1st - Questions from interpolation, regression

A problem

To establish a connection between a highway and a near main road, some points of the two roads were measured, as well as the points of the two planned variations of the connecting roads. Plot the roads, and determine which version of the connecting roads would be shorter? (20 points)

a. Load and display highway points from A1.txt file (2 points)

b. Fit a cubic, second-order spline to the points of the highway and plot the fitted curve in the figure. (2 points)

c. Load the points of the near main road from A2.txt file and plot them (2 points)

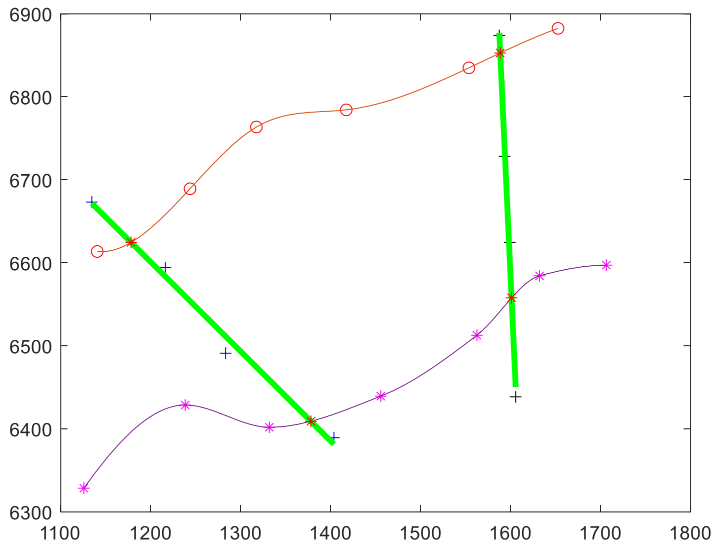
d. Fit a cubic, first-order spline to the points of the main road and plot the curve in the figure. (2 points)

e. Load the points of the two planned variants of the connecting road from A3.txt and A4.txt files. In both cases, examine the value of the collinearity (regression coefficient) of the points. If you find that the points can fall on a line, define the parameters of these lines and plot them in the figure as well. If you find that the points are not collinear, approximate the points with a parabola and plot it in the figure. (6 points)

f. Determine the intersections of the two variants of the connecting road and the highway and main road. (4 points)

g. Calculate the lengths of the two variants of the connecting road and answer the question of the task, which would be shorter? (2 points)

```
% A
clc; clear all; close all;
% a) highway points – 2p
UT1=load('A1.txt')
x1=UT1(:,1); y1=UT1(:,2);
figure(1); plot(x1,y1,'ro')

% b) Spline – 2p
sp1 = @(t) spline(x1,y1,t)
hold on; fplot(sp1,[min(x1),max(x1)])

% c) main road – 2p
UT2=load('A2.txt')
x2=UT2(:,1); y2=UT2(:,2);
plot(x2,y2,'m*')

% d) Hermite interpolation – 2p
sp2 = @(t) interp1(x2,y2,t,'pchip')
fplot(sp2,[min(x2),max(x2)])

% e) connecting road - 6 p
% first variant
VO1=load('A3.txt')
x3=VO1(:,1); y3=VO1(:,2);
plot(x3,y3,'b+')
% collinearity
r = corr2(x3,y3) % -0.9935
```



```
% line
p1 = polyfit(x3,y3,1) % -1.0782        7895
e1 = @(x) polyval(p1,x)
fplot(e1,[min(x3),max(x3)],'g','LineWidth',3);
%  second variant
VO2=load('A4.txt')
x4=VO2(:,1); y4=VO2(:,2);
plot(x4,y4,'k+')
%  collinearity
r = corr2(x4,y4) % -0.9964
% line
p2 = polyfit(x4,y4,1) %   -23.441       44088
e2 = @(x) polyval(p2,x)
fplot(e2,[min(x4),max(x4)],'g','LineWidth',3);

% f) intersections- 4p
% highway – 1st variant
me1=@(t)sp1(t)-e1(t)
me1_x=fzero(me1,1200) % 1178.2
me1_y=sp1(me1_x) % 6624.7
plot(me1_x,me1_y,'r*')
% main road – 1st variant
me2=@(t)sp2(t)-e1(t)
me2_x=fzero(me2,1300) % 1378.1
me2_y=sp2(me2_x) % 6409.1
plot(me2_x,me2_y,'r*')
% highway – 2nd variant
me3=@(t)sp1(t)-e2(t)
me3_x=fzero(me3,1600)  % 1588.5
me3_y=sp1(me3_x) % 6852.8
plot(me3_x,me3_y,'r*')
% main road – 2nd variant
me4=@(t)sp2(t)-e2(t)
me4_x=fzero(me4,1600) % 1601.1
me4_y=sp2(me4_x) % 6557.8
plot(me4_x,me4_y,'r*')

%  g) distance - 2 pont
%  1st
tav1=sqrt((me2_x-me1_x)^2+(me2_y-me1_y)^2) % 293.9535
%  2nd
tav2=sqrt((me4_x-me3_x)^2+(me4_y-me3_y)^2) % 295.2737
%  The first is shorter
```

B problem

To establish a connection between a highway and a near main road, some points of the two roads were measured, as well as the points of the two planned variations of the connecting roads. Plot the roads, and determine which version of the connecting roads would be shorter? (20 points)

a. Load and display highway points from B1.txt file (2 points)

b. Fit a cubic, first-order spline to the points of the highway and plot the fitted curve in the figure. (2 points)

c. Load the points of the near main road from B2.txt file and plot them (2 points)

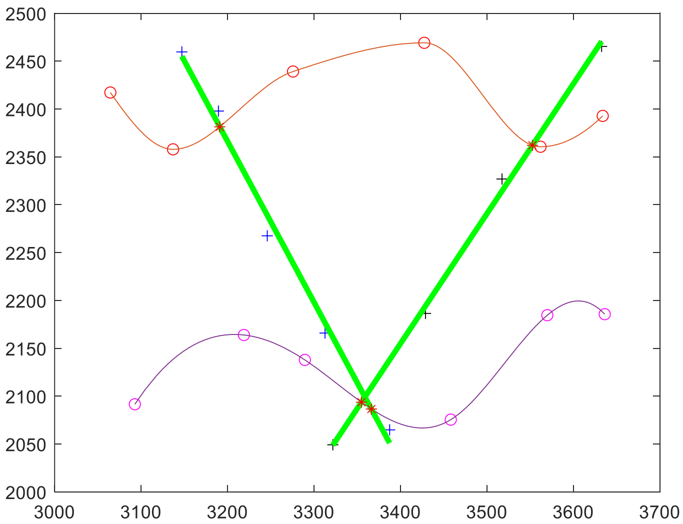
d. Fit a cubic, second-order spline to the points of the main road and plot the curve in the figure. (2 points)

e. Load the points of the two planned variants of the connecting road from B3.txt and B4.txt files. In both cases, examine the value of the collinearity (regression coefficient) of the points. If you find that the points can fall on a line, define the parameters of these lines and plot them in the figure as well. If you find that the points are not collinear, approximate the points with a parabola and plot it in the figure. (6 points)

f. Determine the intersections of the two variants of the connecting road and the highway and main road. (4 points)

g. Calculate the lengths of the two variants of the connecting road and answer the question of the task, which would be shorter? (2 points)

```
%  B
clc; clear all; close all;
%  a) highway - 2 p
UT1=load('B1.txt')
x1=UT1(:,1); y1=UT1(:,2);
figure(1); plot(x1,y1,'ro')

%  b) Hermite interpolation - 2 p
sp1 = @(t) interp1(x1,y1,t,'pchip')
hold on; fplot(sp1,[min(x1),max(x1)])

%  main road - 2 p
UT2=load('B2.txt')
x2=UT2(:,1); y2=UT2(:,2);
plot(x2,y2,'mo');

%  d) Spline - 2 p
sp2 = @(t) spline(x2,y2,t)
fplot(sp2,[min(x2),max(x2)])
%
%  e) connecting road - 6 p
% 1st variant
OK1=load('B3.txt')
x3=OK1(:,1); y3=OK1(:,2);
plot(x3,y3,'b+')
%  collinearity
r = corr2(x3,y3) %  -0.9953
% line
p1 = polyfit(x3,y3,1) %  -1.6814       7746.8
e1 = @(x) polyval(p1,x)
fplot(e1,[min(x3),max(x3)],'g','LineWidth',3);
%  2nd variant
OK2=load('B4.txt')
x4=OK2(:,1); ;y4=OK2(:,2)
plot(x4,y4,'k+')
%  A collinearity
r = corr2(x4,y4) %  -0.9987
% line
p2 = polyfit(x4,y4,1) % 1.3573       -2459.8

e2 = @(x) polyval(p2,x)
fplot(e2,[min(x4),max(x4)],'g','LineWidth',3);

% f) intersections - 4 p
% highway – 1st variant
```



```
oe1=@(t)sp1(t)-e1(t)
oe1_x=fzero(oe1,3100) % 3191
oe1_y=sp1(oe1_x) %  2381.4
plot(oe1_x,oe1_y,'r*')
%
% main road – 1st variant
oe2=@(t)sp2(t)-e1(t)
oe2_x=fzero(oe2,3300) % 3366.3
oe2_y=sp2(oe2_x) %  2086.5
plot(oe2_x,oe2_y,'r*')
%
% highway – 2nd variant
oe3=@(t)sp1(t)-e2(t)
oe3_x=fzero(oe3,3500) % 3552.6
oe3_y=sp1(oe3_x) % 2362.1
plot(oe3_x,oe3_y,'r*')
%
% main road – 2nd variant
oe4=@(t)sp2(t)-e2(t)
oe4_x=fzero(oe4,3300) %  3354.8
oe4_y=sp2(oe4_x) %  2093.7
plot(oe4_x,oe4_y,'r*')
%
%  g) distance- 2 p
%  1st
tav1=sqrt((oe2_x-oe1_x)^2+(oe2_y-oe1_y)^2) % 343.056
%  2nd
tav2=sqrt((oe4_x-oe3_x)^2+(oe4_y-oe3_y)^2) % 333.378
%  The 2nd is shorter
```

C problem

Fit regression curves approximating two different data sets, then determine their intersection points and calculate their distance. Which curve fits better? (20 points)

Note: Do not forget to use the point symbol when dividing with a vector (./) in the same way as with multiplication (.*) and exponentiation (.^) of vectors to perform element by element operations!

a. Load the coordinates of the points of the first curve from C1.txt file and plot them (2 points)

b. A function in the form of $y = \frac{1}{m \cdot x + k}$ can be fitted to the points. Define the parameters of the function and use them to plot the curve in the figure. Determine the residuals and display them in a bar chart. Calculate the corrected empirical standard deviation of the approximation. (6 points)

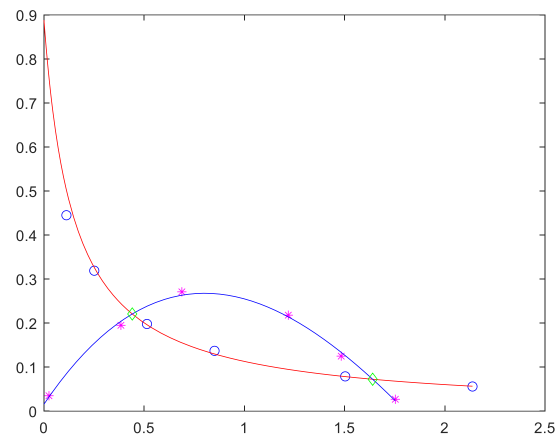
c. Load the second dataset from C2.txt file and display them in the same figure as the first curve (2 points)

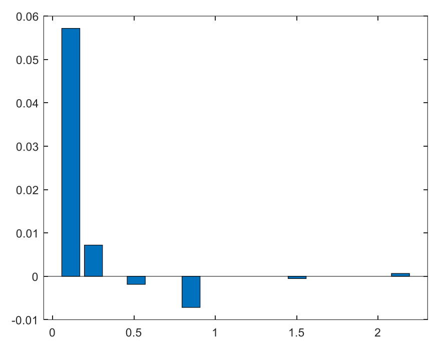
d. Determine the coefficients of the global cubic polynomial that best approximates the points and plot the curve in the figure. Determine the residuals of this approximation, display them in another figure using a bar chart, and calculate the corrected empirical standard deviation of the approximation. (6 points)

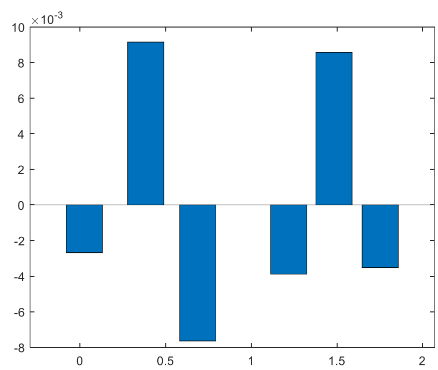
e. Determine the coordinates of the intersection points of the two curves and plot the points in the figure. (2 points)

f. Calculate the distance between the intersection points and answer the question of the problem, which curve is more accurate? (2 points)

```
% C
clc; clear all; close all;
% a) 1st dataset- 2 p
adat1=load('C1.txt')
x1=adat1(:,1); y1=adat1(:,2)
% points
figure (1); hold on; plot(x1,y1,'bo')

% b) regression: y=1/(m*x+k) - 6 p
X=x1
Y=1./y1
A=[X.^1 X.^0]
B=Y
c=A\B
% equation
m=c(1) % 7.7281
k=c(2) % 1.1259
f1=@(t)1./(m*t+k)
fplot(f1,[0 max(x1)],'r-')
% residuals
re1=f1(x1)-y1
figure(2); bar(x1,re1)
S1 = sum(re1.^2)
n1 = length(x1)   % 6
np = 2            % k, m
szoras1 = sqrt(S1/(n1-np))
 % corrected empirical standard deviation:  0.029045

% c) second dataset - 2 p
adat2=load('C2.txt')
x2=adat2(:,1); y2=adat2(:,2)
% points
figure(1); plot(x2,y2,'m*')

% d) polynomial fitting - 6 p
c = polyfit(x2,y2,3) %   0.0703   -0.5053    0.6742    0.0158
f2 = @(x) polyval(c,x)
fplot(f2,[0 max(x2)],'b-')
% residuals
re2=f2(x2)-y2
figure(3); bar(x2,re2)
S2 = sum(re2.^2) %
n2 = length(x2)   % 6
np = 4           % c(1),c(2),c(3),c(4)
szoras2 = sqrt(S2/(n2-np))
% corrected empirical standard deviation  0.011188

% e) intersections- 2 p
h=@(t) f1(t)-f2(t)
% 1st
mo1_x=fzero(h,0.4) %  0.44052
mo1_y=f1(mo1_x) %  0.22074
figure(1); plot(mo1_x,mo1_y,'gd')
```







```
% 2nd
mo2_x=fzero(h,1.6) %   1.6399
mo2_y=f1(mo2_x) % 0.072467
plot(mo2_x,mo2_y,'gd')

%  f) distance - 2 p
tav=sqrt((mo2_x-mo1_x)^2+(mo2_y-mo1_y)^2) % 1.2086
% 0.029045  > 0.011188 -> the second fit is better
```

D problem

2.) Fit regression curves approximating two different data sets, then determine their intersection points and calculate their distance. Which curve fits better? (20 points)

a. Load the coordinates of the points of the first curve from D1.txt file and plot them (2 points)

b. A function in the form of $y = k \cdot x^m$ can be fitted to the points. Define the parameters of the function and use them to plot the curve in the figure. Determine the residuals and display them in a bar chart. Calculate the corrected empirical standard deviation of the approximation. (6 points)

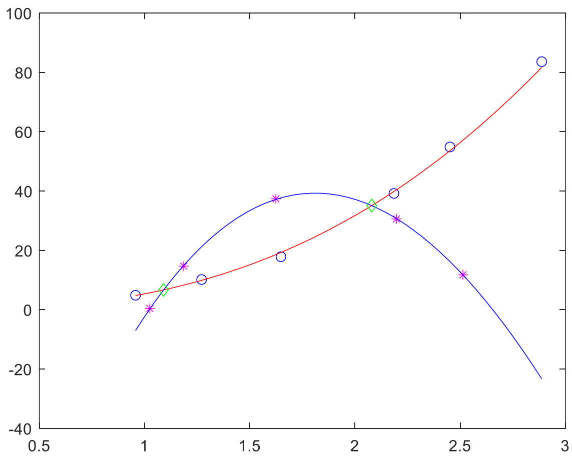
c. Load the second dataset from D2.txt file and display them in the same figure as the first curve (2 points)

d. Determine the coefficients of the global cubic polynomial that best approximates the points and plot the curve in the figure. Determine the residuals of this approximation, display them in another figure using a bar chart, and calculate the corrected empirical standard deviation of the approximation. (6 points)

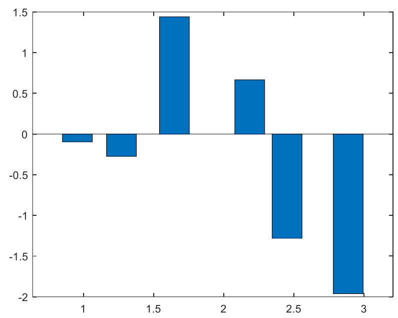
e. Determine the coordinates of the intersection points of the two curves and plot the points in the figure. (2 points)

f. Calculate the distance between the intersection points and answer the question of the problem, which curve is more accurate? (2 points)

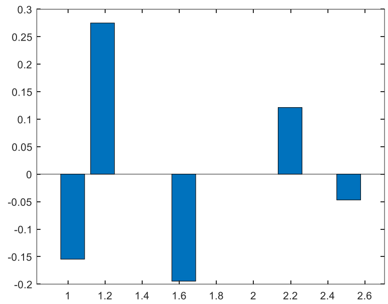
```
%  D
clc; clear all; close all;
% a) 1st dataset - 2 p
adat1=load('D1.txt')
x1=adat1(:,1); y1=adat1(:,2);
%  points
figure (1); hold on;
plot(x1,y1,'bo')

% b)  regression (y=k*x^m) - 6 p
X=log(x1)
Y=log(y1)
A=[X.^1 X.^0]
B=Y
c=A\B %  2.5764, 1.6707
%  equation
m=c(1) % 2.5764
k=exp(c(2)) %  5.3157
f1=@(t) k*t.^m
fplot(f1,[min(x1) max(x1)],'r-')
%  residuals
```



```
re1=f1(x1)-y1
figure(2)
bar(x1,re1)
S1 = sum(re1.^2) %
n1 = length(x1)   % 6
np = 2            % k, m
szoras1 = sqrt(S1/(n1-np))
% 1.4222 - corrected empirical standard deviation
```




```
% c) 2nd dataset - 2 p
adat2=load('D2.txt')
x2=adat2(:,1); y2=adat2(:,2);
% points
figure(1); plot(x2,y2,'m*')

% d) polynomial fitting - 6 p
c=polyfit(x2,y2,3) % 4.9830  -86.4183  263.9029 -184.8340
f2 = @(x) polyval(c,x)
fplot(f2,[min(x1) max(x1)],'b-')
% residuals
re2=f2(x2)-y2
figure(3); bar(x2,re2)
S2 = sum(re2.^2) %
n2 = length(x2)   % 5
np = 4     % c(1),c(2),c(3),c(4)
szoras2 = sqrt(S2/(n2-np))
% 0.39276 - corrected empirical standard deviation
```




```
% e) intersections - 2 p
h=@(t)f1(t)-f2(t)
% 1st
mo1_x=fzero(h,1) % 1.0905
mo1_y=f1(mo1_x) % 6.6449
figure(1); plot(mo1_x,mo1_y,'gd')
% 2nd
mo2_x=fzero(h,2.2) % 2.0796
mo2_y=f1(mo2_x) % 35.059
plot(mo2_x,mo2_y,'gd')

% f) distance - 2 pont
tav=sqrt((mo2_x-mo1_x)^2+(mo2_y-mo1_y)^2) % 28.431
% 1.422    > 0.39276 , the second fit is better
```