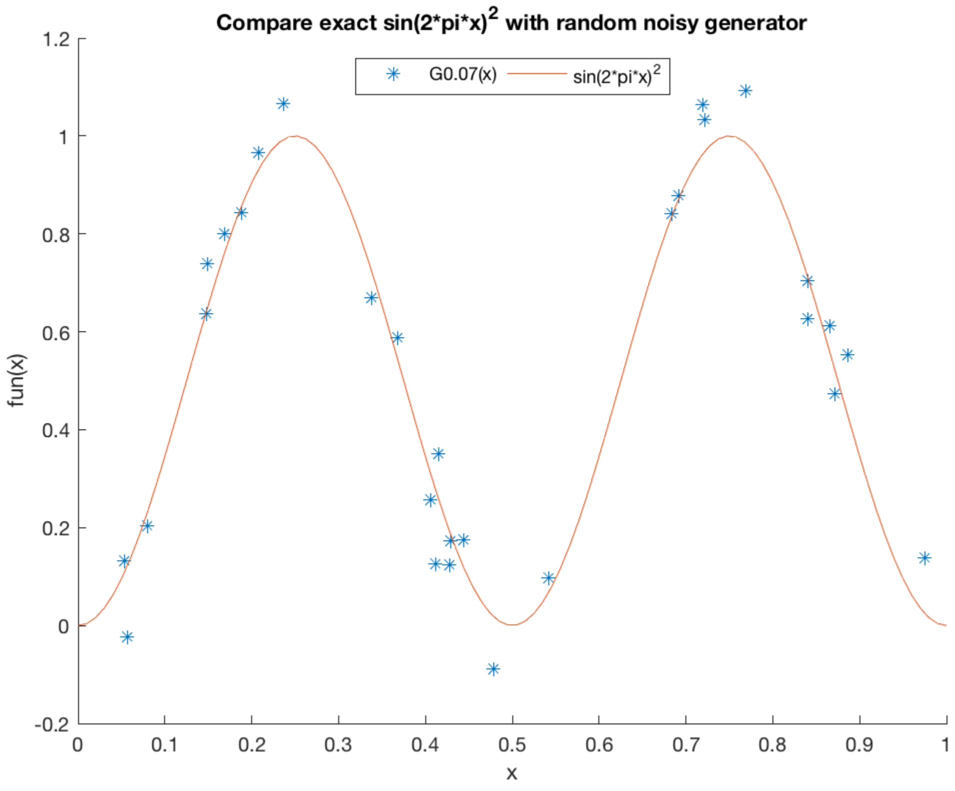
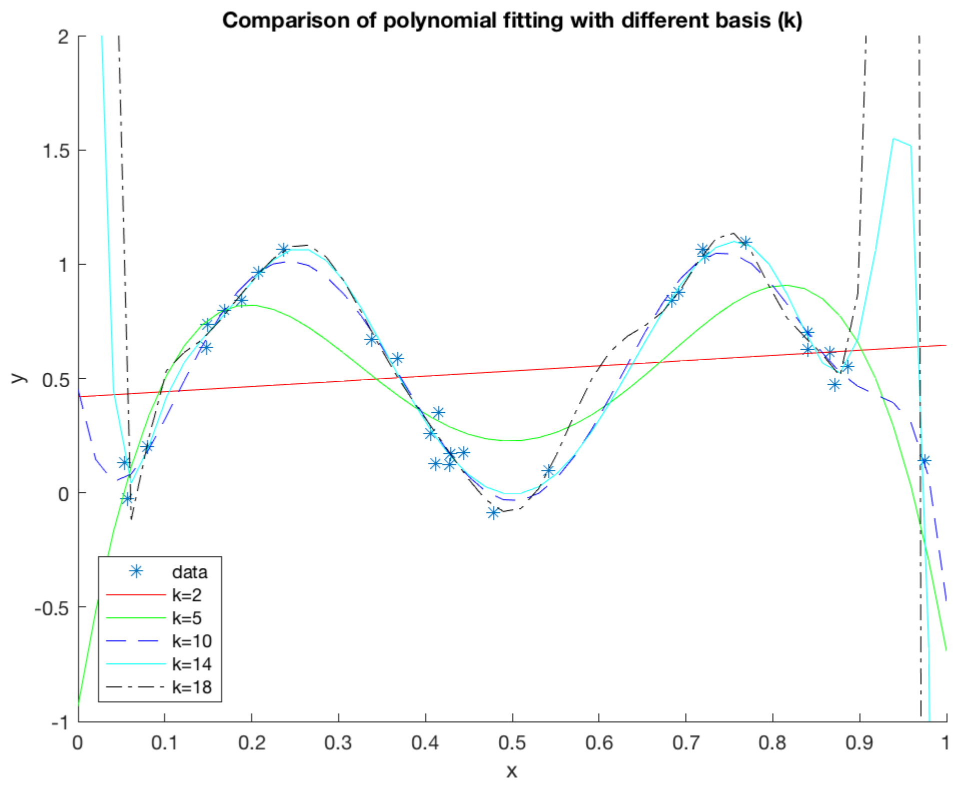
```
%GI07 Homework 1 Question 2 Part 2
%Klaudia Ludwisiak

%%========LINEAR=REGRESSION==Part=B=====overfitting========
%1st Setup the problem
close all;
clear all;

x=rand(1,30); %sample from uniform distrib. between 0 and 1
sd=0.07; %standard deviation
G=Gofx(x,0,sd);

S=[x;G]'; %30 by 2 matrix where 1st col is x and 2nd col is G(x)

figure;
hold on;
scatter(x,G,'*');
plot(linspace(0,1,100),(sin(2*pi*linspace(0,1,100))).^2);
xlabel('x');
ylabel('fun(x)');
title('Compare exact sin(2*pi*x)^2 with random noisy generator');
legend('G0.07(x)','sin(2*pi*x)^2','Location','north','Orientation','horizontal')

% Fit the data set with a polynomial bases of dimension k = 2, 5, 10,
 14, 18
% plot each of these 5 curves superimposed over a plot of data points

% initialise variables:
% dimension:
k=[2, 5, 10, 14, 18];
% weights:
w=cell(size(k));
% polynomial fitted values of x
fx=cell(size(k));
% array of equally spaced values used for plotting (smooth curve
 fitting)
dummy=linspace(0,1,50)';
% fitted more x data points
xfit=cell(size(k));

for i=1:length(k)
   fx{i}=MYpolynom(x',k(1,i)); %fit polynomial of desired order to
 data x
   w{i} = fx{i}\G'; %calculate wight
   xfit{i}=MYpolynom(dummy,k(1,i));
    % fit polynomial of desired order to dummy array data
end

% plot various polynomial fits alongside original data:
figure;
hold on;
plot(x,G,'*');
plot(dummy,xfit{1}*w{1},'r');
plot(dummy,xfit{2}*w{2},'g');
plot(dummy,xfit{3}*w{3},'--b');
plot(dummy,xfit{4}*w{4},'c');
plot(dummy,xfit{5}*w{5},'-.k');
title('Comparison of polynomial fitting with different basis (k)');
xlabel('x');
ylabel('y');
axis([0 1 -1 2]);
legend('data','k=2','k=5','k=10','k=14','k=18','Location','southwest');
hold off

%
%========LINEAR=REGRESSION==Part=C=&=D=====log(MSE)==Test=vs=Train========

% Estimating log(mean squared error) using the property: MSE=SSE/m
% PART C:
% Plot the log of the training error versus the polynomial dimension
 k=1:18
% PART D:
% Repeat part C but for newly generate test data

knew=[1:18]';
% generate new test and train sets to be used with an extended amount
 of
% polynomial basis (k rangin from 1 to 18)
% x coordinates:
xtrain = x';
xtest = rand(1000,1);

% use previously created function to obtain noisy sin function
 approximation:
Gtest = Gofx(xtest,0,sd);
% already initalised G analogous to G(train data) but this has wrong
% dimensions hence transpose here onece to avoid repeated transpose
% operation (call this Gt):
Gt = G';

% Initialize vector for new polynomial basis:
% Initialize vector to store log(MSE) results for test set:
logMSEtrain = zeros(size(knew));
logMSEtest = zeros(size(knew));

% calculate log(MSE(k)) :
for i = 1:size(knew,1)
    % set as temporary variables as they change in size with every
 loop
    Ftrain = MYpolynom(xtrain,i); %polynomial coefficients
    Ftest = MYpolynom(xtest,i);
    ww = Ftrain\Gt; %weight factor
    % calculate logMSE for Test and Tarin data (parts C and D)
    logMSEtrain(i,1) = log(sum((Gt - Ftrain*ww).^2)/size(Gt,1));
    logMSEtest(i,1) = log(sum((Gtest - Ftest*ww).^2)/size(Gtest,1));
end
```

```
% Plot for part C:
figure;
plot(knew,logMSEtrain,'-o')
title('Log(MSE) on TRAINING data for polynomial fitting with different
 basis(k)');
ylabel('log(Mean Square Error)');
xlabel('polynomial basis order (k)');

% Plot for Part D:
figure;
plot(knew', logMSEtest,'-o');
xlabel('polynomial basis order (k)');
ylabel('log(Mean Square Error)');
title('Log(MSE) on TEST data for polynomial fitting with different
 basis (k)');

%%========LINEAR=REGRESSION==Part=E=====log(MSE)==Smoothing========

% Setup:

% number of runs:
run = [1:100];

% Initialize matrix to store MSE results for train and test set:
MSEtrain = zeros(length(run),size(knew,1));
MSEtest = zeros(length(run),size(knew,1));

for t=1:length(run)
% generate new test and train for every loop
xtrain = rand(30,1);
xtest = rand(1000,1);

% use previously created function to obtain noisy sin function
 approximation:
Gtrain = Gofx(xtrain,0,sd);
Gtest = Gofx(xtest,0,sd);

    % calculate MSE:
    for i = 1:size(knew,1)
        % set as temporary variables as they change in size with every
 loop
        Ftrain = MYpolynom(xtrain,i); %polynomial coefficients
        Ftest = MYpolynom(xtest,i);
        ww = Ftrain\Gtrain; %weight factor
        % calculate MSE for Test and Tarin data
        MSEtrain(t,i) = sum((Gtrain - Ftrain*ww).^2)/size(Gtrain,1);
        MSEtest(t,i) = sum((Gtest - Ftest*ww).^2)/size(Gtest,1);
    end
end

%now take the log of average MSE results from each run for all k
 values:
logAveMSEtrain = log(mean(MSEtrain));
```

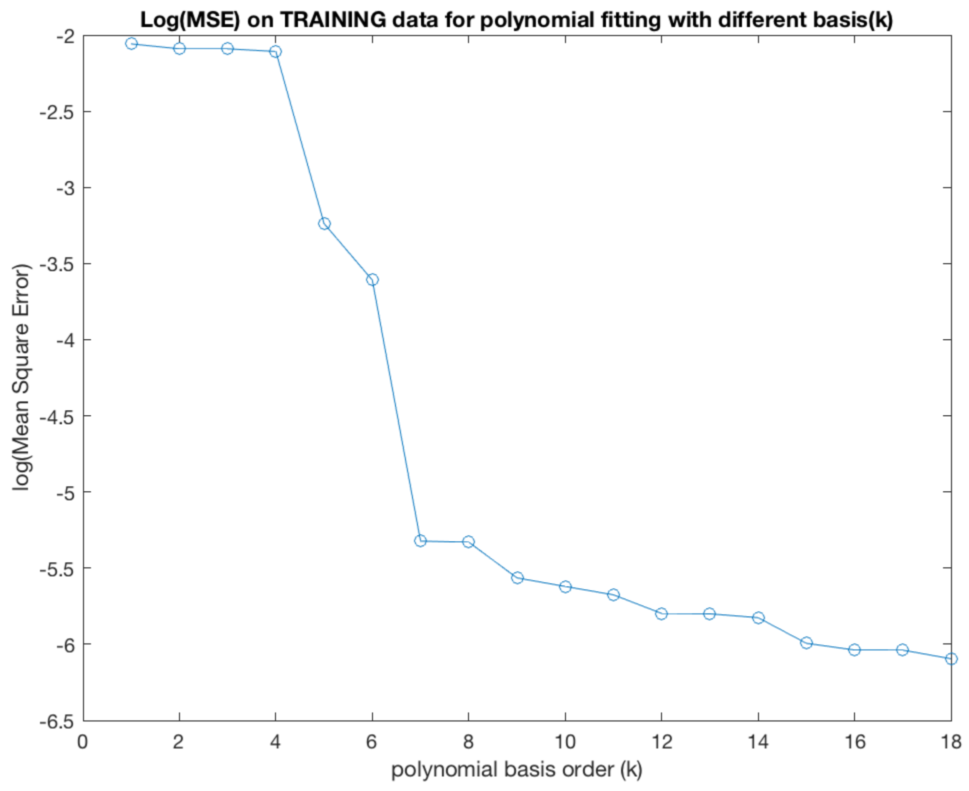
```
logAveMSEtest = log(mean(MSEtest));

figure;
hold on;
plot(knew',logAveMSEtrain);
plot(knew',logAveMSEtest);
xlabel('polynomial basis order (k)');
ylabel('log(Ave(MSE))');
legend('train','test','Location','northeast');
title('Smoothing of log(MSE) results for 100 random trials');
hold off;

Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
Warning: Rank deficient, rank = 17, tol =
3.648565e-14.
```

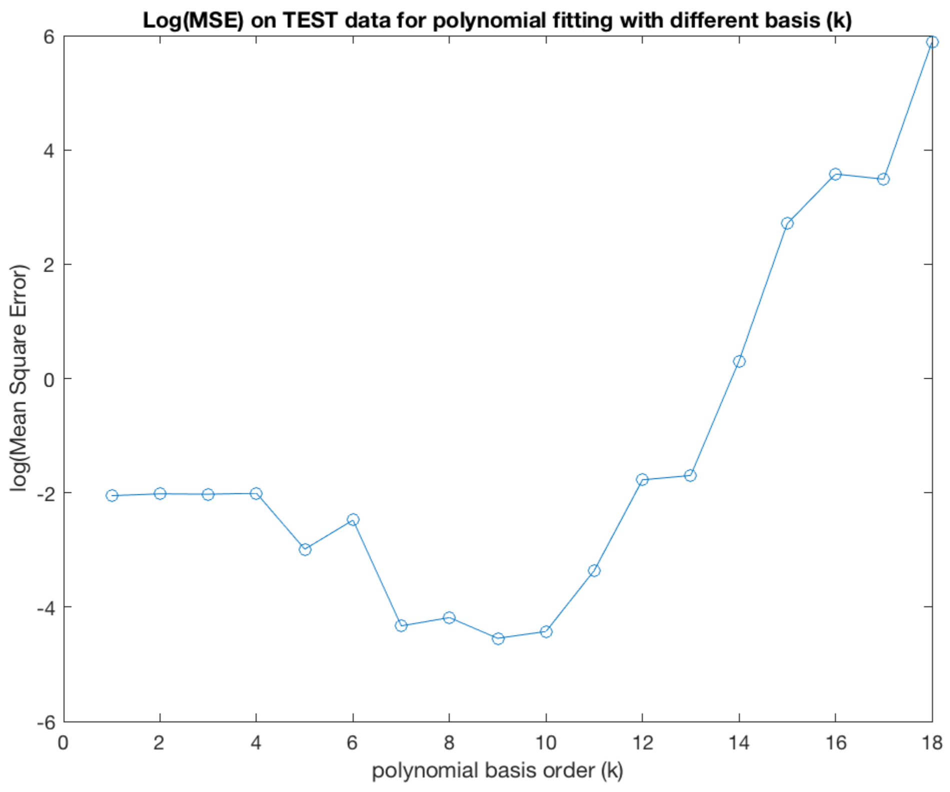


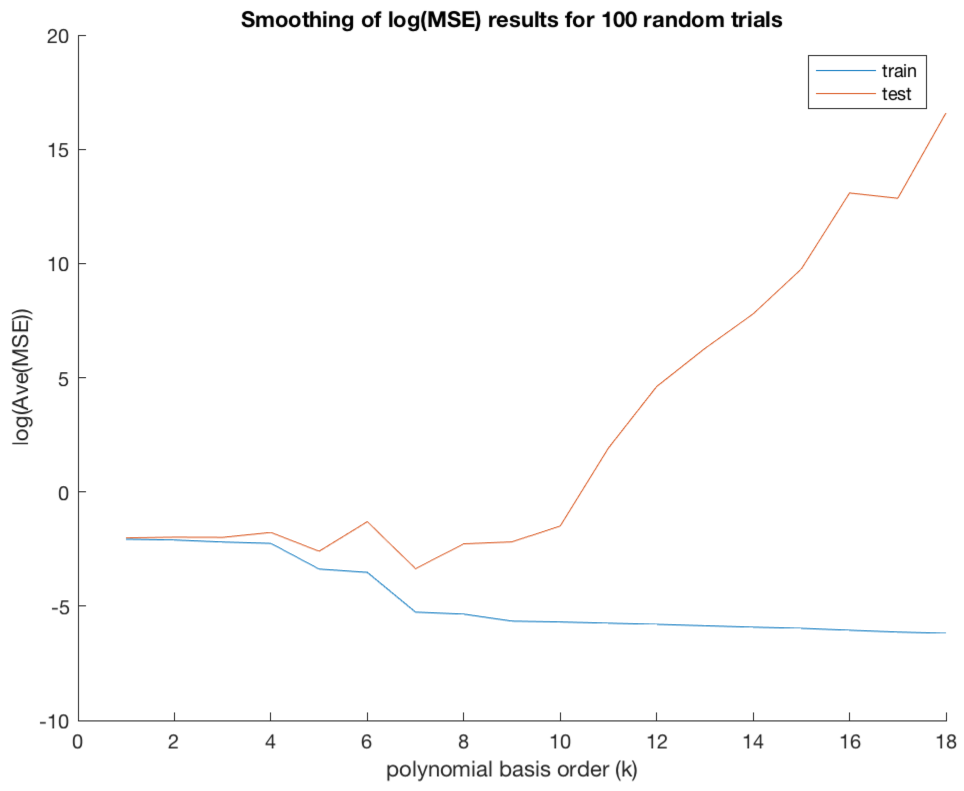
Compare exact sin(2*pi*x)^2 with random noisy generator
G0.07(x)
sin(2*pi*x)^2
fun(x)
x



Comparison of polynomial fitting with different basis (k)
data
k=2
k=5
k=10
k=14
k=18
y
x


Log(MSE) on TRAINING data for polynomial fitting with different basis(k)
log(Mean Square Error)
polynomial basis order (k)
-2
-2.5
-3
-3.5
-4
-4.5
-5
-5.5
-6
-6.5
0
2
4
6
8
10
12
14
16
18


Log(MSE) on TEST data for polynomial fitting with different basis (k)
log(Mean Square Error)
polynomial basis order (k)
6
4
2
0
-2
-4
-6
0
2
4
6
8
10
12
14
16
18



Published with MATLAB® R2016b