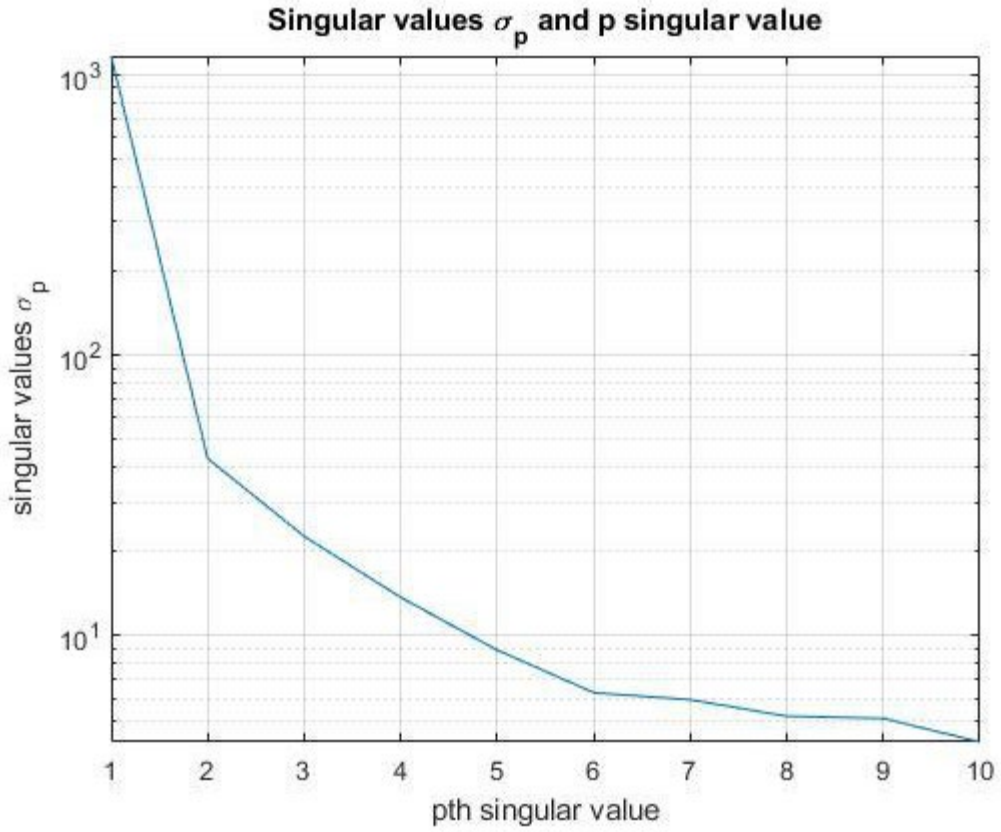
Contents

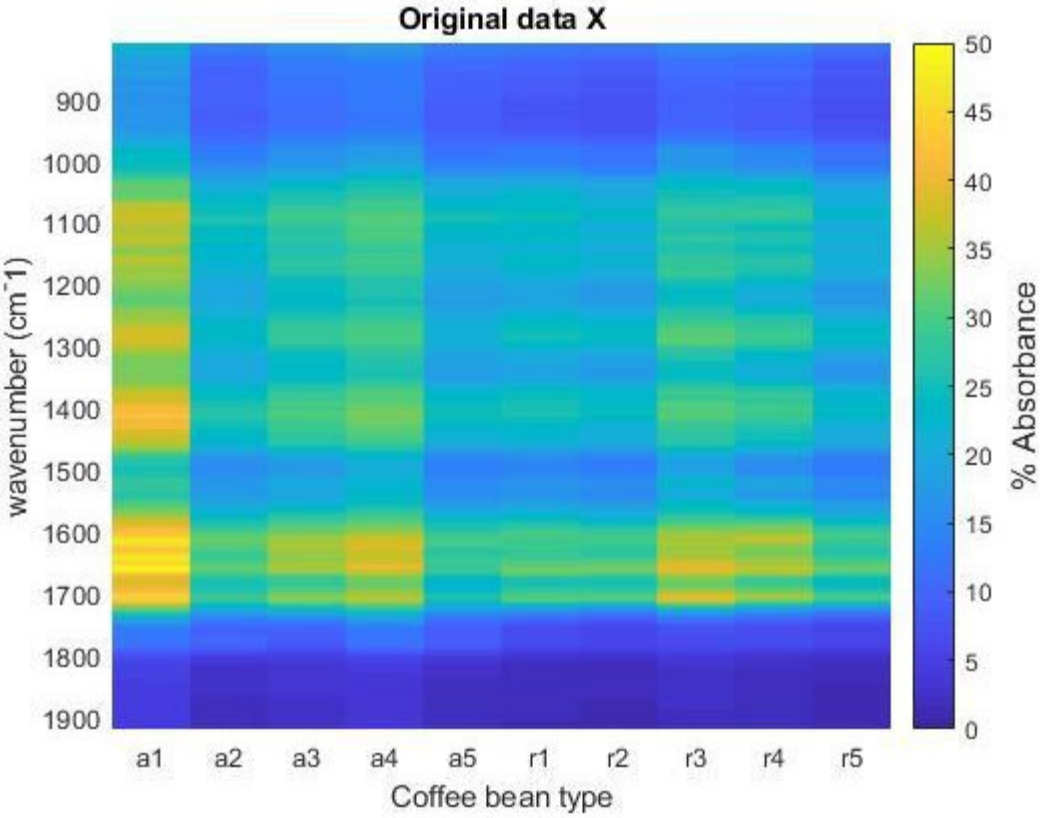

```
clear all
close all
```

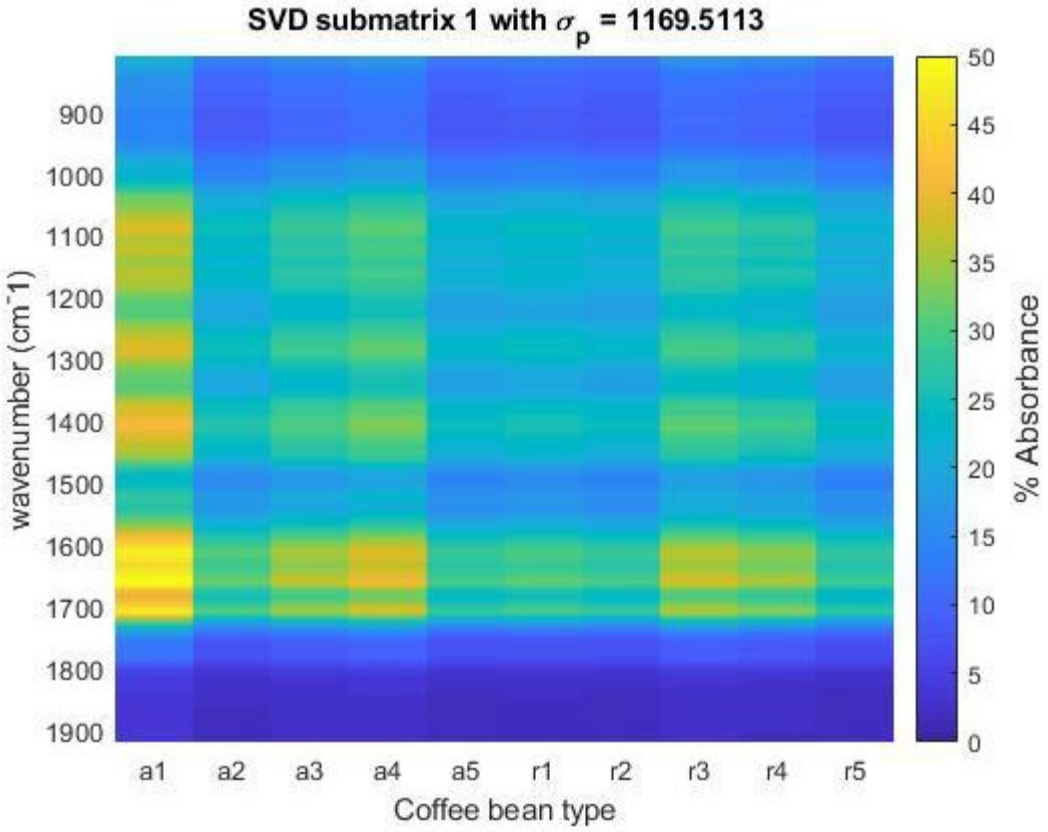
Problem 1 Task 1

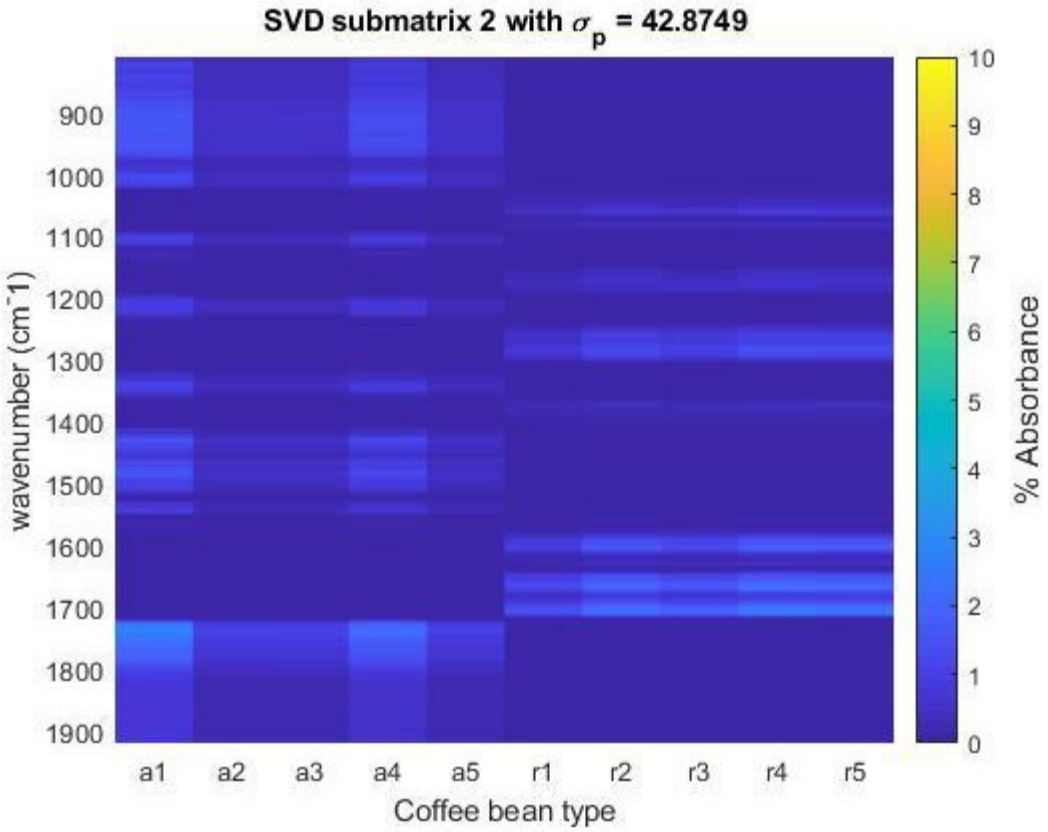
```
[wavenumber,  arabica1,  arabica2,  arabica3,  arabica4,  arabica5,  ...
    robusta1,  robusta2,  robusta3,  robusta4,  robusta5 ]   ...
    =  textread('Problem1_coffee_FTIR_data.txt','%f%f%f%f%f%f  %f%f%f%f%f',  'headerlines', 3);

X = [arabica1 arabica2 arabica3 arabica4 arabica5 robusta1 robusta2 robusta3...
    robusta4 robusta5];
[U, S, V] = svd(X);
S2 = [];
for i = 1:10
    S2(i) = S(i,i);
end

%echo S2, V

f0 = figure('Name', 'Singular Values');

h1 = semilogy(S2);
grid on
title('Singular values \sigma_p and p singular value')
xlabel('pth singular value')
ylabel('singular values \sigma_p')
% A1 = s(kk) * U(:,kk) * V(:,kk)';

%pcolor X and 3 SVD Spectra
xtick_strings = {'a1' 'a2' 'a3' 'a4' 'a5' 'r1' 'r2' 'r3' 'r4' 'r5'};
Dom = S2(1) * U(:,1) * V(:,1)';
Secondary = S2(2) * U(:,2) * V(:,2)';
Tertiary = S2(3) * U(:,3) * V(:,3)';
aaaa = NaN*ones(286,1);
aaa = NaN*ones(1,11);
X2 = [X aaaa];
X2 = [X2; aaa];
aa = [wavenumber; NaN];

Domplot = [Dom aaaa];
Domplot = [Domplot; aaa];
Secplot = [Secondary aaaa];
Secplot = [Secplot; aaa];
Terplot = [Tertiary aaaa];
Tertplot = [Terplot; aaa];
% X3 = [aa X2];
```

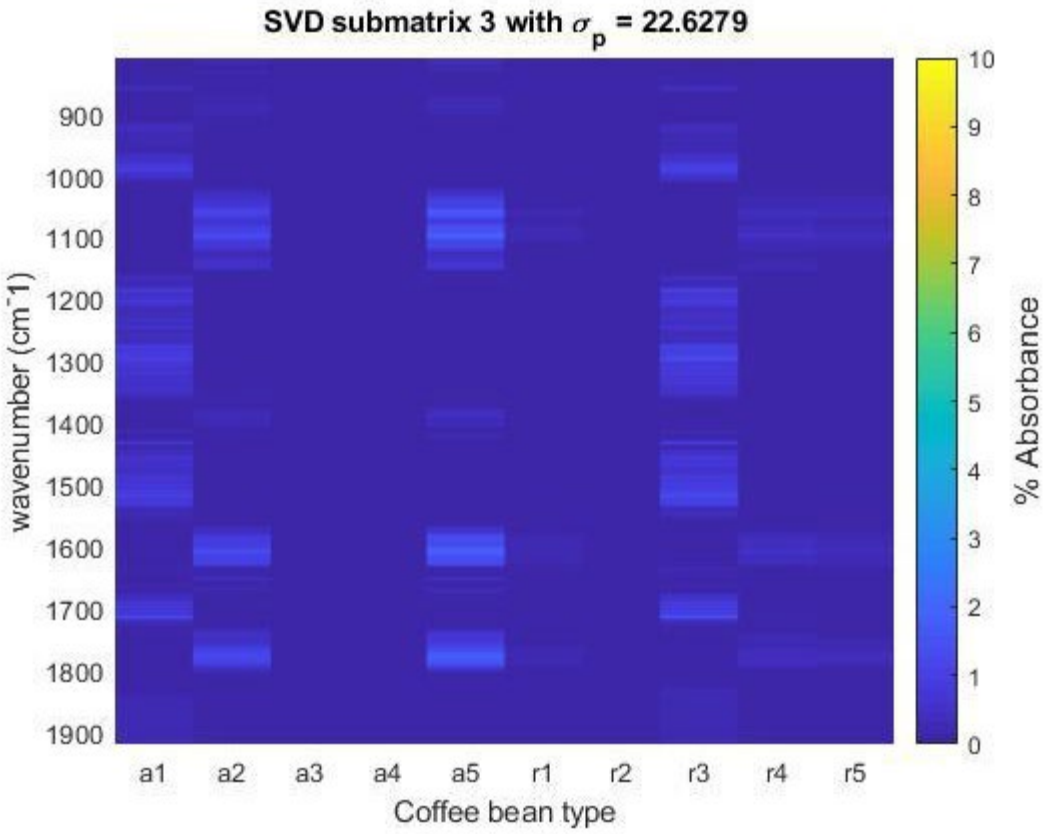
```
x = 1:11;
y = [wavenumber' 1.9206e+03];
[Xmesh1, Ymesh1] = meshgrid(x,y);


f1 = figure('Name', 'pcolor original X');
h2 = pcolor(Xmesh1, Ymesh1, X2);
h2.EdgeColor = 'none';
cb1 = colorbar;
caxis([0 50]);

set(gca, 'YDir', 'reverse')
title('Original data X')
set(gca, 'XTick', [1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5], 'XTickLabels', xtick_strings);

set(get(cb1, 'YLabel'), 'String', '% Absorbance', 'FontSize', 12);
xlabel('Coffee bean type');
ylabel('wavenumber (cm^-1)');

%SVD Spectra
f2 = figure('Name', 'pcolor Dominant');
h3 = pcolor(Xmesh1, Ymesh1, Domplot);
h3.EdgeColor = 'none';
cb2 = colorbar;
caxis([0 50]);
set(gca, 'YDir', 'reverse')
title('SVD submatrix 1 with \sigma_p = 1169.5113')
set(gca, 'XTick', [1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5], 'XTickLabels', xtick_strings);
set(get(cb2, 'YLabel'), 'String', '% Absorbance', 'FontSize', 12);
xlabel('Coffee bean type');
ylabel('wavenumber (cm^-1)');

f3 = figure('Name', 'pcolor Secondary');
h4 = pcolor(Xmesh1, Ymesh1, Secplot);
h4.EdgeColor = 'none';
cb3 = colorbar;
caxis([0 10]);
set(gca, 'YDir', 'reverse')
title('SVD submatrix 2 with \sigma_p = 42.8749')
set(gca, 'XTick', [1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5], 'XTickLabels', xtick_strings);
set(get(cb3, 'YLabel'), 'String', '% Absorbance', 'FontSize', 12);
xlabel('Coffee bean type');
ylabel('wavenumber (cm^-1)');


f4 = figure('Name', 'pcolor Tertiary');
h5 = pcolor(Xmesh1, Ymesh1, Tertplot);
h5.EdgeColor = 'none';
cb4 = colorbar;
caxis([0 10]);
set(gca, 'YDir', 'reverse')
title('SVD submatrix 3 with \sigma_p = 22.6279')
set(gca, 'XTick', [1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5], 'XTickLabels', xtick_strings);
set(get(cb4, 'YLabel'), 'String', '% Absorbance', 'FontSize', 12);
xlabel('Coffee bean type');
ylabel('wavenumber (cm^-1)');
```


Singular values σ_p and p singular value
singular values σ_p
10^3
10^2
10^1
1
2
3
4
5
6
7
8
9
10
pth singular value


Original data X
wavenumber (cm^{-1})
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
a1
a2
a3
a4
a5
r1
r2
r3
r4
r5
Coffee bean type
50
45
40
35
30
25
20
15
10
5
0
% Absorbance

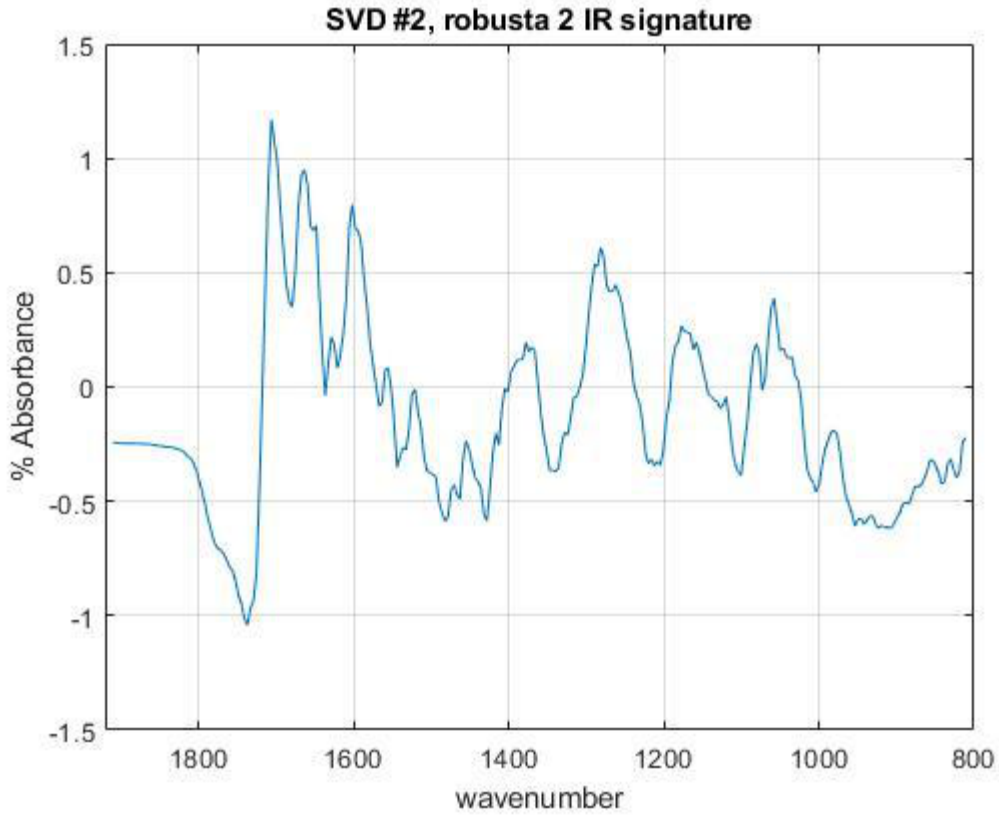

SVD submatrix 1 with σ_p = 1169.5113
wavenumber (cm^-1)
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
a1
a2
a3
a4
a5
r1
r2
r3
r4
r5
Coffee bean type
% Absorbance
0
5
10
15
20
25
30
35
40
45
50


SVD submatrix 2 with σ_p = 42.8749
wavenumber (cm^-1)
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
a1
a2
a3
a4
a5
r1
r2
r3
r4
r5
Coffee bean type
% Absorbance
0
1
2
3
4
5
6
7
8
9
10



Task #2

```
%SVD number 2 is the most like what we want
f5 = figure('Name', 'SVD Robusta 2 IR Signature');
g1 = plot(wavenumber, -1*Secondary(:,2));
grid on
title('SVD #2, robusta 2 IR signature');
xlabel('wavenumber')
ylabel('% Absorbance')
xlim([800 1.9206e+03])
set(gca, 'XDir', 'reverse')

%Answer to Task #2
%Caffeine and chlorogenic acid contents tend to be higher in Robusta
%compared with Arabica.
```



Task 3

```
%deveiation matrix for original X
xmean = [];
for i=1:10
    xmean(i) = mean(X(:,i));
end

xm_m = repmat(xmean, length(wavenumber), 1);

D = X - xm_m;

S = 1/(286-1) * D' * D;

[eigVec,eigVal] = eig(S); %echo these

mineigVec = min(eigVec);



%normalized by x1/min(x1)
V_norm = eigVec./mineigVec;

%echo this
```

Task 4

```
% sum(sum(eigVal))
% disp('The first three PCs can explain ')

TotVar = (eigVal(1,1) + eigVal(2,2) + eigVal(3,3))/(sum(sum(eigVal)))*100;
```

```
% The reason that five are positive and five are negative in the second PC
```

Echoing all Results etc.

```
diary vjprob1.txt

echo on
disp('Task 1')

V
S2

disp('Task 2')
disp('Caffeine and chlorogenic acid contents tend to be higher in Robusta compared with Arabica.')

disp('Task 3')
eigVal
eigVec
V_norm

disp('Task 4')
disp('The first three PCs can explain ')
TotVar
disp('Of the total variance')

disp('The reason why five values are negative and the other fiver are positive is that 5 entries
directly correspond with each group. Each group will have the same sign across the 5 beans.')
echo off
```

```
disp('Task 1')
Task 1

V

V =

  Columns 1 through 7

   -0.4410    0.4605   -0.4480   -0.4863   -0.3851    0.0367   -0.0400
   -0.2821    0.1672    0.3887    0.3499   -0.2680   -0.1205   -0.5182
   -0.3271    0.1688    0.0170   -0.1510    0.7831   -0.0553    0.1774
   -0.3577    0.3627    0.0414    0.5422    0.0352   -0.2624    0.2941
   -0.2632    0.1644    0.5596   -0.1938   -0.0030    0.3233    0.1099
   -0.2764   -0.2223    0.0733    0.0042    0.0944    0.6882   -0.2014
   -0.2623   -0.3630    0.0252    0.0471   -0.2786   -0.0511    0.5669
   -0.3364   -0.2948   -0.5406    0.3890    0.1335    0.1347   -0.2353
   -0.3133   -0.4032    0.1451   -0.3634    0.1136   -0.5597   -0.3426
   -0.2553   -0.3770    0.1169   -0.0326   -0.2192   -0.0082    0.2551

  Columns 8 through 10

    0.0384   -0.0474   -0.0356
   -0.0299   -0.5129   -0.0071
   -0.0244   -0.4279   -0.0997
    0.2903    0.4464   -0.0681
   -0.5246    0.3322    0.2254
    0.5796    0.0970   -0.0207
```

```
    0.1121   -0.3516    0.5087
   -0.4845    0.1072    0.1366
    0.1605    0.3099    0.1439
   -0.1692   -0.0365   -0.7967

S2

S2 =

   1.0e+03 *

  Columns 1 through 7

    1.1695    0.0429    0.0226    0.0137    0.0089    0.0063    0.0059

  Columns 8 through 10

    0.0052    0.0051    0.0042


disp('Task 2')
Task 2
disp('Caffeine and chlorogenic acid contents tend to be higher in Robusta compared with Arabica.')
Caffeine and chlorogenic acid contents tend to be higher in Robusta compared with Arabica.

disp('Task 3')
Task 3
eigVal

eigVal =

  Columns 1 through 7

  833.2564         0         0         0         0         0         0
         0    3.1618         0         0         0         0         0
         0         0    1.7724         0         0         0         0
         0         0         0    0.6588         0         0         0
         0         0         0         0    0.2774         0         0
         0         0         0         0         0    0.0612         0
         0         0         0         0         0         0    0.1374
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0

  Columns 8 through 10

         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
    0.1202         0         0
         0    0.0897         0
         0         0    0.0935

eigVec

eigVec =
```

```
  Columns 1 through 7

    0.4126   -0.4298    0.4991    0.4915   -0.3858   -0.0334    0.0280
    0.2740   -0.2419   -0.3602   -0.3479   -0.2686   -0.0173   -0.1602
    0.3169   -0.1877    0.0046    0.1528    0.7829   -0.1011   -0.0494
    0.3360   -0.3784    0.0005   -0.5398    0.0353   -0.0586   -0.2430
    0.2543   -0.2316   -0.5382    0.1937   -0.0029    0.2330    0.3366
    0.2896    0.1832   -0.0910   -0.0049    0.0941   -0.0295    0.6720
    0.2822    0.3527   -0.0657   -0.0509   -0.2780    0.5085   -0.0136
    0.3516    0.3288    0.5093   -0.3900    0.1336    0.1367    0.1253
    0.3362    0.3699   -0.1872    0.3600    0.1140    0.1427   -0.5756
    0.2768    0.3490   -0.1562    0.0293   -0.2188   -0.7951    0.0098

  Columns 8 through 10

   -0.0256    0.0456    0.0304
   -0.4861    0.5251   -0.0745
    0.2020    0.4106   -0.0772
    0.3086   -0.4331    0.3255
    0.0594   -0.3852   -0.4848
   -0.2363   -0.0141    0.5975
    0.5734    0.3467    0.0637
   -0.2674   -0.1429   -0.4612
   -0.3222   -0.2773    0.2005
    0.2522    0.0014   -0.1807

V_norm

V_norm =

  Columns 1 through 7

    1.6222    1.0000   -0.9275   -0.9105    1.0000    0.0420   -0.0487
    1.0773    0.5629    0.6693    0.6445    0.6962    0.0217    0.2783
    1.2460    0.4368   -0.0086   -0.2830   -2.0296    0.1272    0.0858
    1.3212    0.8805   -0.0010    1.0000   -0.0915    0.0738    0.4222
    1.0000    0.5389    1.0000   -0.3589    0.0075   -0.2931   -0.5848
    1.1386   -0.4263    0.1690    0.0090   -0.2439    0.0371   -1.1675
    1.1098   -0.8207    0.1221    0.0942    0.7207   -0.6396    0.0236
    1.3824   -0.7650   -0.9463    0.7224   -0.3464   -0.1720   -0.2177
    1.3219   -0.8607    0.3479   -0.6668   -0.2954   -0.1795    1.0000
    1.0882   -0.8120    0.2903   -0.0542    0.5673    1.0000   -0.0170

  Columns 8 through 10

    0.0526   -0.1053   -0.0627
    1.0000   -1.2125    0.1538
   -0.4155   -0.9482    0.1593
   -0.6349    1.0000   -0.6715
   -0.1223    0.8895    1.0000
    0.4861    0.0325   -1.2325
   -1.1795   -0.8005   -0.1314
    0.5501    0.3300    0.9513
    0.6628    0.6403   -0.4136
   -0.5189   -0.0032    0.3727


disp('Task 4')
Task 4
disp('The first three PCs can explain ')
```

```
The first three PCs can explain
TotVar

TotVar =

   99.8287

disp('Of the total variance')
Of the total variance

disp('The reason why five values are negative and the other fiver are positive is that 5 entries
directly correspond with each group. Each group will have the same sign across the 5 beans.')
The reason why five values are negative and the other fiver are positive is that 5 entries directly
correspond with each group. Each group will have the same sign across the 5 beans.
echo off
```

Published with MATLAB® R2018b

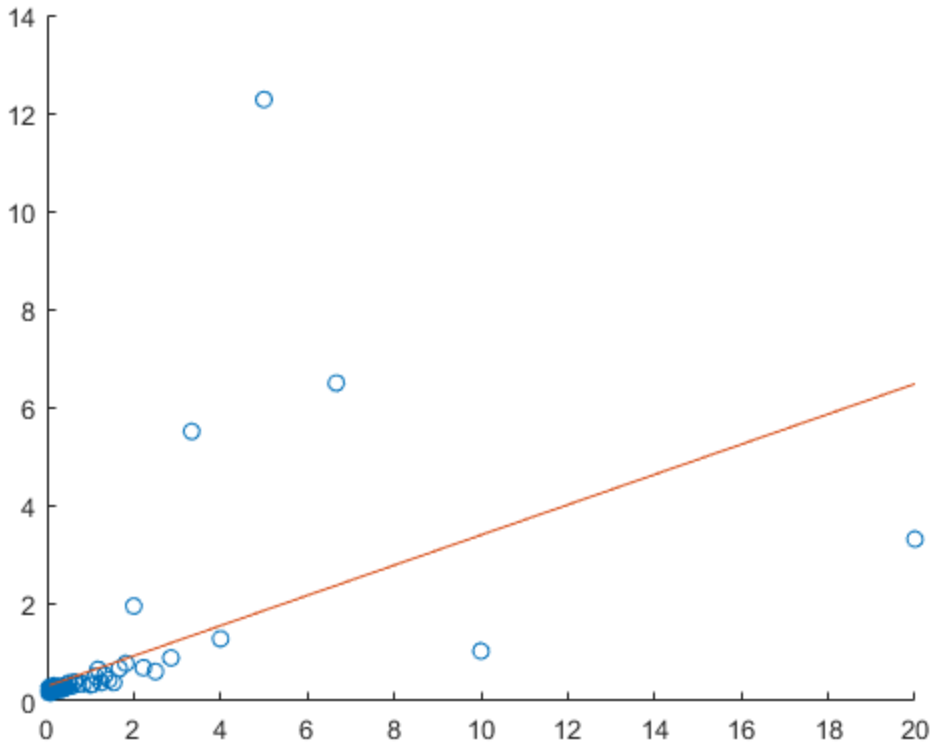
Table of Contents

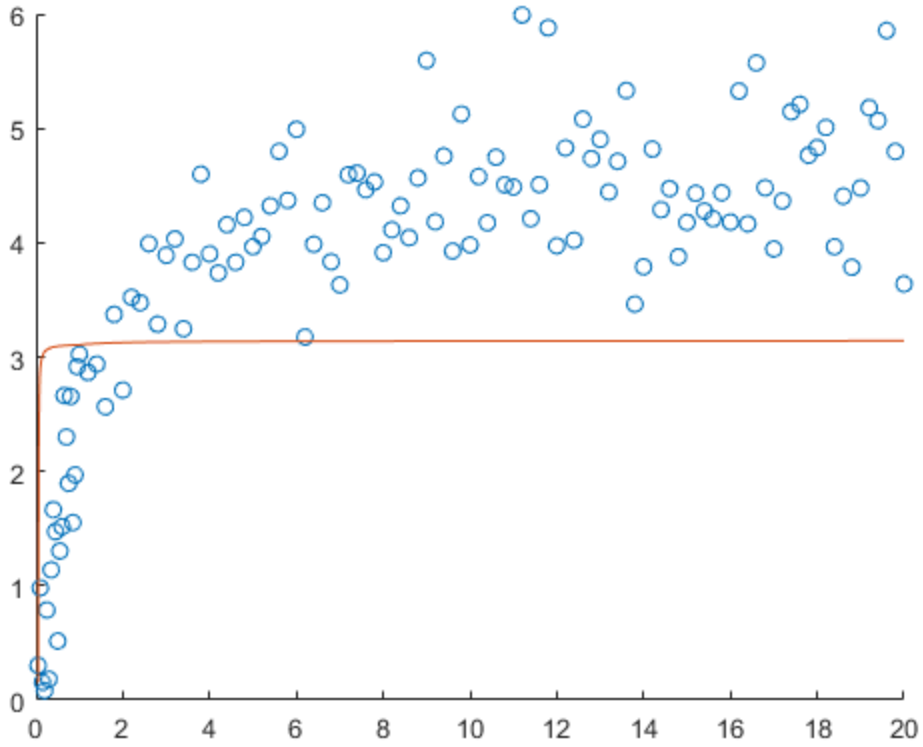

```
close all
clear all
```

Problem 2 Part 1

```
[data_ID, S, reactionrate] = textread('Problem2_kineticsData.txt', '%f
%f%f', 'headerlines', 1);
t = S;
y = reactionrate;

%take reciprocal to build A and b
t_recip = 1./S;
y_recip = 1./reactionrate;

%take reciprocal to build A and b
z = ones(115, 1);
A = [t_recip z];
b = y_recip;
G = A' * A;
b_new = A' * b;
x_ster = G\b_new;

yrecipfit = x_ster(1) .* t_recip + x_ster(2);

f0 = figure('Name', 'Reciprocal-Transformed Data');

h1 = scatter(t_recip, y_recip);
hold on
plot(t_recip, yrecipfit)
% xlabel(1/S)
% ylabel(1/y) %when i run this the plots just become single points
% title('Initial fit on reciprocal transformed data')

Vmax = 1/x_ster(2);
Km = x_ster(1) * Vmax;


f1 = figure('Name', 'Original Data');

h2 = scatter(t, y);
hold on
yorigfit = Vmax * (S)./(S + Km);
```

```
plot(t_recip, flip(yorigfit))
% ylabel(y)
% xlabel(S)
% title('Initial fit on original data')
```



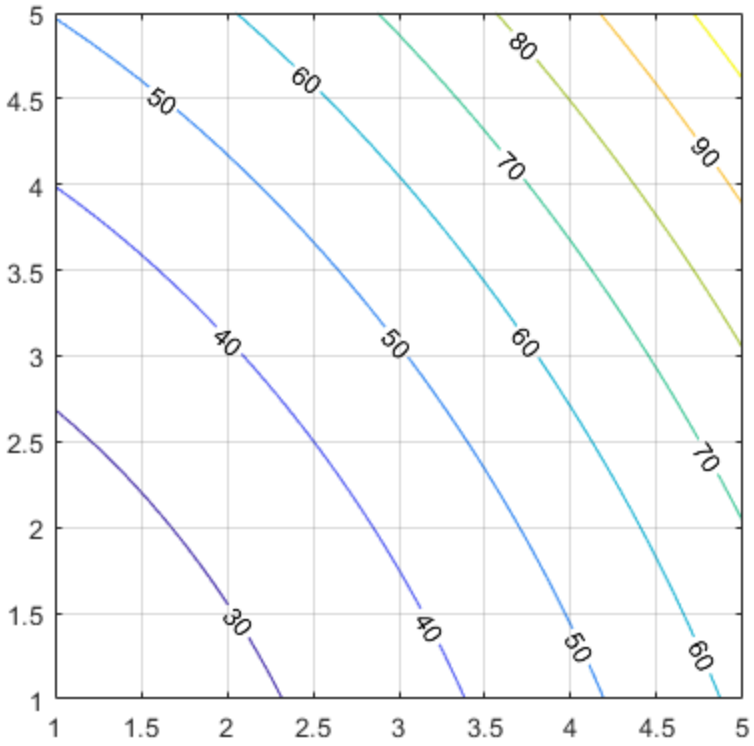


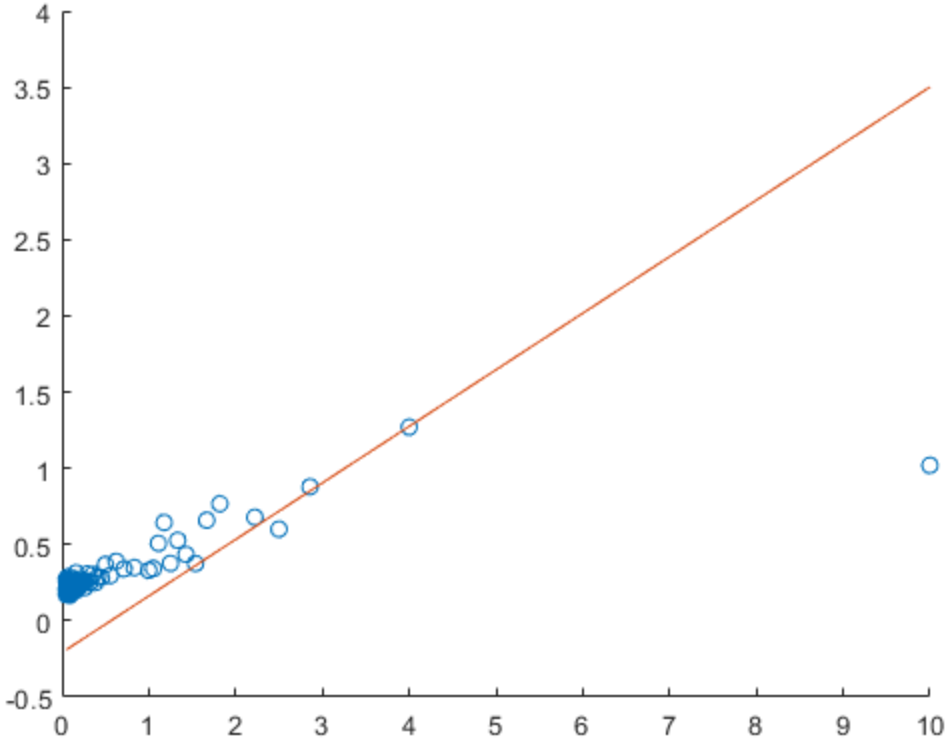
Task 2

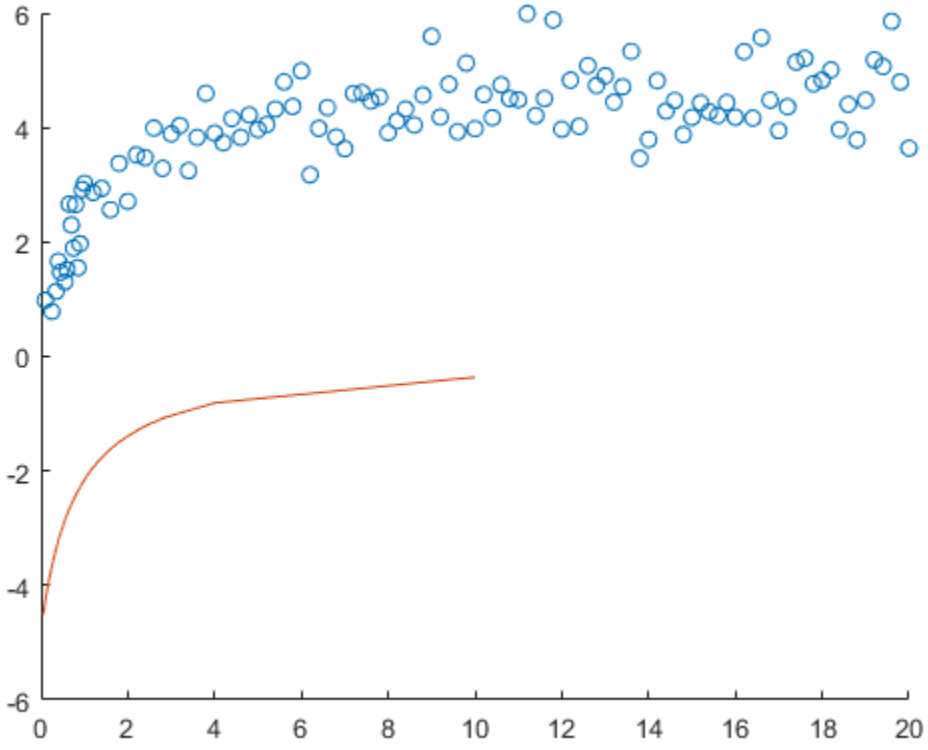
```
data_IDnew=data_ID(6:end);
t_new=[t(2); t(5); t(7:9); t(11:end)];
y_new=[y(2); y(5); y(7:9); y(11:end)];
tn_recip = 1./t_new;
yn_recip = 1./y_new;
z=ones(1,110);
An=[tn_recip z'];
bn=[yn_recip];
Gn=An'*An;
b_newn=An'*bn;
xn=Gn\b_newn;

x1 = 1:0.05:5;
x2 = 1:0.05:5;

[X1, X2] = meshgrid(x1, x2);

Z = (Gn(1)*X1.^2 + Gn(4) * X2.^2 + 2 * Gn(2) * X1.*X2) - 2 * (31.5457
 * X1 + X2 *32) + norm(b_newn)^2;
z2 = Z./100;
% ZP=Gn(1).*X1.^2+(2*Gn(4)).*X1.*X2+Gn(2).*X2.^2-(2*b_newn(1)).*X1-
(2*b_newn(2)).*X2+norm(b_newn)^2;
```

```
f2 = figure('Name', 'Contour Plot');
contour(X1,X2,z2, 'ShowText', 'on')
axis square
grid on

%wo

x0 = [3;1];

zaxisaltitude = (b_newn-Gn*x0)'*(b_newn-Gn*x0);
w0 = 2*(b_newn - Gn * x0);
alph = 0.5 * (w0' * w0)/(w0' * Gn * w0);
x1 = x0 + alph*w0;

rvec = 1;
xk = x0;
n = 2;
zaxisalttot = zaxisaltitude;
wktot = w0;
xktot = x0;
alphktot = alph;
while rvec>1e-4
    zaxisaltk = (b_newn-Gn*xk)'*(b_newn-Gn*xk);
    zaxisalttot = [zaxisalttot zaxisaltk];
    wk = 2*(b_newn - Gn * xk);
    alphk = 0.5 * (wk' * wk)/(wk' * Gn * wk);
    xk = xk + alph*wk;
    rvec = abs(zaxisaltk - zaxisalttot(n-1));
    wktot = [wktot wk];
    alphktot = [alphktot alphk];

    xktot = [xktot xk];
    n = n+1;
end

%couldn't get the plot to work properly

optimumx_ster = xktot(:,2);

yrecipfitnew = optimumx_ster(1) .* tn_recip + optimumx_ster(2);

f3 = figure('Name', 'Reciprocal-Transformed Data');

h1 = scatter(tn_recip, yn_recip);
hold on
plot(tn_recip, yrecipfitnew)
% xlabel(1/S)
% ylabel(1/y)
% title('New fit on Reciprocal Transformed Data')

%this obviously isn't right, but i'm not sure where i'm wrong

Vmax_new = 1/optimumx_ster(2);
Km_new = optimumx_ster(1) * Vmax;
```

```
f4 = figure('Name', 'Original Data');

h5 = scatter(t_new, y_new);
hold on
ynewfit = Vmax_new * (t_new)./(t_new + Km_new);
plot(tn_recip, ynewfit)
% ylabel(y)
% xlabel(S)
% title('New fit on Original Data')
```




4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
1
2
3
4
5
6
7
8
9
10


6
4
2
0
-2
-4
-6
0
2
4
6
8
10
12
14
16
18
20

Echoing Resuls

```
diary vjprob2.txt
echo on
disp('Part 1')
G
b_new
x_ster

data_ID(1)
data_ID(3)
data_ID(4)
data_ID(6)
data_ID(10)

Vmax
Km

disp('Part 2')

zaxisalttot
xktot
wktot
alphktot

Vmax_new
Km_new

echo off

disp('Part 1')
Part 1
G

G =

  642.7496   86.4750
   86.4750  115.0000

b_new

b_new =

  224.4653
   61.5112

x_ster

x_ster =

    0.3085
    0.3029
```

```
data_ID(1)

ans =

     1

data_ID(3)

ans =

     3

data_ID(4)

ans =

     4

data_ID(6)

ans =

     6

data_ID(10)

ans =

    10


Vmax

Vmax =

    3.3012

Km

Km =

    1.0183


disp('Part 2')
Part 2

zaxisalttot

zaxisalttot =

   1.0e+05 *
```

```
    2.9383    2.9383

xktot

xktot =

    3.0000    0.3710
    1.0000   -0.2086

wktot

wktot =

 -985.0231 -985.0231
 -452.8390 -452.8390

alphktot

alphktot =

    0.0027    0.0027


Vmax_new

Vmax_new =

   -4.7936

Km_new

Km_new =

    1.2248


echo off
```

Published with MATLAB® R2018b

Contents

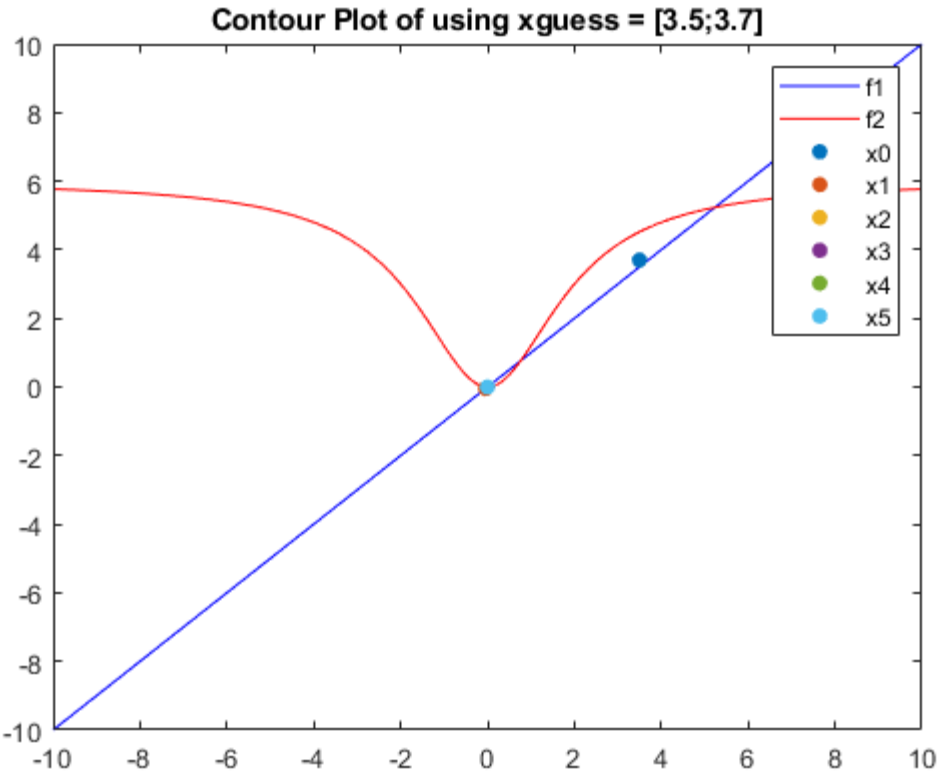

```
close all
clear all
```

Problem 3 Part 2

```
a = 1;
b = 0.5;
c = 3;
d = 4;

x1 = -10:0.1:10;
x2 = -10:0.1:10;
[X1, X2] = meshgrid(x1,x2);

f1 = -a.*X1 + X2;
f2 = ((c.*(X1.^2))./(X1.^2 + d)) - b.*X2;
figure;
[c1 c2] = contour(X1,X2,f1,[0 0], 'Color', 'b');
% c2.levels
hold on
[c3 c4] = contour(X1,X2,f2,[0 0], 'Color', 'r');
legend('f1', 'f2');
title('Contour Plot of using xguess = [3.5;3.7]');
hold on
%
% J11 = -1;
% J21 = 1;
% J12 = 3*(-x_1.^2 + 4)./(x_1^.2 + 4).^2;
% J22 = -0.5;
%
% J = [J11 J21;
%     J12 J22];
% figure;
x0 = [3.5;3.7];

xplotint = plot(3.5,3.7,'.','MarkerSize',20);
x_k = x0;
xt = x0;

hold on
for i=1:1:5
    fk = [-x_k(1) + x_k(2); 3*x_k(1)^2/(x_k(1)^2+4) - 3*x_k(2)];

    J11 = -1;
    J12 = 1;
    J21a = (24 * x_k(1));
    J21b = ((x_k(1)^2) + 4)^2;
    J21 = J21a / J21b;
    J22 = -0.5;
    J = [J11 J12;
```

```
        J21 J22];

    x_k = x_k - J\fk;
    x_k = x_k/1.0e+03; %my values are very very high, used this to decrease
    xt = [xt x_k];
    xplot_k = plot(x_k(1), x_k(2), '.', 'MarkerSize',20);
end

lengdv =  ['f1'; 'f2'; 'x0'; 'x1'; 'x2'; 'x3'; 'x4'; 'x5'; 'x6'];
legend(lengdv)




%I also tried to do the example problem from the overview file
%Regardless if I did it by hand or use the same method as above,
%I could not get the correct answers at all
%I saw no convergence, and my values were huge
% xk = [3;1]
% for i=1:1:5
%     fk = [2*(xk(1) - xk(1)*xk(2) + xk(2)^2)-4;  2*(xk(1) + xk(1)*xk(2) + xk(2)^2)-4];
%
%     J11 = 4*xk(1)+2*(xk(2));
%     J12 = 2*xk(1) + 4*(xk(2));
%     J21 = 4*xk(1)-2*(xk(2));
%     J22 = -2*xk(1) + 4*xk(2);
%     J = [J11 J12;
%         J21 J22];
%
%     xk = xk(1) - inv(J) * fk
%
%
% end
```

```
Warning: Ignoring extra legend entries.
```

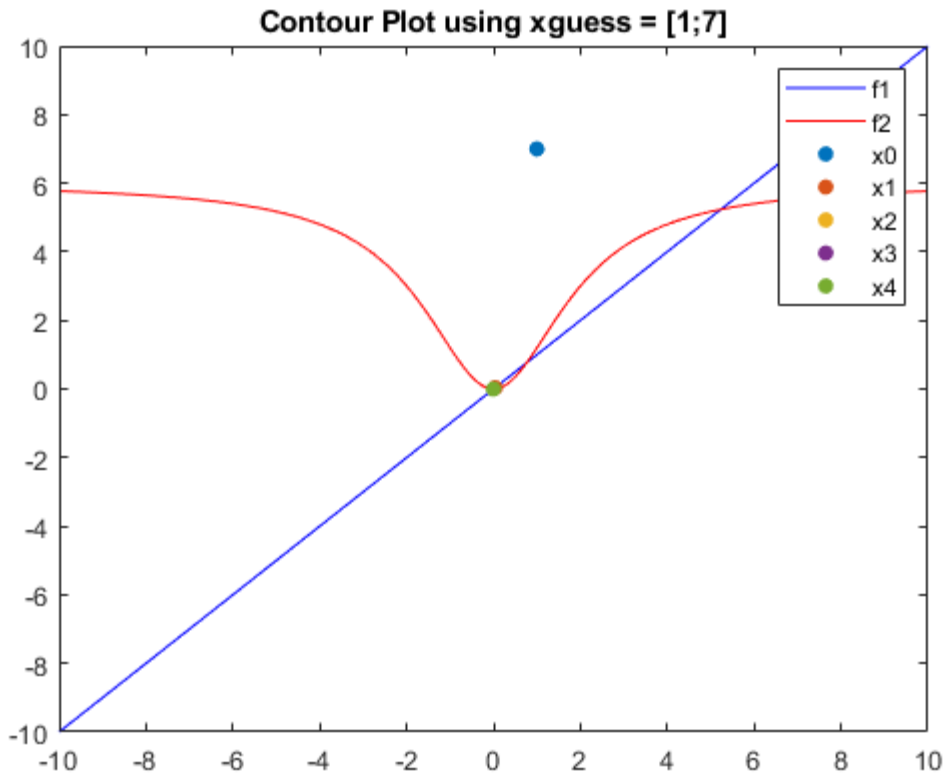


Part 3

```
a = 1;
b = 0.5;
c = 3;
d = 4;

x1 = -10:0.1:10;
x2 = -10:0.1:10;
[X1, X2] = meshgrid(x1,x2);

f1 = -a.*X1 + X2;
f2 = ((c.*(X1.^2))./(X1.^2 + d)) - b.*X2;
figure;
[c1 c2] = contour(X1,X2,f1,[0 0], 'Color', 'b');
% c2.levels
hold on
[c3 c4] = contour(X1,X2,f2,[0 0], 'Color', 'r');
legend('f1', 'f2');
title('Contour Plot using xguess = [1;7]');
hold on

x0 = [1;7];

xplotint = plot(1,7,'.','MarkerSize',20);
x_k = x0;


hold on
n=0;
err = 1;
xt2 = x0;
while err>1e-4
```

```
    fk = [-x_k(1) + x_k(2); 3*x_k(1)^2/(x_k(1)^2+4) - 3*x_k(2)];
    err=abs(fk(1))+abs(fk(2));
    J11 = -1;
    J12 = 1;
    J21a = (24 * x_k(1));
    J21b = ((x_k(1)^2) + 4)^2;
    J21 = J21a / J21b;
    J22 = -0.5;
    J = [J11 J12;
        J21 J22];

    x_k = x_k - J\fk;
    x_k = x_k/1.0e+03;
    xt2 = [xt2 x_k];
    n = n+1;
    xplot_k = plot(x_k(1), x_k(2), '.', 'MarkerSize',20);
end
lengdv =  ['f1'; 'f2'; 'x0'; 'x1'; 'x2'; 'x3'; 'x4'; 'x5'; 'x6'];
legend(lengdv)
```

```
Warning: Ignoring extra legend entries.
```



Echoing and Returning Results

```
diary vjprob3.txt
echo on
disp('Part 2')
disp('Iteration using guess [3.5;3.7]');
xt
disp('Considering we are using tangents, we should expect the behavior for the first guess to be far
```

```
and as the overall tangent "size" decreases, the answer iterate closer to true values.');

disp('Part 3')
disp('Iteration using guess [1;7]')
xt2

echo off
```

```
disp('Part 2')
Part 2
disp('Iteration using guess [3.5;3.7]');
Iteration using guess [3.5;3.7]
xt

xt =

    3.5000   -0.0445    0.0002   -0.0000    0.0000   -0.0000
    3.7000   -0.0445    0.0002   -0.0000    0.0000   -0.0000

disp('Considering we are using tangents, we should expect the behavior for the first guess to be far
and as the overall tangent "size" decreases, the answer iterate closer to true values.');
Considering we are using tangents, we should expect the behavior for the first guess to be far and
as the overall tangent "size" decreases, the answer iterate closer to true values.

disp('Part 3')
Part 3
disp('Iteration using guess [1;7]')
Iteration using guess [1;7]
xt2

xt2 =

    1.0000    0.0388   -0.0002    0.0000   -0.0000
    7.0000    0.0388   -0.0002    0.0000   -0.0000


echo off
```

Published with MATLAB® R2018b

Table of Contents


```
close all
clear all
```

Problem 4 Part 1

```
%load images and change black and white tone
%want white -127.5 black 127.5 from 255 0
%when A = 0, -(0-127.5) = 127.5
%when A = 255, -(255-127.5) = -127.5
nbits = 8;
A1 = imread('StevenTyler_BW_cropped(1)(1).tif');
A1 = single(A1);
A1 = -(A1 - (2^nbits-1)/2);
A2 = imread('MelissaRivers_BW_cropped(1)(1).tif');
A2 = single(A2);
A2 = -(A2 - (2^nbits-1)/2);
A3 = imread('LivTyler_BW_cropped(1)(1).tif');
A3 = single(A3);
A3 = -(A3 - (2^nbits-1)/2);
A4 = imread('KieferSutherland_BW_cropped(1)(1).tif');
A4 = single(A4);
A4 = -(A4 - (2^nbits-1)/2);
A5 = imread('RachelSutherland_BW_cropped(1)(1).tif');
A5 = single(A5);
A5 = -(A5 - (2^nbits-1)/2);
A6 = imread('EmilyDeschanel_BW_cropped(1)(1).tif');
A6 = single(A6);
A6 = -(A6 - (2^nbits-1)/2);
A7 = imread('ZooeyDeschanel_BW_cropped(1)(1).tif');
A7 = single(A7);
A7 = -(A7 - (2^nbits-1)/2);
A8 = imread('ZooeyDeschanel2_BW_cropped(1)(1).tif');
A8 = single(A8);
A8 = -(A8 - (2^nbits-1)/2);
A9 = imread('KatyPerry_BW_cropped(1)(1).tif');
A9 = single(A9);
A9 = -(A9 - (2^nbits-1)/2);

% sanity check
% imshow(mat2gray(A2))
```

Part 2

```
A1 = A1';
x1 = A1(:);
A2 = A2';
x2 = A2(:);
A3 = A3';
x3 = A3(:);
A4 = A4';
x4 = A4(:);
A5 = A5';
x5 = A5(:);
A6 = A6';
x6 = A6(:);
A7 = A7';
x7 = A7(:);
A8 = A8';
x8 = A8(:);
A9 = A9';
x9 = A9(:);

X = [x1 x2 x3 x4 x5 x6 x7 x8 x9];
```

Part 3 and 4

```
%deveiation matrix for original X
xmean = [];
for i=1:9
    xmean(i) = mean(X(:,i));
end

xm_m = repmat(xmean, length(x1), 1);

D = X - xm_m;

S = 1/(length(x1)-1) * D' * D;

[eigVec,eigVal] = eig(S); %echo these

mineigVec = min(eigVec);

%normalized by x1/min(x1)
V_norm = eigVec./mineigVec;

%Pearson Correlation

for i=1:9
    Dnorm(:,i) = D(:,i)./norm(D(:,i));
end

R = Dnorm'*Dnorm;
```

```
% Rllone = R(find(R < 1,5,'last'));
diary vjprob4.txt
echo on
disp('2nd max is 0.8335, index at 7,8 and 8, this is because both
 those images are of the same person Zooey Deschanel')
```

```
disp('2nd max is 0.8335, index at 7,8 and 8, this is because both
 those images are of the same person Zooey Deschanel')
2nd max is 0.8335, index at 7,8 and 8, this is because both those
 images are of the same person Zooey Deschanel
```

Part 5

```
disp('The first three PCs can explain')

TotVar = (eigVal(1,1) + eigVal(2,2) + eigVal(3,3))/(sum(sum(eigVal)))
 *100

disp('percent of the total variance')


disp('In the second PC eigenvector and for group 1 photos, the entries
 are most influenced by mouth shape.')
disp('For Group 2 photos, the last 3 entries all have the same sign
 and relatively large magnitudes as the most prominent features
 recongized are possibly basic facial shape.')
disp('For Group 3 photos those are images of the same person at
 different angles, so the features are largely the same.')
disp('The oddball entry seems to be photo 4, as the abs(value) is the
 smallest.')
disp('In the third PC eigenvector those psoitions (1,2.6) all
 correspond to people who are smiling wide showing their teeth. These
 reveal a nearly max intensity white.')
disp('The nearly max instensity present will essentially act as a
 persisten max value through all our manipulations.')


%echoing all other values needed

disp('Part 4 Values')
eigVal
eigVec
V_norm

disp('Part 3 Values')
S
R
echo off


%% Part 5
disp('The first three PCs can explain')
The first three PCs can explain
```

```
TotVar = (eigVal(1,1) + eigVal(2,2) + eigVal(3,3))/(sum(sum(eigVal)))
 *100

TotVar =

  single

   77.2885


disp('percent of the total variance')
percent of the total variance


disp('In the second PC eigenvector and for group 1 photos, the entries
 are most influenced by mouth shape.')
In the second PC eigenvector and for group 1 photos, the entries are
 most influenced by mouth shape.
disp('For Group 2 photos, the last 3 entries all have the same sign
 and relatively large magnitudes as the most prominent features
 recongized are possibly basic facial shape.')
For Group 2 photos, the last 3 entries all have the same sign and
 relatively large magnitudes as the most prominent features recongized
 are possibly basic facial shape.
disp('For Group 3 photos those are images of the same person at
 different angles, so the features are largely the same.')
For Group 3 photos those are images of the same person at different
 angles, so the features are largely the same.
disp('The oddball entry seems to be photo 4, as the abs(value) is the
 smallest.')
The oddball entry seems to be photo 4, as the abs(value) is the
 smallest.
disp('In the third PC eigenvector those psoitions (1,2.6) all
 correspond to people who are smiling wide showing their teeth. These
 reveal a nearly max intensity white.')
In the third PC eigenvector those psoitions (1,2.6) all correspond
 to people who are smiling wide showing their teeth. These reveal a
 nearly max intensity white.
disp('The nearly max instensity present will essentially act as a
 persisten max value through all our manipulations.')
The nearly max instensity present will essentially act as a persisten
 max value through all our manipulations.


%echoing all other values needed

disp('Part 4 Values')
Part 4 Values
eigVal

eigVal =

  9×9 single matrix
```

```
 1.0e+04 *

  Columns 1 through 7

    1.8706         0         0         0         0         0         0
         0    0.4649         0         0         0         0         0
         0         0    0.4069         0         0         0         0
         0         0         0    0.2418         0         0         0
         0         0         0         0    0.0674         0         0
         0         0         0         0         0    0.0736         0
         0         0         0         0         0         0    0.1685
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0

  Columns 8 through 9

         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
    0.1234         0
         0    0.1311

eigVec

eigVec =

  9×9 single matrix

  Columns 1 through 7

    0.1684    0.6266   -0.4424    0.2087    0.0736    0.1279    0.1221
    0.2011    0.3782    0.6078   -0.5394   -0.0135    0.1064    0.2342
    0.3213    0.4964   -0.1796   -0.1047    0.0079   -0.0053   -0.3295
    0.1031    0.0275    0.1659    0.6053   -0.1171    0.4160    0.4302
    0.1590    0.1282    0.2538    0.3810    0.2827   -0.7928    0.0151
    0.3944   -0.0560    0.4540    0.3302   -0.1951    0.2051   -0.4879
    0.4900   -0.3555   -0.1909   -0.1093    0.5293    0.2243   -0.2283
    0.3828   -0.1655   -0.2502   -0.1018   -0.7534   -0.2627    0.0043
    0.4999   -0.2092   -0.0744   -0.1075    0.1219   -0.0966    0.5884

  Columns 8 through 9

    0.5489    0.0407
    0.1459   -0.2618
   -0.7022    0.0740
   -0.3100   -0.3560
    0.0228   -0.1988
    0.2414    0.3869
    0.1032   -0.4381
    0.0943   -0.3278
```

```
   -0.0995    0.5558

V_norm

V_norm =

  9×9 single matrix

  Columns 1 through 7

    1.6335   -1.7627    1.0000   -0.3869   -0.0977   -0.1614   -0.2502
    1.9506   -1.0639   -1.3737    1.0000    0.0179   -0.1342   -0.4801
    3.1168   -1.3962    0.4060    0.1941   -0.0105    0.0067    0.6753
    1.0000   -0.0773   -0.3750   -1.1223    0.1554   -0.5247   -0.8818
    1.5422   -0.3607   -0.5737   -0.7064   -0.3752    1.0000   -0.0310
    3.8267    0.1575   -1.0260   -0.6121    0.2589   -0.2588    1.0000
    4.7540    1.0000    0.4314    0.2026   -0.7025   -0.2829    0.4679
    3.7143    0.4656    0.5655    0.1888    1.0000    0.3314   -0.0088
    4.8495    0.5884    0.1681    0.1992   -0.1618    0.1219   -1.2060

  Columns 8 through 9

   -0.7817   -0.0929
   -0.2077    0.5976
    1.0000   -0.1689
    0.4414    0.8127
   -0.0325    0.4538
   -0.3438   -0.8833
   -0.1470    1.0000
   -0.1342    0.7482
    0.1417   -1.2688


disp('Part 3 Values')
Part 3 Values
S

S =

  9×9 single matrix

   1.0e+03 *

  Columns 1 through 7

    3.6726    0.5110    2.1887    0.3043    0.5569    0.5221    0.8432
    0.5110    3.8449    1.4910    0.3265    0.9678    1.8129    0.9799
    2.1887    1.4910    4.0326    0.4014    0.9264    1.9229    2.2882
    0.3043    0.3265    0.4014    1.9335    0.8820    0.9947    0.6370
    0.5569    0.9678    0.9264    0.8820    1.7316    1.6492    1.0286
    0.5221    1.8129    1.9229    0.9947    1.6492    4.7529    3.2284
    0.8432    0.9799    2.2882    0.6370    1.0286    3.2284    5.8348
    1.1077    0.7803    2.0085    0.4982    0.7853    2.2420    3.8905
    1.1248    1.4830    2.4174    0.8953    1.1327    3.2575    4.4829
```

```
  Columns 8 through 9

    1.1077    1.1248
    0.7803    1.4830
    2.0085    2.4174
    0.4982    0.8953
    0.7853    1.1327
    2.2420    3.2575
    3.8905    4.4829
    3.7344    3.5533
    3.5533    5.9452

R

R =

  9×9 single matrix

  Columns 1 through 7

    1.0000    0.1360    0.5687    0.1142    0.2208    0.1250    0.1822
    0.1360    1.0000    0.3787    0.1197    0.3751    0.4241    0.2069
    0.5687    0.3787    1.0000    0.1438    0.3506    0.4392    0.4717
    0.1142    0.1197    0.1438    1.0000    0.4820    0.3281    0.1897
    0.2208    0.3751    0.3506    0.4820    1.0000    0.5749    0.3236
    0.1250    0.4241    0.4392    0.3281    0.5749    1.0000    0.6131
    0.1822    0.2069    0.4717    0.1897    0.3236    0.6131    1.0000
    0.2991    0.2059    0.5176    0.1854    0.3088    0.5322    0.8335
    0.2407    0.3102    0.4937    0.2641    0.3530    0.6128    0.7611

  Columns 8 through 9

    0.2991    0.2407
    0.2059    0.3102
    0.5176    0.4937
    0.1854    0.2641
    0.3088    0.3530
    0.5322    0.6128
    0.8335    0.7611
    1.0000    0.7541
    0.7541    1.0000

echo off
```

Published with MATLAB® R2018b