Machine Learning Assignment # 03

Name: Muhammad Gul

Reg # 186383

Each matlab code should be run in separate windows and each piece of code also displays the corresponding required plots.

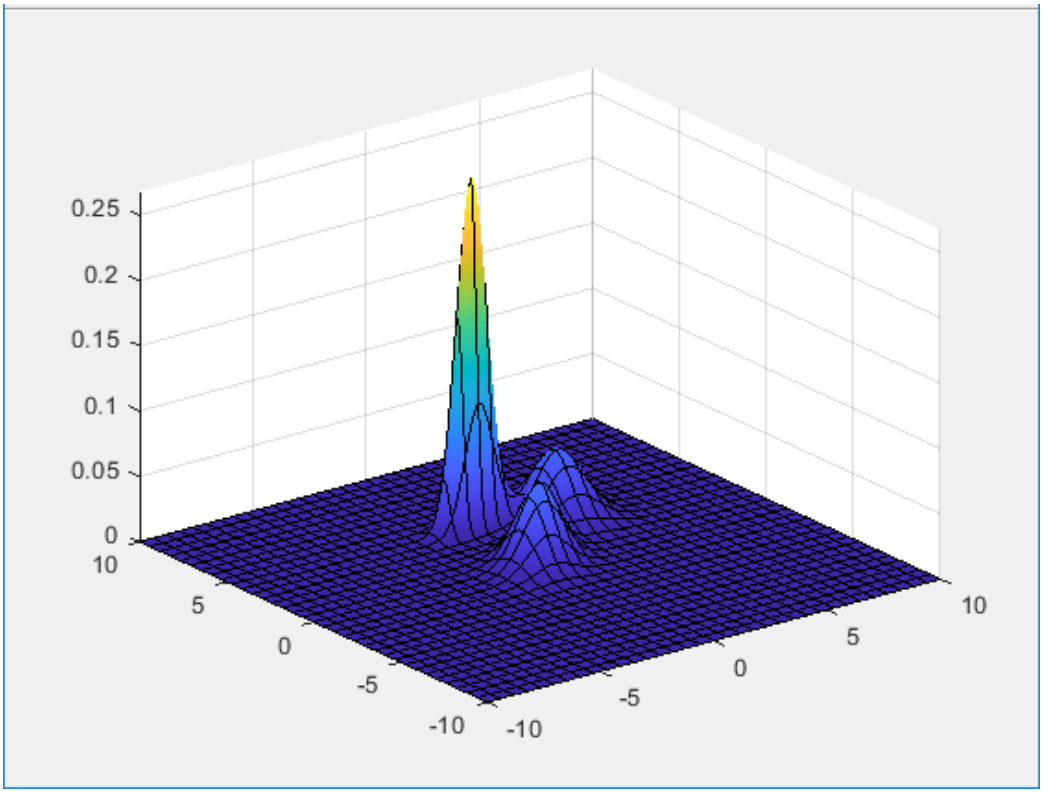
Task 1:

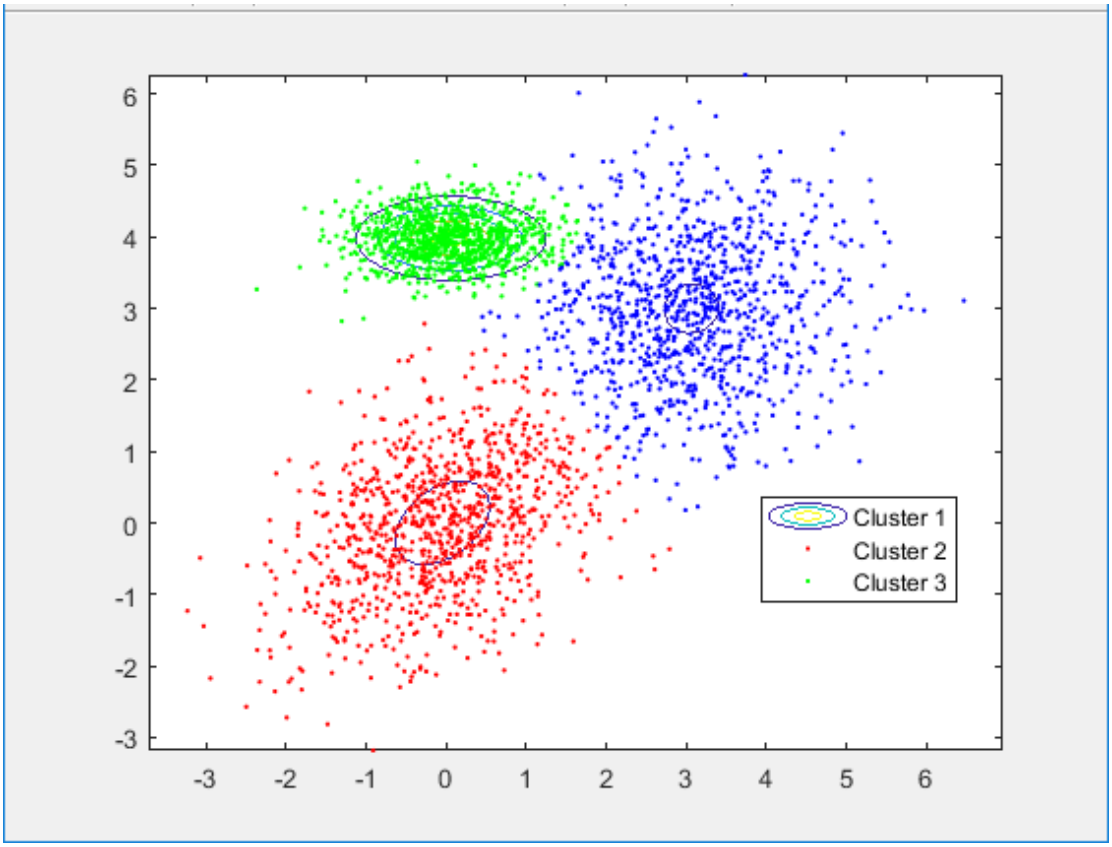
A 3D plot of GMMs for the given set of parameters:

Code is Below:

```
%---------- Task 1 ----------%
%Defining the distribution parameters (means and covariances) of
%three bivariate Gaussian mixture components individually.
mu1 = [0 0];         % Mean of the 1st component
sigma1 = [1 0.4;0.4 1];  % Covariance of the 1st component
mu2 = [3 3];       % Mean of the 2nd component
sigma2 = [1 0; 0 1];  % Covariance of the 2nd component
mu3 = [0 4];       % Mean of the 3rd component
sigma3=[0.4 0; 0 0.1]; % Covariance of the 3rd component
%--- Combined arrays of Means and Cov
mu = [0 0;3 3;0 4];          % Means
sigma = cat(3,sigma1,sigma2,sigma3); %Covariances
%Here goes a 3D surf plot of the corresponding "mu" and "sigma" arrays of means and cov
gm_dist = gmdistribution(mu,sigma);
figure
fsurf(@(x,y)reshape(pdf(gm_dist,[x(:),y(:)]),size(x)),[-10 10]);
% Here an equal number of random variates from each component are generated,
% and then combined into three sets of random variates.
rng('default') % For reproducibility
```

```
r1 = mvnrnd(mu1,sigma1,1000);

r2 = mvnrnd(mu2,sigma2,1000);

r3 = mvnrnd(mu3,sigma3,1000);

X = [r1; r2; r3];

%The combined data set X contains random variates following a mixture of three bivariate Gaussian distribution.

%Fitting a three-component GMM to X.

gm_Plot_Model = fitgmdist(X,3);

%Ploting X by using scatter and to Visualize the fitted model gm by using pdf and fcontour.

figure

% scatter(X(:,1),X(:,2),10,'.') % Scatter plot with points of size 10

% hold on

gmPDF = @(x,y)reshape(pdf(gm_Plot_Model,[x(:),y(:)]),size(x));

fcontour(gmPDF,[-2 6 -2 6])

hold on

%Partition of the data into clusters by passing the fitted GMM and the data to cluster.

idx = cluster(gm_Plot_Model,X);

%Using gscatter to create a scatter plot grouped by idx.

gscatter(X(:,1),X(:,2),idx);

legend('Cluster 1','Cluster 2','Cluster 3','Location','best');
```


0.25
0.2
0.15
0.1
0.05
0
10
5
0
-5
-10
-10
-5
0
5
10


6
5
4
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
4
5
6
Cluster 1
Cluster 2
Cluster 3

```
mu =

     0     0
     3     3
     0     4

>> sigma

sigma(:,:,1) =

    1.0000    0.4000
    0.4000    1.0000


sigma(:,:,2) =

     1     0
     0     1


sigma(:,:,3) =

    0.4000         0
         0    0.1000
```

Given set of Parameters

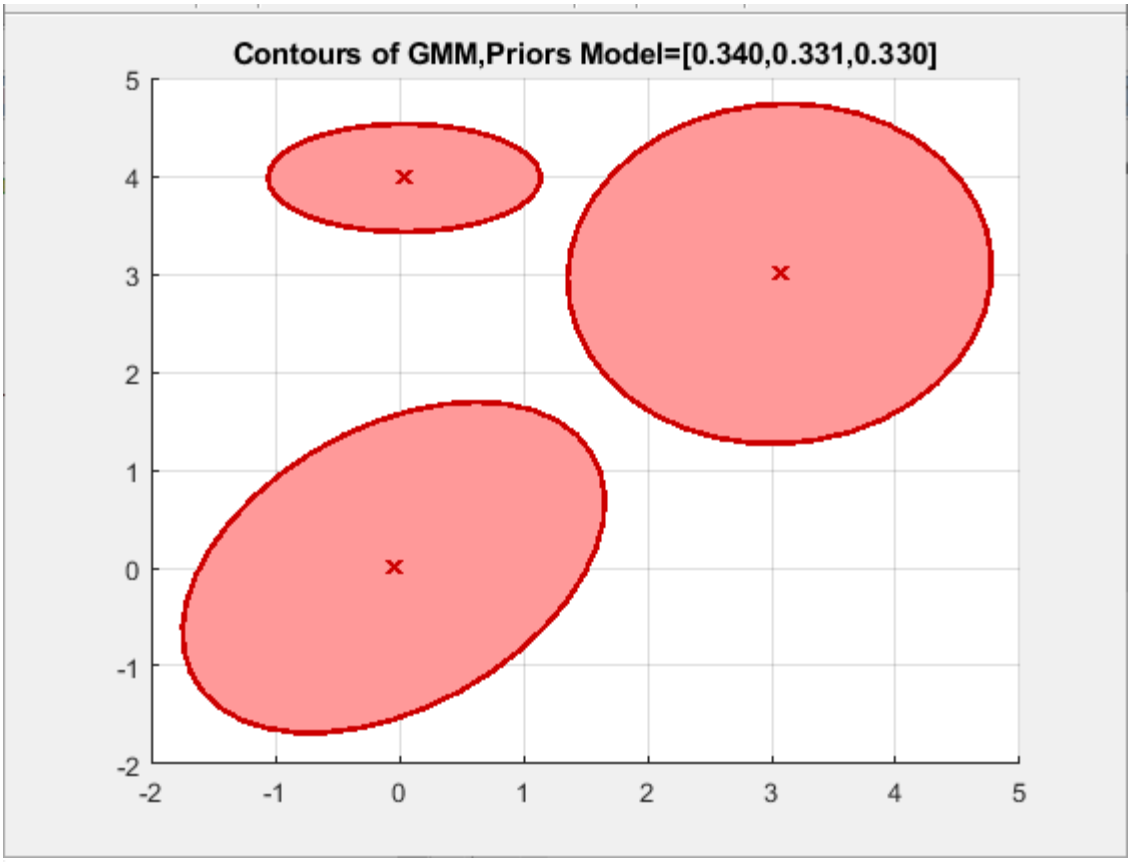
Task 2 to 4:

Trying to estimate parameters of GMMs using EM for known value of k=3;

```
delete(gcp('nocreate')); %Disable all active PC workers
Mu_given = [0 0;3 3;0 4];          % Means
Sigma_given = cat(3,[1 0.4;0.4 1],[1 0; 0 1],[0.4 0; 0 0.1]);  % Covariances 1-by-2by-2 array
Mdl = gmdistribution(Mu_given,Sigma_given); % MOdel
rng(1); % For reproducibility
X = random(Mdl,1000);
%Mdl is a 2-dimensional gmdistribution model with 3 components.
%X is a 10000-by-2 matrix of data generated from Mdl.
%Invoke a parallel pool of workers. Specify options for parallel computing.
pool = parpool;                    % Invokes workers
```

```
stream = RandStream('mlfg6331_64');  % Random number stream
options = statset('UseParallel',1,'UseSubstreams',1,...
   'Streams',stream);
%[Priors, Mu, Sigma] = EM_init_kmenas(X',3);
[idx,C] = kmeans(X,3);
%plot results
figure;
plot(X(idx==1,1),X(idx==1,2),'r.','MarkerSize',12)
hold on
plot(X(idx==2,1),X(idx==2,2),'g.','MarkerSize',12)
hold on
plot(X(idx==3,1),X(idx==3,2),'b.','MarkerSize',12)
plot(C(:,1),C(:,2),'kx',...
    'MarkerSize',12,'LineWidth',3)
legend('Cluster 1','Cluster 2','Cluster 3','Centroids',...
      'Location','NW')
title 'Cluster Assignments and Centroids'
hold off
[nbVar, nbData] = size(X);
Mu0 = C';
for i=1:3
  idtmp = find(idx==i);
  Priors0(i) = length(idtmp);
  Sigma0(:,:,i) = cov(X, X');
end
Priors0 = Priors0 ./ sum(Priors0);
[Priors_Init, Mu_Init, Sig_Init, Pix_Init]= EM(X', Priors0, Mu0, Sigma0);
s = struct('mu',Mu_Init','Sigma',Sig_Init,'PComponents',Priors_Init);
options1 = statset('Display','final');
```

```
e = 1e-5;
GMModel =gmdistribution.fit(X,3,'CovType','full','Options',options1,'Start',s,'Regularize',e);
muModel = GMModel.mu;
SigModel = GMModel.Sigma;
figure;
hold on; grid on;
Wmodel = GMModel.ComponentProportion;
plotGMM(muModel', SigModel, [.8 0 0], 1);
title(sprintf('Contours of GMM, Wmodel = [%0.3f, %0.3f,%0.3f]',sort(Wmodel,'descend')));
gm_dist_x = gmdistribution(muModel,SigModel);
figure
fsurf(@(x,y)reshape(pdf(gm_dist_x,[x(:),y(:)]),size(x)),[-10 10]);
```




```
BestModel =

Gaussian mixture distribution with 3 components in 2 dimensions
Component 1:
Mixing proportion: 0.342883
Mean:    3.0036    2.9734

Component 2:
Mixing proportion: 0.321382
Mean:   -0.0998   -0.0418

Component 3:
Mixing proportion: 0.335735
Mean:    0.0312    3.9824
```

Estimated Parameters

Task 5:

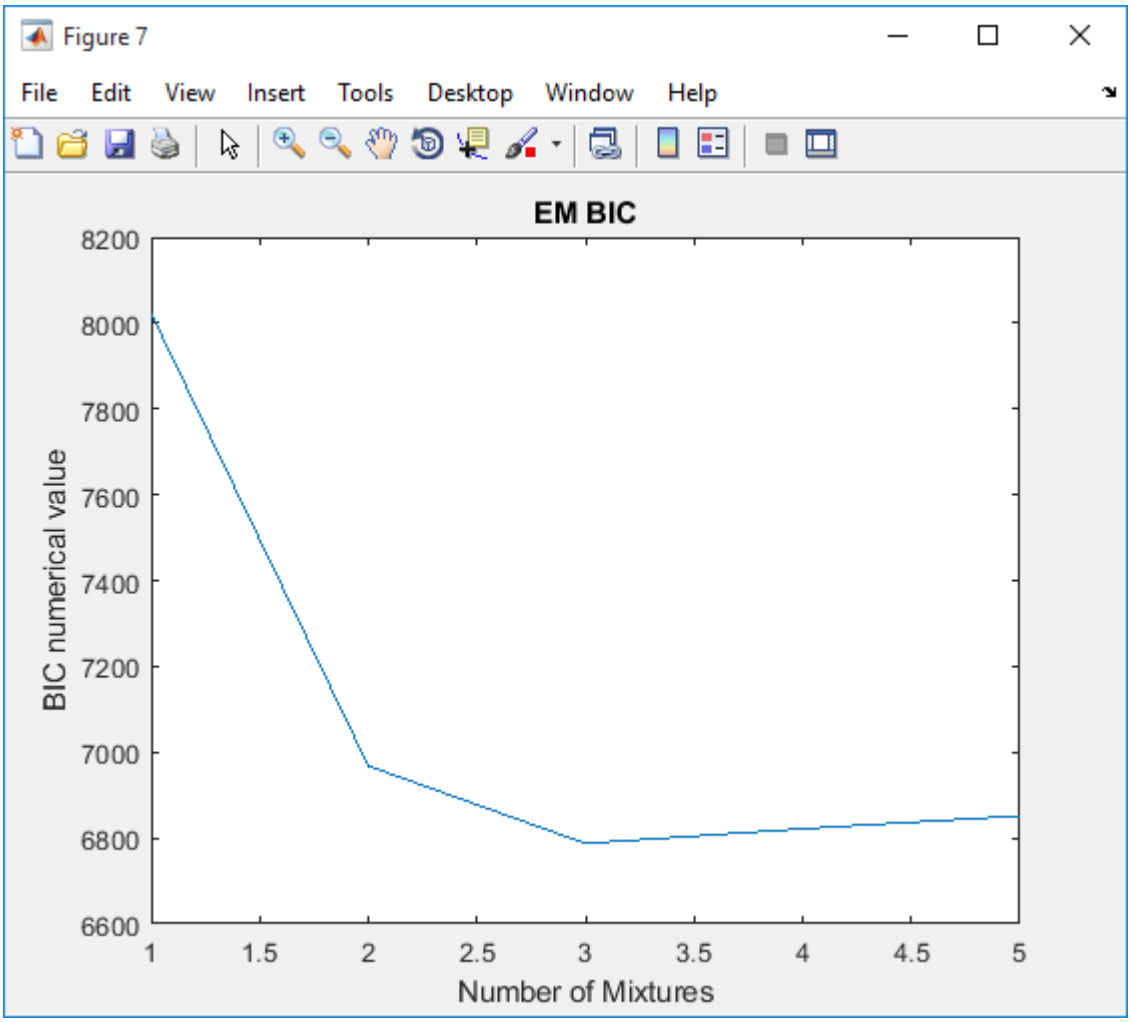
Trying EM for unknown value of k=Number of Mixtures

But we check BIC for k in range of 1 to 5.

```
delete(gcp('nocreate')); %Disable all active PC workers
Mu_given = [0 0;3 3;0 4];           % Means
Sigma_given = cat(3,[1 0.4;0.4 1],[1 0; 0 1],[0.4 0; 0 0.1]);  % Covariances 1-by-2by-2 array
Mdl = gmdistribution(Mu_given,Sigma_given); % MOdel
rng(1); % For reproducibility
X = random(Mdl,1000);
plot(X(:,1),X(:,2),'go','MarkerSize',2);
%X is a 4999-by-2 matrix of data loaded from Lab-07.mat
NumMix_Upper_Limit=5;
Bic_model = zeros(1,NumMix_Upper_Limit);
NumMix=1;
for Test=1:NumMix_Upper_Limit
delete(gcp('nocreate'));
%Invoke a parallel pool of workers. Specify options for parallel computing.
pool = parpool;                    % Invokes workers
stream = RandStream('mlfg6331_64');  % Random number stream
options = statset('UseParallel',1,'UseSubstreams',1,...
    'Streams',stream);
%[Priors, Mu, Sigma] = EM_init_kmenas(X',3);
[idx,C] = kmeans(X,NumMix);
[nbVar, nbData] = size(X);
Mu0 = C';
for i=1:NumMix
  idtmp = find(idx==i);
  Priors0(i) = length(idtmp);
  Sigma0(:,:,i) = cov(X, X');
end
Priors0 = Priors0 ./ sum(Priors0);
[Priors_Init, Mu_Init, Sig_Init, Pix_Init]= EM(X', Priors0, Mu0, Sigma0);
```

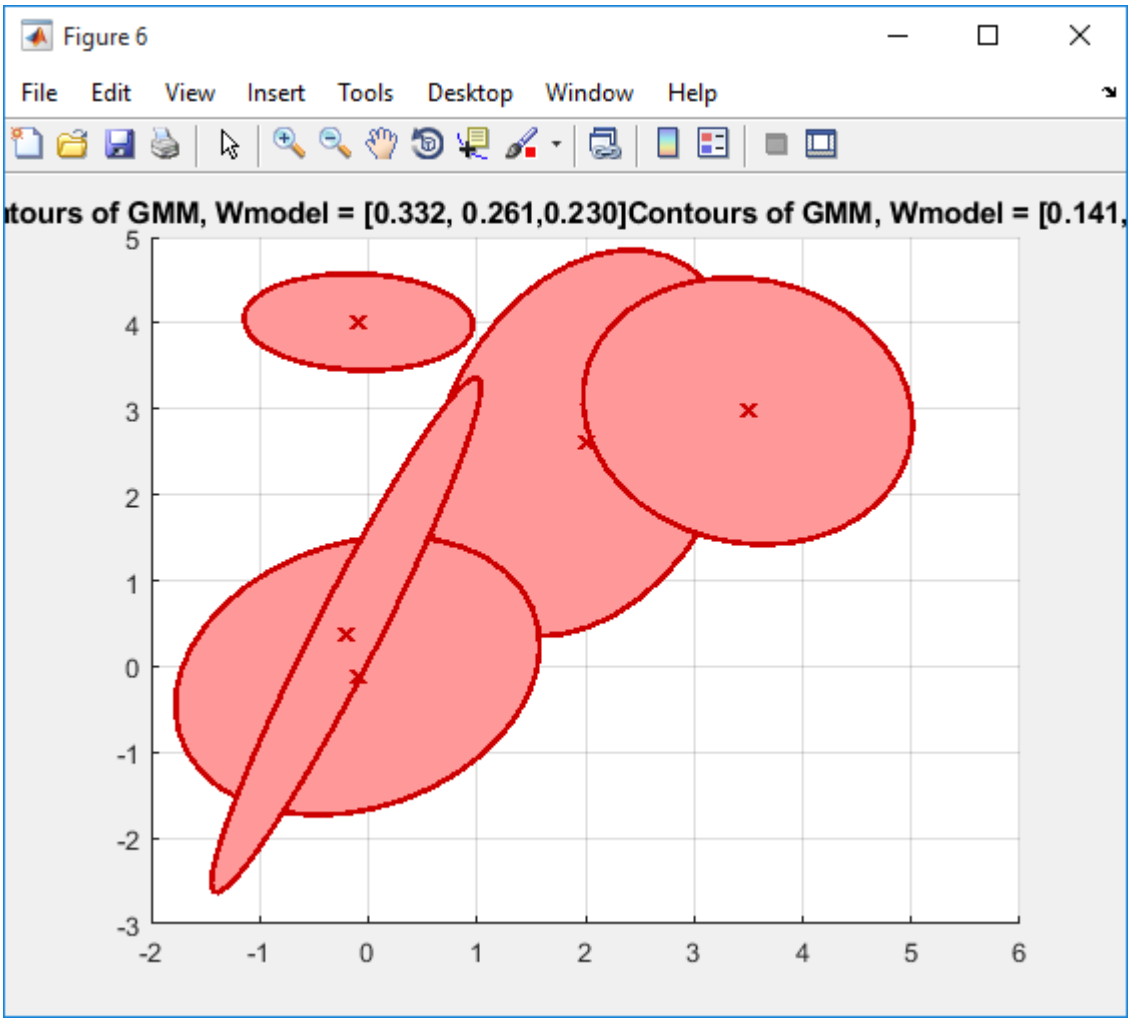
```
s = struct('mu',Mu_Init','Sigma',Sig_Init,'PComponents',Priors_Init);
options1 = statset('Display','final');
e = 1e-5;
GMModel =gmdistribution.fit(X,NumMix,'CovType','full','Options',options1,'Start',s,'Regularize',e);
muModel = GMModel.mu;
SigModel = GMModel.Sigma;
figure;
hold on; grid on;
Wmodel = GMModel.ComponentProportion;
plotGMM(muModel', SigModel, [.8 0 0], 1);
title(sprintf('Contours of GMM, Wmodel = [%0.3f, %0.3f,%0.3f]',sort(Wmodel,'descend')));
Bic_model(1,NumMix)=GMModel.BIC;
   if (NumMix==NumMix_Upper_Limit)
      %plot Bic vs Num of Mixtures
      NumMix2plot=1:5;
      figure
      plot(NumMix2plot,Bic_model)
      title('EM BIC')
      xlabel('Number of Mixtures')
      ylabel('BIC numerical value')
      break;
   end
   NumMix=NumMix+1;
End
```

BIC shows a best fit for k=3 as expected, the BIC graph is below



For k>3

EM results in over fitting, which is shown below:



Task 6 to 8:

Finding the Best Number of GMMs for the given data Using BIC

```
delete(gcp('nocreate')); %Disable all active PC workers
load ('Lab-07.mat')
plot(X(:,1),X(:,2),'go','MarkerSize',2);
%X is a 4999-by-2 matrix of data loaded from Lab-07.mat
NumMix_Upper_Limit=9;
Bic_model = zeros(1,NumMix_Upper_Limit);
NumMix=1;
for Test=1:NumMix_Upper_Limit
```

```
delete(gcp('nocreate'));
%Invoke a parallel pool of workers. Specify options for parallel computing.
pool = parpool;                    % Invokes workers
stream = RandStream('mlfg6331_64');  % Random number stream
options = statset('UseParallel',1,'UseSubstreams',1,...
    'Streams',stream);
%[Priors, Mu, Sigma] = EM_init_kmenas(X',3);
[idx,C] = kmeans(X,NumMix);
[nbVar, nbData] = size(X);
Mu0 = C';
for i=1:NumMix
  idtmp = find(idx==i);
  Priors0(i) = length(idtmp);
  Sigma0(:,:,i) = cov(X, X');
end
Priors0 = Priors0 ./ sum(Priors0);
[Priors_Init, Mu_Init, Sig_Init, Pix_Init]= EM(X', Priors0, Mu0, Sigma0);
s = struct('mu',Mu_Init','Sigma',Sig_Init,'PComponents',Priors_Init);
options1 = statset('Display','final');
e = 1e-5;
GMModel =gmdistribution.fit(X,NumMix,'CovType','full','Options',options1,'Start',s,'Regularize',e);
muModel = GMModel.mu;
SigModel = GMModel.Sigma;
figure;
hold on; grid on;
Wmodel = GMModel.ComponentProportion;
plotGMM(muModel', SigModel, [.8 0 0], 1);
title(sprintf('Contours of GMM, Wmodel = [%0.3f, %0.3f,%0.3f]',sort(Wmodel,'descend')));
Bic_model(1,NumMix)=GMModel.BIC;
```

```
    if (NumMix==NumMix_Upper_Limit)
        %plot Bic vs Num of Mixtures
        NumMix2plot=1:9;
        figure
        plot(NumMix2plot,Bic_model)
        title('EM BIC')
        xlabel('Number of Mixtures')
        ylabel('BIC numerical value')
        break;
    end
    NumMix=NumMix+1;
end
```

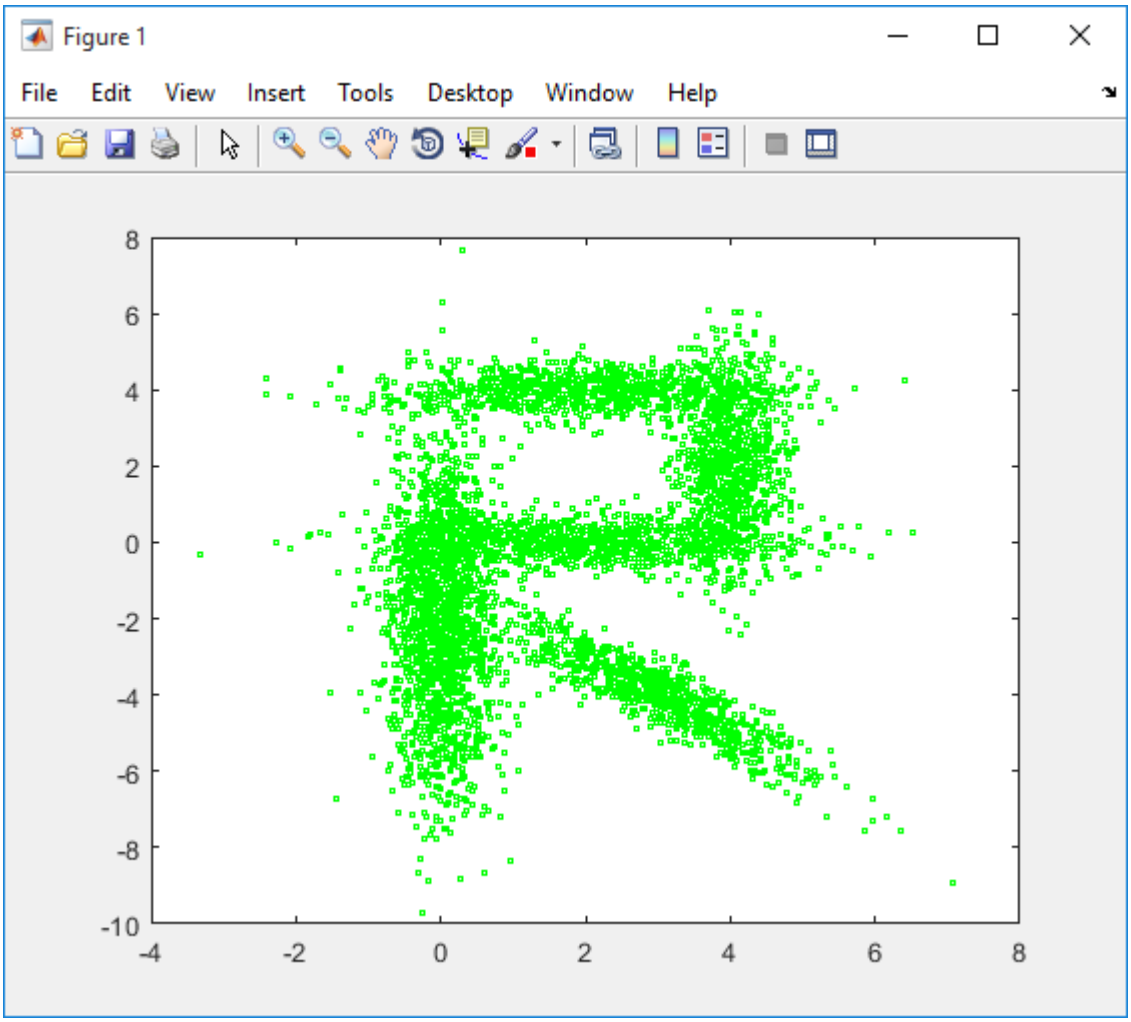
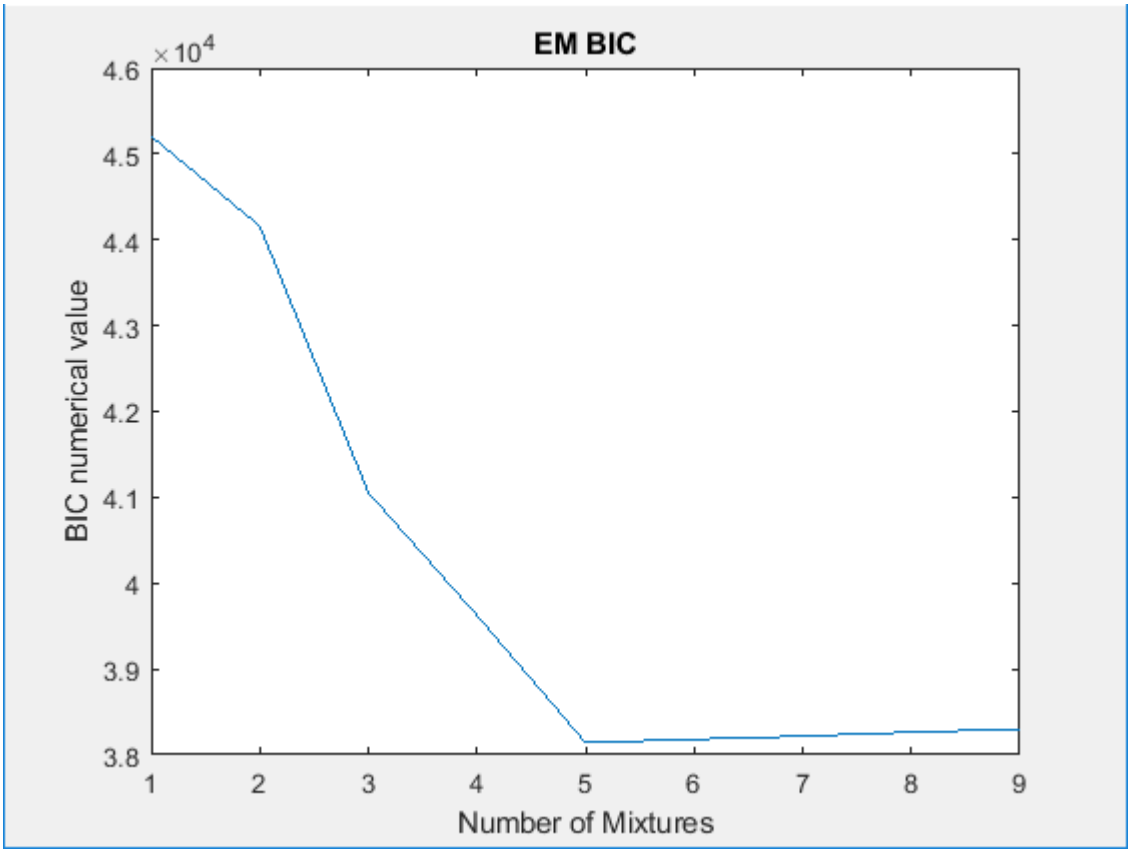


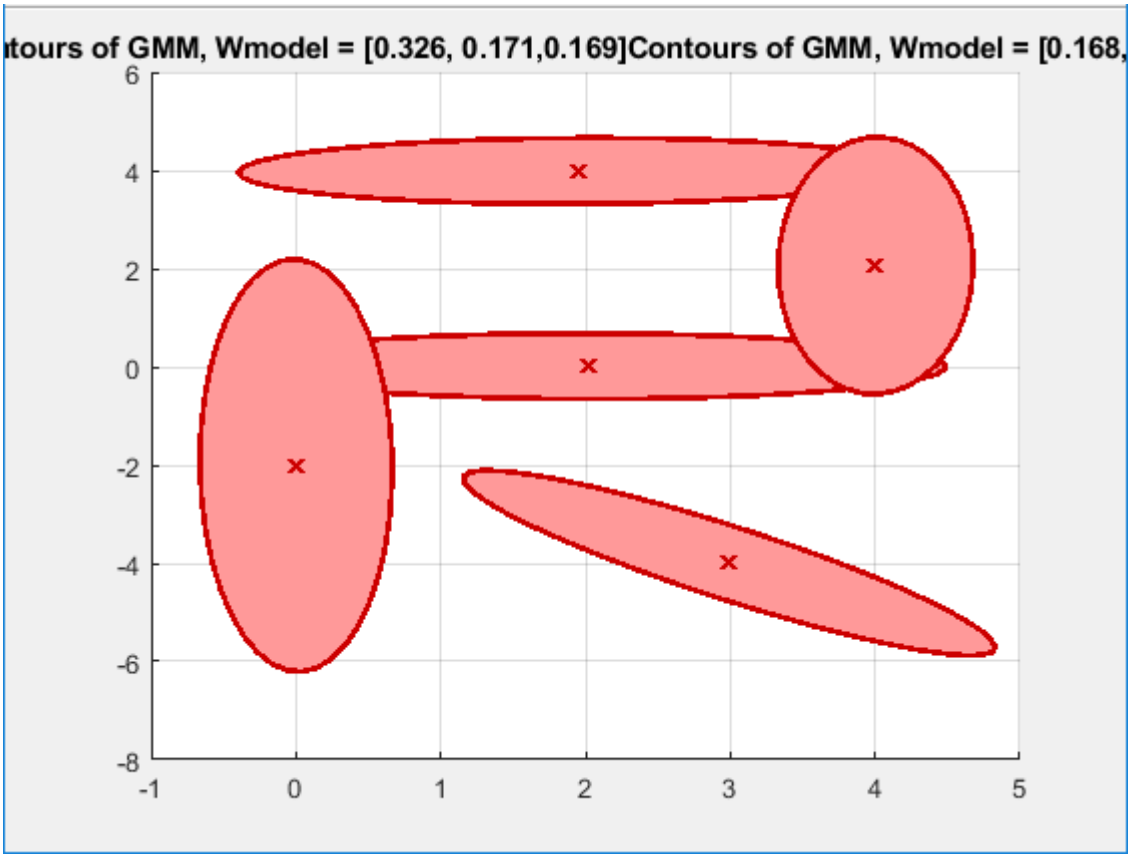
Figure 1
File Edit View Insert Tools Desktop Window Help
8
6
4
2
0
-2
-4
-6
-8
-10
-4
-2
0
2
4
6
8

Data plot:



This graph that the best fit for the data “Lab-07.mat” is k=5

The plot for k=5 is below;



For the cases:

K=1 to 4 (underfitting) and k=6 to 9 (Overfitting)

The corresponding graphs are below:

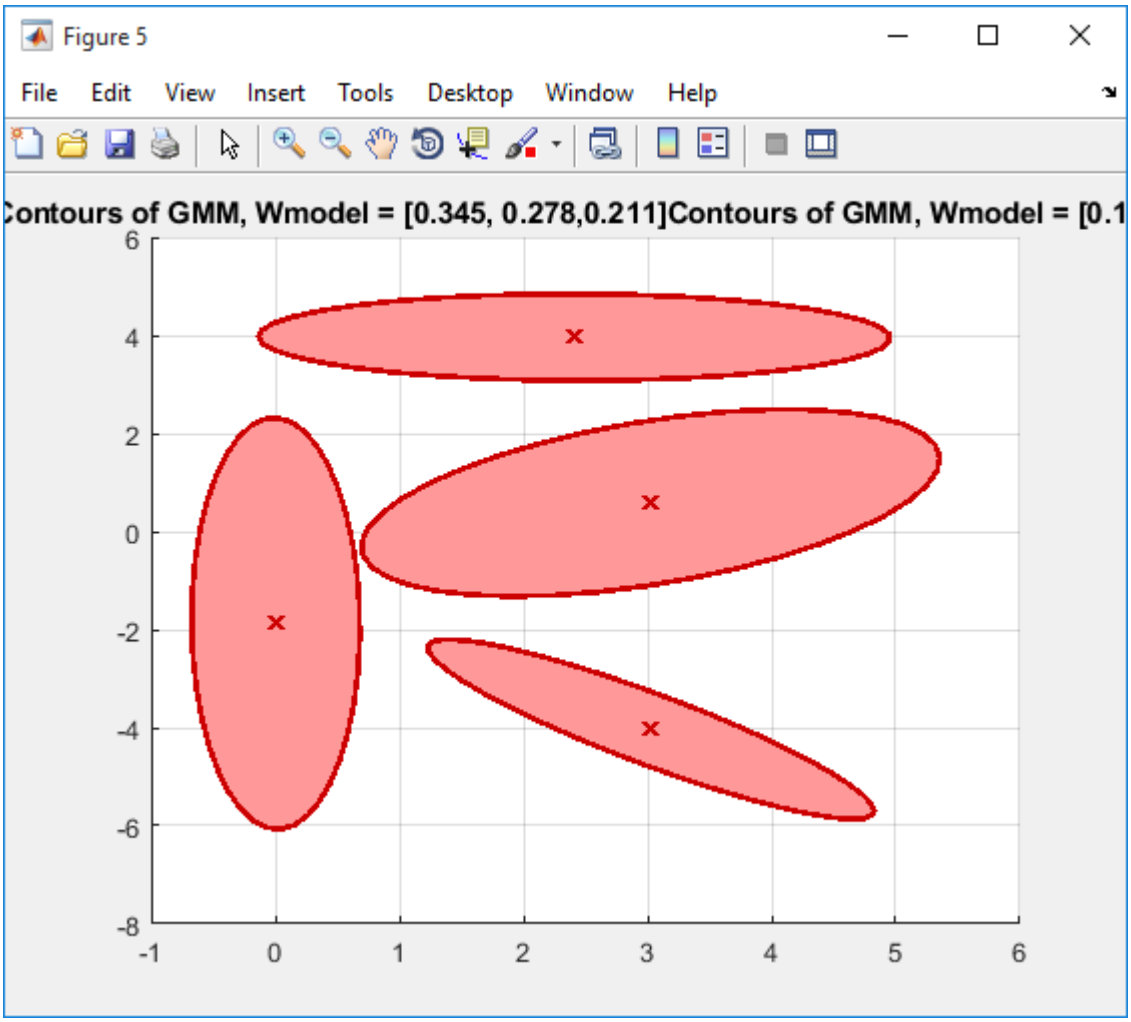
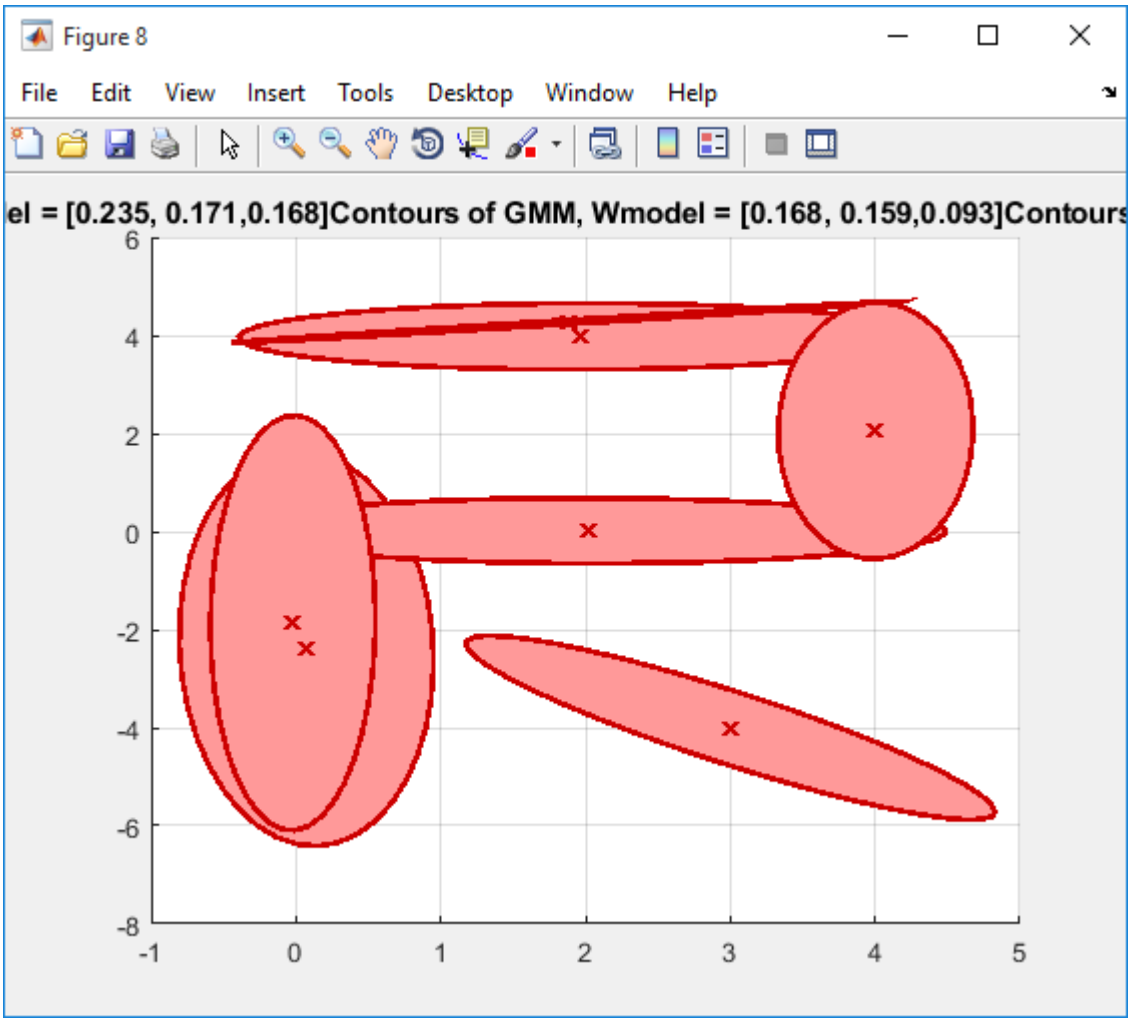
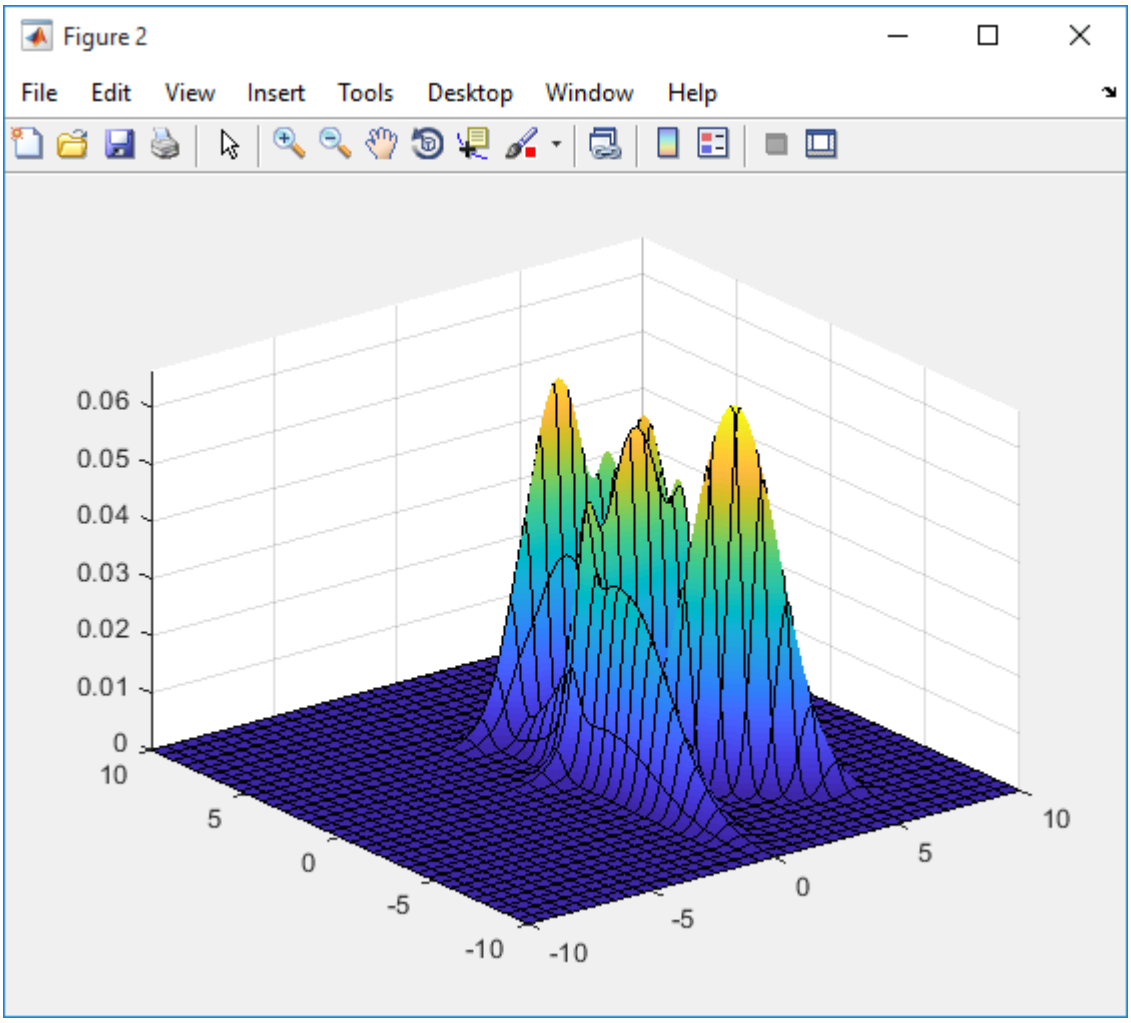


Figure 5
File Edit View Insert Tools Desktop Window Help
Contours of GMM, Wmodel = [0.345, 0.278,0.211]Contours of GMM, Wmodel = [0.1
6
4
2
0
-2
-4
-6
-8
-1
0
1
2
3
4
5
6



And finally the best Model 3d plot and estimated parameters are:



Weight and Mu matrix for the 5 components:

```
Wmodel =

    0.1711    0.3265    0.1651    0.1690    0.1683

>> GMModel.mu

ans =

    4.0033    2.0624
    0.0013   -2.0063
    1.9526    3.9887
    2.9917   -3.9931
    2.0197    0.0143
```

Sigma of Model estimated:

```
    0.1512    0.0100
    0.0100    2.2836


ans(:,:,2) =

    0.1472   -0.0214
   -0.0214    5.8846


ans(:,:,3) =

    1.8419    0.0255
    0.0255    0.1515


ans(:,:,4) =

    1.1247   -1.0511
   -1.0511    1.1853


ans(:,:,5) =

    2.0359   -0.0105
   -0.0105    0.1493
```