% % %      Coefficients (with 95% confidence bounds):
% % %        p1 =       1.991  (1.956, 2.025)
% % %        p2 =      0.9938  (0.7949, 1.193)
% % %
% % %
% % %
% % % gof_1 =
% % %
% % %   struct with fields:
% % %
% % %               sse: 25.5084
% % %           rsquare: 0.9926
% % %               dfe: 99
% % %         adjrsquare: 0.9925
% % %              rmse: 0.5076
% % %
% % %
% % % output_1 =
% % %
% % %   struct with fields:
% % %
% % %            numobs: 101
% % %          numparam: 2
% % %         residuals: [101×1 double]
% % %          Jacobian: [101×2 double]
% % %          exitflag: 1
% % %         algorithm: 'QR factorization and solve'
% % %        iterations: 1


fitobject_1 =

     Linear model Poly1:
     fitobject_1(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =       2.012  (1.981, 2.043)
       p2 =      0.9866  (0.8081, 1.165)

gof_1 =

  struct with fields:

           sse: 20.5441
       rsquare: 0.9941
           dfe: 99
     adjrsquare: 0.9941
          rmse: 0.4555


output_1 =
```

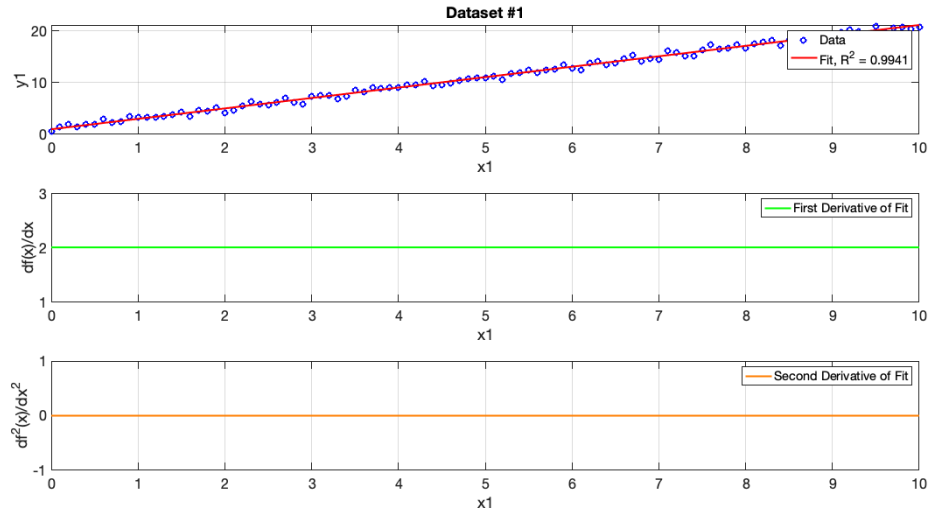
```
  struct with fields:

        numobs: 101
     numparam: 2
    residuals: [101×1 double]
     Jacobian: [101×2 double]
     exitflag: 1
    algorithm: 'QR factorization and solve'
   iterations: 1


fitobject_1 =

     Linear model Poly1:
     fitobject_1(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =       2.012  (1.981, 2.043)
       p2 =      0.9866  (0.8081, 1.165)

gof_1 =

  struct with fields:

           sse: 20.5441
       rsquare: 0.9941
           dfe: 99
    adjrsquare: 0.9941
          rmse: 0.4555


output_1 =

  struct with fields:

        numobs: 101
     numparam: 2
    residuals: [101×1 double]
     Jacobian: [101×2 double]
     exitflag: 1
    algorithm: 'QR factorization and solve'
   iterations: 1
```

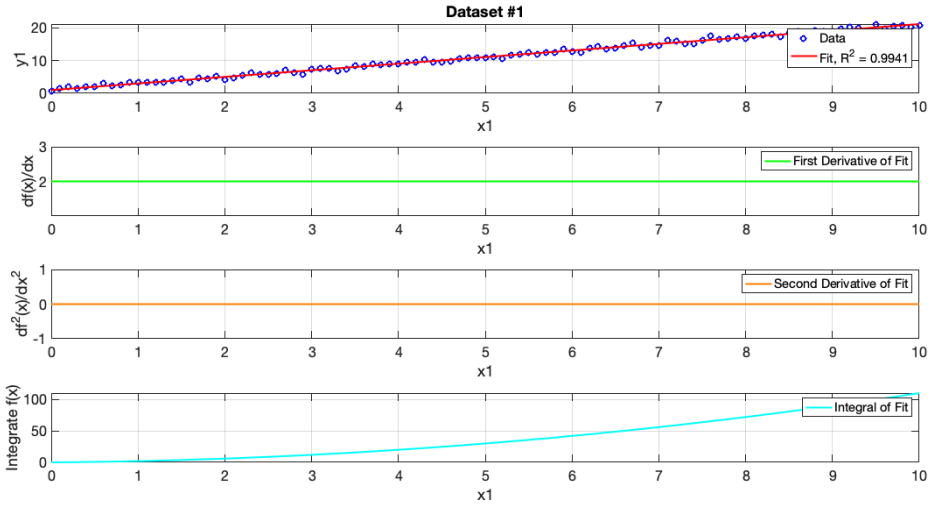
Part 2.2: Evaluate Fit

```
% % % % See Fit Postprocessing with fit object
% % % coeffnames
% % % coeffvalues
% % % feval
% % % differentiate
% % % integrate

% Given coefficient from fitobject --> predict y given x
```

```
y1_feval = feval( fitobject_1, x1 );
% figure,plot(x1, y1_pred, 'c', x1, y1_feval, 'r--', 'LineWidth', 2);



% Extract coefficient names from fit object
coeffs_1 = coeffnames( fitobject_1 );


% Extract coefficent values from fit object
coeffvals_1 = coeffvalues( fitobject_1 );


% Differentiate fit object
%[fx, fy, fxx, fxy, fyy] = differentiate(FO, X, Y)
[fx_1, fxx_1] = differentiate( fitobject_1, x1);

figure;
set(gcf, 'Position', [0 0 2560 1280]/2);
for ii = 1:1

    subplot(3,1,1);
    plot(x1,y1,'o','Color','b', 'LineWidth', 2);
    hold on;
    plot(x1,y1_pred,'Color','r', 'LineWidth', 2);
    xlabel('x1');
    ylabel('y1');
    title('Dataset #1');
    h_legend = legend('Data', sprintf('Fit, R^2 = %.4f',
 r_squared_1));
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);


    subplot(3,1,2);
    plot(x1, fx_1,'Color','g', 'LineWidth', 2);
    xlabel('x1');
    ylabel('df(x)/dx');
    h_legend = legend('First Derivative of Fit');
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);


    subplot(3,1,3);
    plot(x1, fxx_1,'Color', [1 0.5 0], 'LineWidth', 2);
    xlabel('x1');
    ylabel('df^2(x)/dx^2');
    h_legend = legend('Second Derivative of Fit');
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);
```

```
end



% Integrate fit object
x1_int = integrate(fitobject_1, x1, 0);
figure;
set(gcf, 'Position', [0 0 2560 1280]/2);
for ii = 1:1

    subplot(4,1,1);
    plot(x1,y1,'o','Color','b', 'LineWidth', 2);
    hold on;
    plot(x1,y1_pred,'Color','r', 'LineWidth', 2);
    xlabel('x1');
    ylabel('y1');
    title('Dataset #1');
    h_legend = legend('Data', sprintf('Fit, R^2 = %.4f',
 r_squared_1));
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);


    subplot(4,1,2);
    plot(x1, fx_1,'Color','g', 'LineWidth', 2);
    xlabel('x1');
    ylabel('df(x)/dx');
    h_legend = legend('First Derivative of Fit');
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);


    subplot(4,1,3);
    plot(x1, fxx_1,'Color', [1 0.5 0], 'LineWidth', 2);
    xlabel('x1');
    ylabel('df^2(x)/dx^2');
    h_legend = legend('Second Derivative of Fit');
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);


    subplot(4,1,4);
    plot(x1, x1_int,'Color', 'c', 'LineWidth', 2);
    xlabel('x1');
    ylabel('Integrate f(x)');
    h_legend = legend('Integral of Fit');
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);

end
```

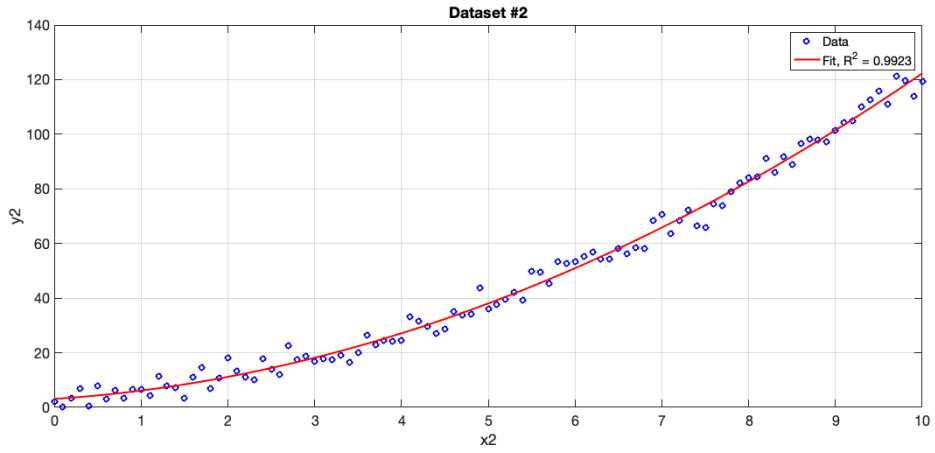


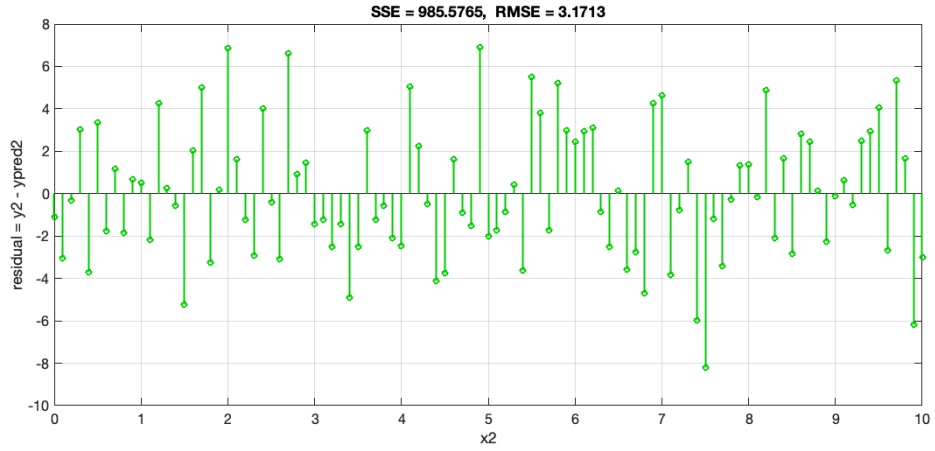


Part 2.3 Polynomial Fit

```
x2;
y2;


fitType2 = fittype('poly2'); % upto poly9 Y = p1*x^9+p2*x^8+...+p10
options2 = fitoptions;
% Search List of Library Models for Curve and Surface Fitting
 ***********


[fitobject_2, gof_2, output_2] = fit(x2, y2, fitType2, options2);

% Linear model Poly2:
%      fitobject_2(x) = p1*x^2 + p2*x + p3
```

```
%      Coefficients (with 95% confidence bounds):
%        p1 =       1.086  (1.012, 1.159)
%        p2 =       1.135  (0.3714, 1.898)
%        p3 =       4.377  (2.726, 6.028)


fitobject_2.p1;
fitobject_2.p2;
fitobject_2.p3;

y2_feval = feval( fitobject_2, x2 );


A2 = [x2.^2   x2    ones(length(x2),1)  ];
B2 = y2;

% Solve for coefficients
P2 = A2\B2;
res2 = y2 - y2_feval;


figure;
set(gcf, 'Position', [0 0 2560 2560]/2);
for ii = 1:1

    subplot(2,1,1);
    plot(x2, y2,'o','Color','b', 'LineWidth', 2);
    hold on;
    plot(x2, y2_feval,'Color','r', 'LineWidth', 2);
    xlabel('x2');
    ylabel('y2');
    title('Dataset #2');
    h_legend = legend('Data', sprintf('Fit, R^2 = %.4f',
 gof_2.rsquare));
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);

    subplot(2,1,2);
    stem(x2, res2, 'LineWidth', 2, 'Color', [0  0.8  0]);
    xlabel('x2');
    ylabel('residual = y2 - ypred2');
    title(sprintf('SSE = %.4f,  RMSE = %.4f',  gof_2.sse,
 gof_2.rmse));
    grid on;
    set(gca, 'FontSize', 16);

end
```



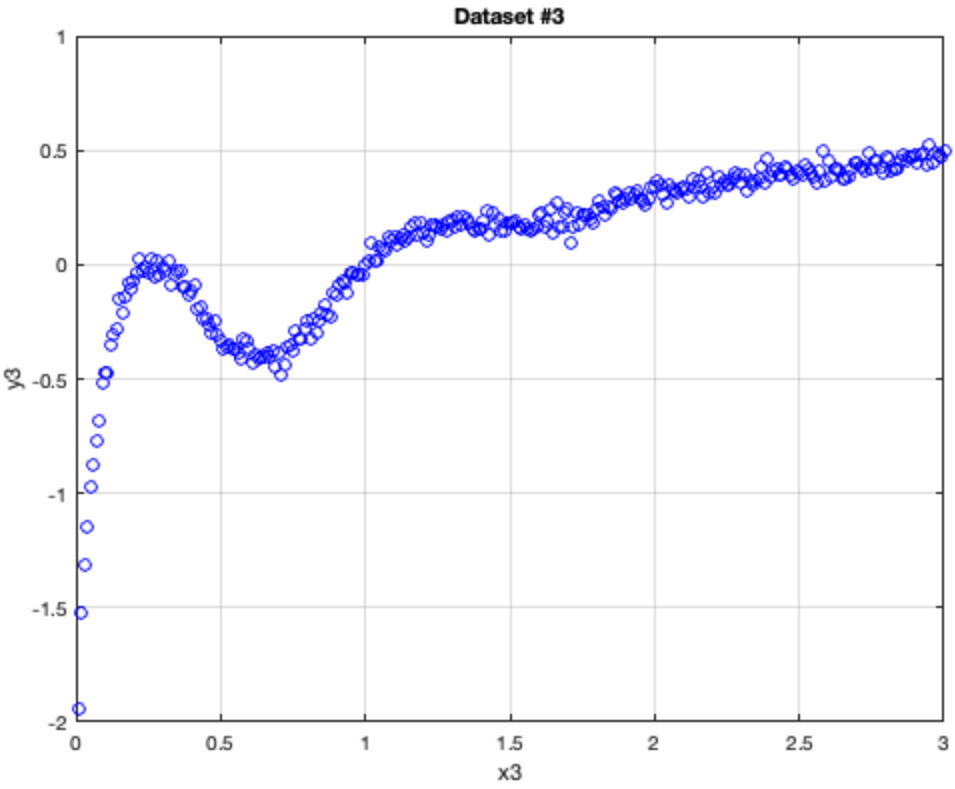


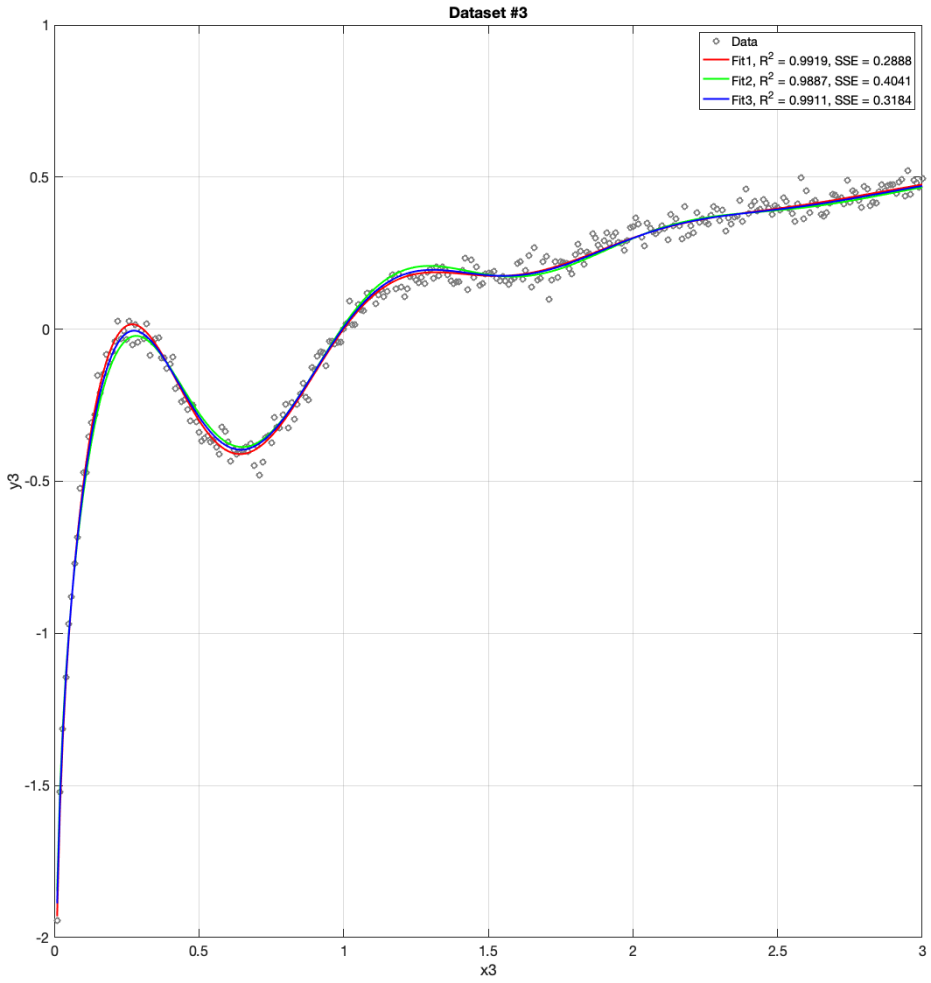
Part 2.4 Arbitrary Function Fit

```
clc;
close all;


Data3 = load('workspace_dataset3.mat');
x3 = Data3.x3;
y3 = Data3.y3;
figure,plot(x3,y3,'o','Color','b'); xlabel('x3'); ylabel('y3');
 title('Dataset #3'); grid on;

%polyfun3 = @(x) 1*sin(2*pi*1*x).*exp(-2*x) + 1*log10(1*x3) + 0;

f = 1;
a_vec = [1  0.8  0.9];
```

```
fitobject_3 = [];
gof_3       = [];
output_3    = [];
feval_3     = [];

StartPoint_Coeff = [-1.9  1.0  0.97  0.01];



for ii = 1:3

    a = a_vec(ii);
    fittype_3 = fittype( @(b, c, d, e, x) a*sin(2*pi*f*x).*exp(b*x) +
 c*log(d*x) + e);
    options = fitoptions( fittype_3 );
    options.Robust = 'on';

    [fitobject_3{ii}, gof_3{ii}, output_3{ii}]  = fit( x3, y3,
 fittype_3, 'StartPoint', StartPoint_Coeff );


    y3_feval{ii} = feval( fitobject_3{ii}, x3 );
    res_3{ii}   = y3 - y3_feval{ii};
end


figure;
set(gcf, 'Position', [0 0 2560 2560]/2);
for ii = 1:1

    ColorMatrix = [1 0 0; 0 1 0; 0 0 1];


    plot(x3, y3,'o','Color', 0.5*[1 1 1], 'LineWidth', 2);
    hold on;
    for ii = 1:3
        plot(x3, y3_feval{ii},'Color', ColorMatrix(ii,:), 'LineWidth',
 2);
    end
    xlabel('x3');
    ylabel('y3');
    title('Dataset #3');
    h_legend = legend('Data', sprintf('Fit1, R^2 = %.4f, SSE = %.4f',
 gof_3{1}.rsquare, gof_3{1}.sse), ...
                                sprintf('Fit2, R^2 = %.4f, SSE = %.4f',
 gof_3{2}.rsquare, gof_3{2}.sse), ...
                                sprintf('Fit3, R^2 = %.4f, SSE = %.4f',
 gof_3{3}.rsquare, gof_3{3}.sse));
    set(h_legend, 'Location', 'NorthEast', 'Color', [1 1 0.9]);
    grid on;
    set(gca, 'FontSize', 16);
```

```
end
```



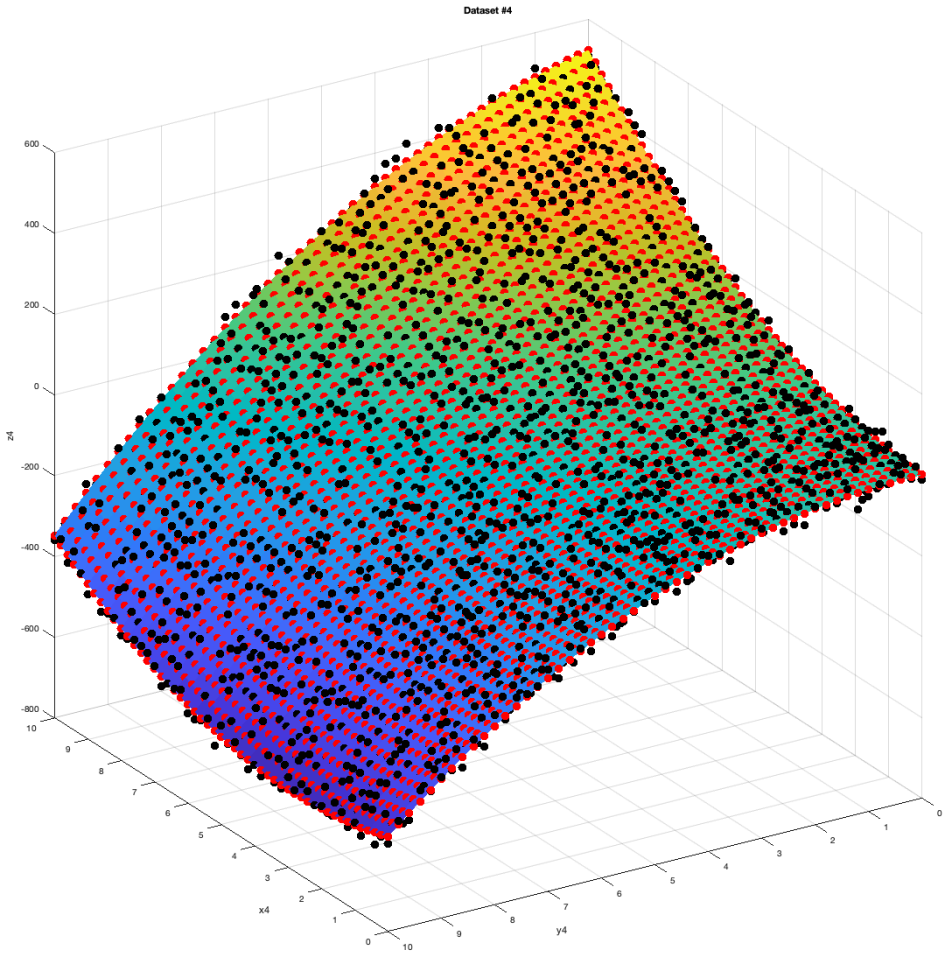


Part 2.5 Surface Fit

```
%x4, y4, z4;
% figure;
% set(gcf, 'Position', [0 0 2560 2560]/2);
% surf(x4, y4, z4, 'FaceAlpha', 0, 'Marker', 'o', 'MarkerFaceColor',
 0.5*[1 1 1], 'EdgeColor', 'none');
% xlabel('x4'); ylabel('y4'); zlabel('z4'); title('Dataset #4'); grid
 on; view(-122,24);
```

```
x4_vec = reshape(x4, numel(x4), 1);
y4_vec = reshape(y4, numel(y4), 1);
z4_vec = reshape(z4, numel(z4), 1);
```

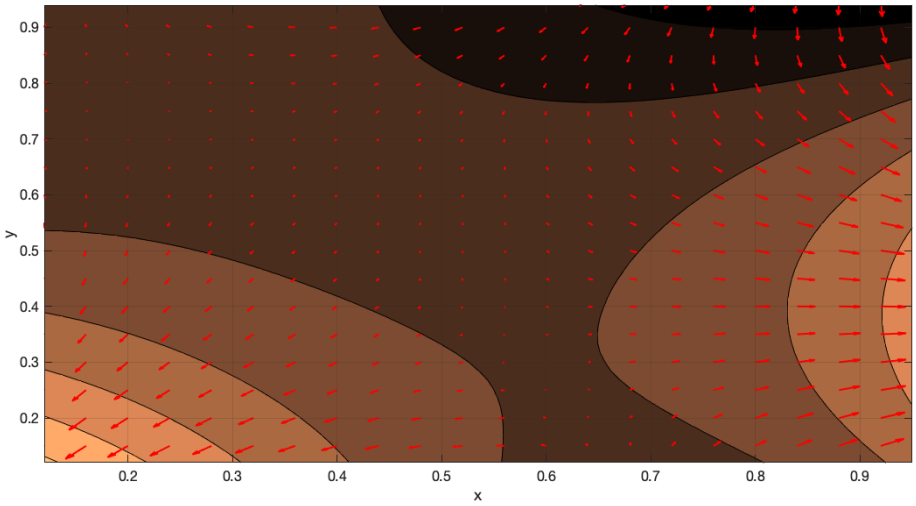
```
[fitobject_4, gof_4, output_4]  = fit( [x4_vec, y4_vec],
 z4_vec, 'poly22' );
z4_feval = feval( fitobject_4, [x4_vec, y4_vec]);

z4_feval_reshape = reshape(z4_feval, size(x4));


figure;
set(gcf, 'Position', [0 0 2560 2560]/2);
for ii = 1:1

    surf(x4, y4, z4, 'FaceAlpha',
 0, 'Marker', 'o', 'MarkerFaceColor', 'k', 'MarkerSize',
 10, 'EdgeColor', 'none');
    hold on;
    surf(x4, y4, z4_feval_reshape, 'FaceAlpha',
 1, 'Marker', 'o', 'MarkerFaceColor', 'r', 'MarkerSize',
 10, 'EdgeColor', 'none');

    xlabel('x4'); ylabel('y4'); zlabel('z4'); title('Dataset #4');
 grid on; view(-122,24);

end
```



Part 2.6 Surface Fit + Gradient

```
x6 = [0.64; 0.95; 0.21; 0.71; 0.24; 0.12; 0.61; 0.45; 0.46;...
      0.66; 0.77; 0.35; 0.66];

y6 = [0.42; 0.84; 0.83; 0.26; 0.61; 0.58; 0.54; 0.87; 0.26;...
      0.32; 0.12; 0.94; 0.65];

z6 = [0.49; 0.051; 0.27; 0.59; 0.35; 0.41; 0.3; 0.084; 0.6;...
      0.58; 0.37; 0.19; 0.19];

[fitobject_6, gof_6, output_6] = fit( [x6, y6],
 z6, 'poly32', 'normalize', 'on' );
z6_feval = feval( fitobject_6, [x6, y6]);

z6_feval_reshape = reshape(z6_feval, size(x6));
```

```
[xx6, yy6] = meshgrid( 0:0.04:1, 0:0.05:1 );


[fx6, fy6] = differentiate( fitobject_6, xx6, yy6 );


figure;
set(gcf, 'Position', [0 0 2560 1280]/2);

plot( fitobject_6, 'Style', 'Contour' );
hold on
h = quiver( xx6, yy6, fx6, fy6, 'r', 'LineWidth', 2 );
hold off
colormap( copper );
xlabel('x');
ylabel('y');
zlabel('z');
set(gca, 'FontSize', 16);
grid on;
```



Published with MATLAB® R2018a