1. zh

1) A hiking trail was surveyed with rapid GPS measurement method. The local x-y coordinates of the measured points can be found in the 'dataA.txt' file. This hiking trail is crossed by a bikeway and its vertices are presented in the following table.

x	52	217	281	342	455	512	568
y	42	41	40	38	36	34	33

   a. Plot the hiking trail with black circles and display the bikeway on he same figure with blue crosses! (Figure 1) (2 points)

   b. Fit a cubic, second order spline to the waypoints of the bikeway and display them on the same figure! (Figure 1) (3 points)

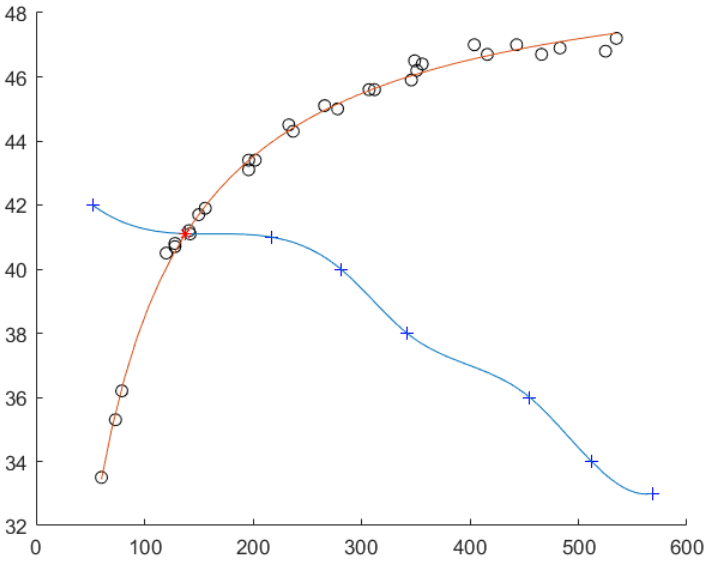
   c. Fit a second order polynomial to the measured GPS points of the hiking trail! Determine the coefficients and plot the fitted curve in a new figure with the measurement points. Determine the number of excess measurements (the degree of freedom) and according to that determine the corrigated standard deviation of the residuals! (Figure 2) (4 points)

   d. Examine an alternative fitting as well! The alternative relation between the measured points in the hiking trail's dataset is in form of $y = \frac{mx}{x+k}$. With linearized regression determine the k,m parameters! Draw the fitted $y = \frac{mx}{x+k}$ function into the diagram! Determine the number of excess measurements (the degree of freedom) and according to that determine the corrigated standard deviation of the residuals! (Figure 2) (5 points)

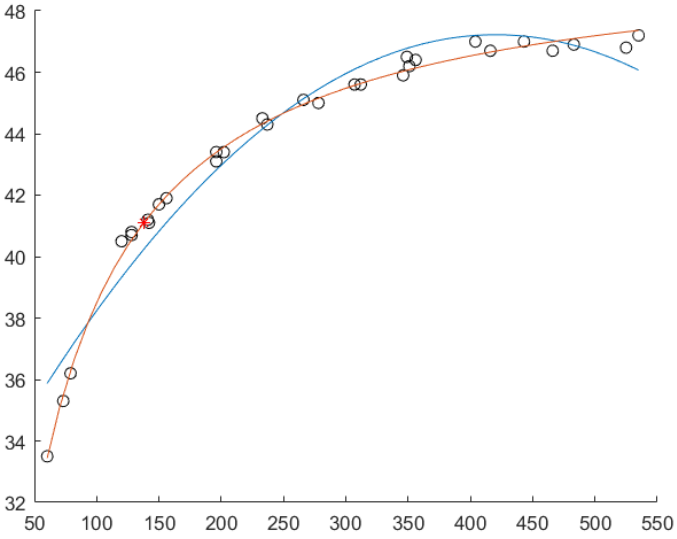
   e. Decide which function fits better on the hiking trail dataset based on the corrigated standard deviation of the residuals and the plots. Use the better fit and find where the hiking trail intersects with the bikeway. Display the better fit of hiking trail on the first plot. Draw the intersection point on both figure with a red star! (4 points)

```
%% Feladat 1
 % Megoldas

clc;
```

```
clear all;
close all;
% a.
data = load('dataA.txt');
x = data(:,1);
y = data(:,2);

cycle_path = [ 52 217   281 342
455 512 568;...
                42  41  40  38
36  34  33 ]';

path_x = cycle_path(:,1);
path_y = cycle_path(:,2);

figure(1)
hold on
plot(x,y,'ko')
plot(path_x,path_y,'b+')
```




```
% b.
sp = @(t)
spline(path_x,path_y,t);
fplot(sp,[min(path_x),max(path_x)])

% c.
figure(2)
hold on
plot(x,y,'ko')
p = polyfit(x,y,2);
format long
p % -0.000087615268288   0.073630792186166  31.753437590856507
format short
fpoly = @(x) polyval(p,x);
fplot(fpoly,[min(x),max(x)])
r_poly = fpoly(x)-y;
S = sum(r_poly.^2);
n = length(x);
np = 2;
% fölös meresek szama
n-np
szoras_poly = sqrt(S/(n-np));
% korrigalt tapasztatlai szoras
szoras_poly   % 0.8227

% d.
Y = 1./y;
X = 1./x;
pXY = polyfit(X,Y,1);

m_est = 1/pXY(2);
k_est = pXY(1)/pXY(2);

m_est % 50.0062
k_est % 29.8676

y_est = @(x) m_est*x./(x+k_est);
fplot(y_est,[min(x),max(x)])

r_alter = y_est(x)-y;
S = sum(r_alter.^2);
```



```
n = length(x);
np = 2;
n-np
szoras_alter = sqrt(S/(n-np))
% korrigalt tapasztatlai szoras
szoras_alter % 0.2343

% e.
froot = @(x) y_est(x) - sp(x);
X_inter = fzero(froot,100);
Y_inter = sp(X_inter);

figure(1)
fplot(y_est,[min(x),max(x)])
plot(X_inter,Y_inter,'r*')

figure(2)
plot(X_inter,Y_inter,'r*')

X_inter % 137.8863
Y_inter % 41.1029
```

1) A hiking trail was surveyed by a drone. The local x-y coordinates of the measured points can be found in the 'dataB.txt' file. This hiking trail is crossed by a bikeway and its vertices are presented in the following table.

x	52	217	281	342	455	512	568
y	42	41	40	38	36	34	33

   a. Plot the hiking trail with black circles and display the bikeway on he same figure with blue crosses! (Figure 1) (2 points)

   b. Fit a cubic, second order spline to the waypoints of the bikeway and display them on the same figure! (Figure 1) (3 points)

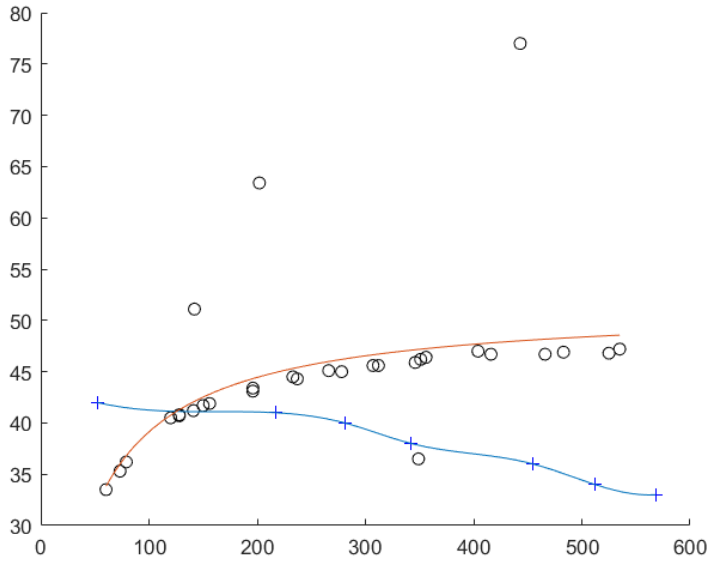
   c. We know that the relation between the measured points in the hiking trail's dataset is in form of $y = \frac{mx}{x+k}$. With linearized regression determine the k,m parameters! Draw the fitted $y = \frac{mx}{x+k}$ function into the diagram! Calculate the residuals and display them on a separate bar plot (histogram)! (Figure 1) (5 points)

   d. During the drone surveys some erroneous measurements were introduced. Filter the outliers based on the residuals. Repeat the the previous fitting with the filtered dataset. Display the new fitted curve alongside with the filtered datapoints in a new figure. Determine the number of excess measurements (the degree of freedom) and according to that determine the corrigated standard deviation of the residuals (Figure 2) (5 points)

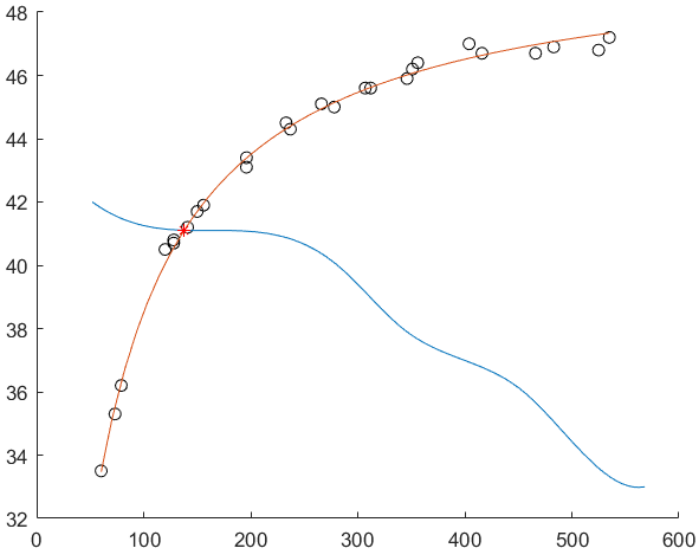
   e. Find the instersection point of the hiking trail and the bikeway. Display the interpolated bikeway on the second figure as well and draw the intersection point with a red star! (Figure 2) (3 points)

```
%% Feladat 2
 % Megoldas

clc;
clear all;
close all;
% a.
data = load('dataB.txt');
```

```
x = data(:,1);
y = data(:,2);

cycle_path = [ 52 217   281 342 455 512 568;...
               42  41   40  38  36  34  33 ]';

path_x = cycle_path(:,1);
path_y = cycle_path(:,2);

figure(1)
hold on
plot(x,y,'ko')
plot(path_x,path_y,'b+')

% b.
sp = @(t) spline(path_x,path_y,t);
fplot(sp,[min(path_x),max(path_x)])

% d.
Y = 1./y;
X = 1./x;
pXY = polyfit(X,Y,1);

m_est = 1/pXY(2);
k_est = pXY(1)/pXY(2);

m_est % 51.4045
k_est % 31.3176

y_est = @(x) m_est*x./(x+k_est);
fplot(y_est,[min(x),max(x)])

r = y_est(x)-y;

x_filtered = x(abs(r)<2);
y_filtered = y(abs(r)<2);


Y_filt = 1./y_filtered;
X_filt = 1./x_filtered;
pXY_filt = polyfit(X_filt,Y_filt,1);
```



```
m_est_filt = 1/pXY_filt(2);
k_est_filt =
pXY_filt(1)/pXY_filt(2);

m_est_filt % 49.9678
k_est_filt % 29.7236

y_est_filt = @(x)
m_est_filt*x./(x+k_est_filt);

r_filt = y_est_filt(x_filtered)-
y_filtered;

S = sum(r_filt.^2);

n = length(x_filtered);
np = 2;
n-np
szoras_filter = sqrt(S/(n-np))
% korrigalt tapasztatlai szoras
szoras_filter % 0.2260

figure(2)
hold on
plot(x_filtered,y_filtered,'ko')
fplot(sp,[min(path_x),max(path_x)])
fplot(y_est_filt,[min(x),max(x)])

% e.
froot = @(x) y_est_filt(x) - sp(x);
X_inter = fzero(froot,100);
Y_inter = sp(X_inter);

figure(2)
plot(X_inter,Y_inter,'r*')

X_inter % 137.8178
Y_inter % 41.1030
```



1) A hiking trail was surveyed with drone and GPS. The local x-y coordinates of datapoints can be found in the 'dataC_GPS.txt' and the 'dataC_Drone.txt' file. The first part of the hiking trail was measured by GPS and the second part was measured by drone. Unfortunately the drone measurement sufferes from a constant bias in the y direction. Also the middle part of the hiking trail was not surveyed. Fix the drone measurement and determine the missing part of the hiking trail. We know that the relation between the hiking trail's x and y coordinates is in form of

$$y = a * \cos\left(\frac{x}{50}\right) + b * (x - 750) + c$$

   a. Read the data datapoints from the files and plot them on the same figure but with different markers. Use blue crosses for the GPS data and green circle for the drone data. (Figure 1) (2 points)

   b. Fit the given function to the GPS measurement points and determine the a,b,c coefficients. Determine the number of excess measurements (the degree of freedom) and according to that determine the corrigated standard deviation of the residuals! Display the fitted curve! (Figure 1)) (3 points)

   c. Fit the given function to the drone measurement points and determine the a,b,c coefficients. Determine the number of excess measurements (the degree of freedom) and according to

that determine the corrigated standard deviation of the residuals! Display the fitted curve! (Figure 1) (3 points)

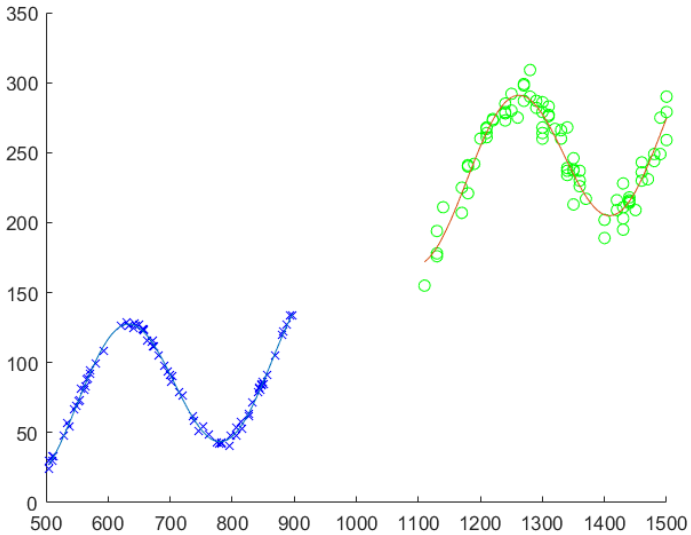
d. Determine the y bias of the drone measurement from the two coefficeint set. The bias can be derived as a difference of the constant parameters from the two coeffeiceint set. Fix the drone measurement with the derived bias and display the GPS and the corrected drone points in a new diagram (Figure 2)! (2 points)

e. Fit the given function to the combined data of GPS and the corrected drone and determine the a,b,c coefficients. Display the fitted curve in the whole domain (Figure 2) (3 points)

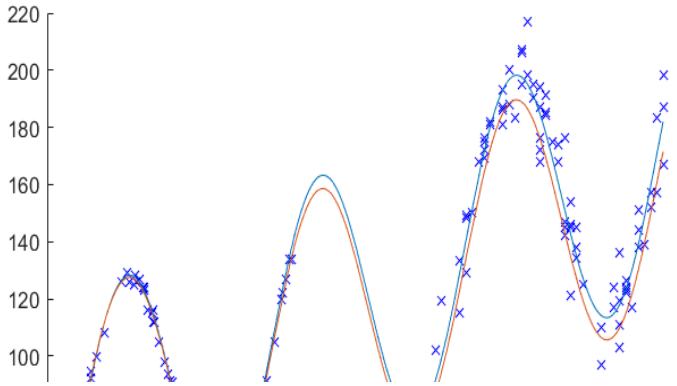
f. Extrapolate the fitted curve derived only from the GPS data to whole domain. Determine the corrigated standard deviation of the residuals in the extrapolated domain! (Figure 2)(3 points)

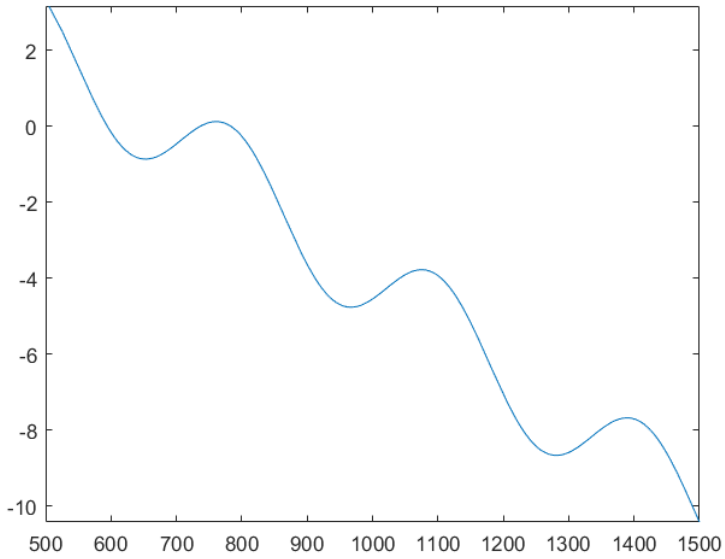
g. Display the differences of the fitted curve based on only GPS data and the fitted curve based on the GPS data the the corrected drone data on the whole domain (Figure 3) (2 points)

```
%% Feladat 3
 % Megoldas

clc;
clear all;
close all;
%% a.
data_gps = load('dataC_GPS.txt');
data_drone = load('dataC_Drone.txt');

x1 = data_gps(:,1);
y1 = data_gps(:,2);

x2 = data_drone(:,1);
y2 = data_drone(:,2);

x = [x1;x2];
figure(1)
hold on
plot(x1,y1,'bx');
plot(x2,y2,'go');

%% b
A1 = [cos(x1/50)  (x1-750)
ones(length(x1),1)];
b1 = A1\y1
b1
%    49.518684456306822
%     0.098890029726097
%    89.765637661879737
f1 = @(x) b1(1)*cos(x/50) +
b1(2)*(x-750)+ b1(3);
figure(1)
hold on
fplot(f1,[min(x1),max(x1)])

r1 = f1(x1)-y1;
S1 = sum(r1.^2);
n1 = length(x1);
np = 3;
szoras1 = sqrt(S1/(n1-np));
szoras1 % 2.081082858239105
```



```
%% c
A2 = [cos(x2/50) (x2-750) ones(length(x2),1)];
b2 = A2\y2
b2

% 1.0e+02 *
%    0.515656981315730
%    0.001132880503023
%    1.816674308211202

f2 = @(x) b2(1)*cos(x/50) +
b2(2)*(x-750)+ b2(3);
figure(1)
hold on
fplot(f2,[min(x2),max(x2)]);

r2 = f2(x2)-y2;
S2 = sum(r2.^2);
n2 = length(x2);
np = 3;
szoras2 = sqrt(S2/(n2-np))
szoras2 % 9.977525902119572
```






```
%% d.
jump = b2(3)-b1(3);
y_rec = [y1;y2-jump];
figure(2); hold on;
plot(x,y_rec,'bx');

%% e.
A = [cos(x/50) (x-750)
ones(length(x),1)];
b_rec = A\y_rec;
b_rec
%   50.838357221084600
%    0.111303343917813
%   90.668603188349437

figure(2)
f_rec = @(x) b_rec(1)*cos(x/50) + b_rec(2)*(x-750)+ b_rec(3);
fplot(f_rec,[min(x),max(x)])

%% f
figure(2)
fplot(f1,[min(x),max(x)])

r12 = f1(x2)-y2+jump;
S12 = sum(r12.^2);
n2 = length(x2);
np = 3;
szoras12 = sqrt(S12/(n2-np));
szoras12 % 13.130728088603126
% ez így kevéssé helyes mert np = 0 mivel extrapoláció volt.
% Helyesen így lenne
% szoras12 = sqrt(S12/(n2)); % 12.8853
% Javaslom bármelyiket is használja a hallgató kapja meg a max pontot erre
% a részre
```

```
%% g
figure(3)
f_diff = @(x) f1(x)-f_rec(x)
fplot(f_diff,[min(x),max(x)])
```

1. A straight road was surveyed and the local x-y coordinates of the measurement points were collected in the 'dataD.txt' file. The measurements suffer from noise and systematic bias coming from coordinate transformation. The x and y corrections for the bias are expressed as the function of x and presented in the following tables. Interpolate the x and y corrections as a function of x and correct the measurments coordinates. (For sake of clarification: the corrections shall be added to the data points)

x	0	100	200	300	400	500	600
Δx	-5	-1	7	15	19	24	36

x	0	50	150	250	350	450	550
Δy	52	61	63	72	75	79	81

   a. Load the road measurement points and plot them! (Figure 1) (2 points)

   b. Plot the Δx and Δy corrections in a new figure. Fit a cubic, first order Hermite spline to the corrections and display them on the same figure! (Figure 2) (4 points)

   c. Calculate the x corrections of the measured roadpoints based on the x coordinates with the help of the previous interpolation and correct the x coordinates. Then calculate the y corrections of the measured roadpoint based on the corrected x coordinates with the help the previous interpolation and correct the y coordinates. Plot the corrected points with a different marker (Figure 1) (4 points)

   d. Fit a straight line to the correctied roadpoints and determine the coefficients! Determine the number of excess measurements (the degree of freedom) and according to that determine the corrigated standard deviation of the residuals. Plot the fitted curve! (Figure 1) (4 points)

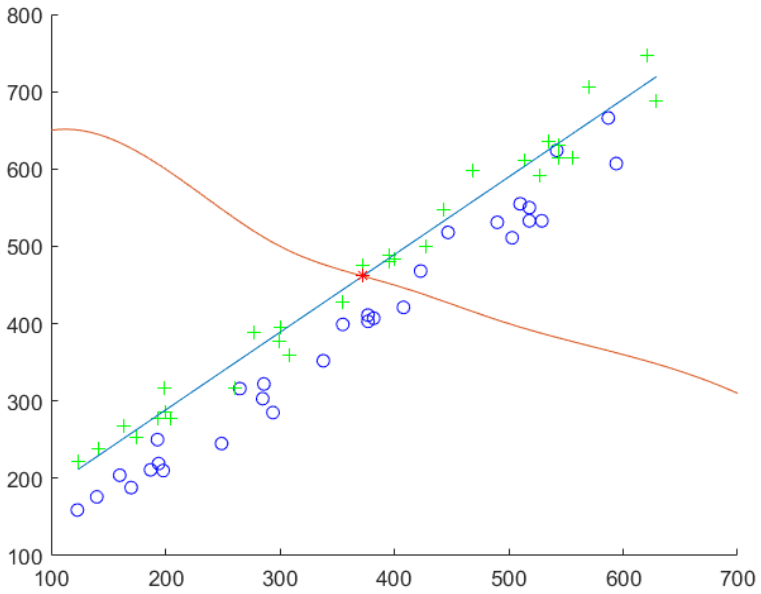
   e. The surveyed road has an intersection with another road which coordinates are error free and can be found in the following table:

x	100	200	300	400	500	600	700
y	650	600	500	450	400	360	310

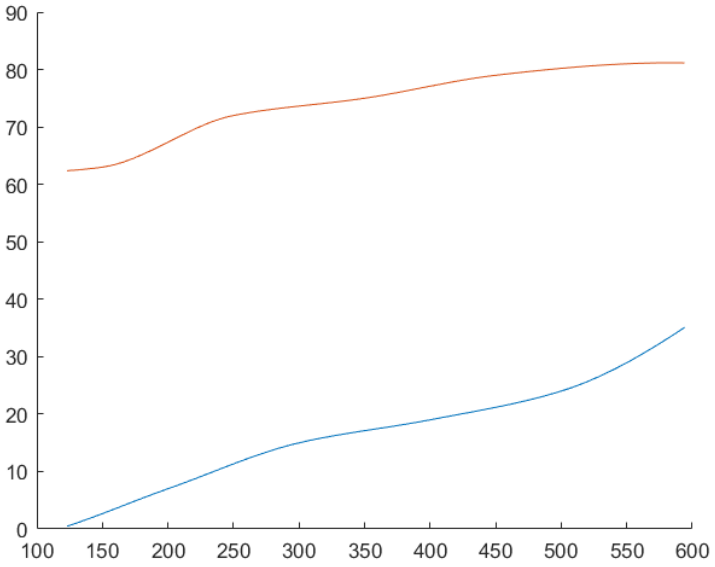
   Interpolate the second road with cubic, second order spline and plot it. Determine the intersection of the two roads and mark the intersection point with a red star (Figure 1) (4 points)

```
%% Feladat 4
 % Megoldas
clc;
clear all;
close all;
format short
```

```
%% a.
data = load('dataD.txt');

x = data(:,1);
y = data(:,2);

figure(1)
hold on
plot(x,y,'bo')

%% b.
correction_x = [0 100 200 300
400 500 600;
                -5 -1  7  15  19
24   36];
correction_y = [0  50 150 250
350 450 550;
                52 61  63  72
75  79  81];

corr_intp_x = @(t)
```




```
interp1(correction_x(1,:),correction_x(2,:),t,'pchip');
corr_intp_y = @(t) interp1(correction_y(1,:),correction_y(2,:),t,'pchip');

figure(2)
hold on
fplot(corr_intp_x,[min(x),max(x)])
fplot(corr_intp_y,[min(x),max(x)])

%% c.
x_corrected = x + corr_intp_x(x);
y_corrected = y +
corr_intp_y(x_corrected);
figure(1)
plot(x_corrected,y_corrected,'g+')

%% d
p =
polyfit(x_corrected,y_corrected,1);
fline = @(x) polyval(p,x);

r = fline(x_corrected)-y_corrected;
S = sum(r.^2);
n = length(x);
np = 2;
szoras = sqrt(S/(n-np));
szoras  % 22.3304
```




```
figure(1)
fplot(fline,[min(x_corrected),max(x_corrected)])

%% e.
road = [ 100 200 300 400 500 600 700;
           650 600 500 450 400 360 310];


froad = @(t) spline(road(1,:),road(2,:),t);

figure(1)
fplot(froad,[min(road(1,:)),max(road(1,:))])
```

```
fcross = @(x) fline(x)-froad(x);
crosspoint_x = fzero(fcross,400);
crosspoint_y = fline(crosspoint_x);

crosspoint_x % 372.4109
crosspoint_y % 461.4809
plot(crosspoint_x,crosspoint_y,'r*')
```