TERRAIN GRIDDING USING LOCAL INTERPOLATION METHODS

ESSE 4640: Lab 4

Karthiga Thanaseelan
11/5/2021

Introduction

The focus of this lab is using local interpolation methods for gridding and terrain surface modelling. The process of gridding is used to estimate the elevations at specific points, grid points, that are spaced apart in a regular grid pattern. Both the Nearest Neighbour Elevation Assignment method and the Inverse Distance Weighting method were used to interpolate the elevations given the irregularly distributed set of data points. A program was created in Matlab to generate a grid with 100m spacing and calculate the elevations at the grid points.

Methodology and Results

Question 1: Reading the Data and Generating the Grid

The given data file “DTM_Lab4_XYZ_F2021.txt” with the X, Y, and Z coordinates was input into an excel sheet which was read in by column using the “xlsread” function to create variables for X, Y, and Z coordinates. The were 199 irregularly spaced coordinates given in the file. The “meshgrid” function was used to create a regular grid with 100m spacing. The generated grid was 51 by 51 which totalled to 2601 grid points. Coordinates that are outside the generated grid were then removed from the list of coordinates. Figure 1 shows the grid and the data points.

Question 2: The Inverse Distance Weighting Method

The Inverse Distance Weighting method assigns each of the grid points a weighted sum of the closest 15 points. The distances were found using the distance formula:

$$d = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2},$$

where i denotes the index of the grid points and j denotes the index of the data points. After these points were selected, the weights were found using the following weighting schemes:

1. $w_i = \frac{1}{d}$
2. $w_i = \frac{1}{d^2}$
3. $w_i = \frac{1}{d^5}$

The weighted sum was then determined as:

$$z(X,Y) = \frac{\sum_{i=1}^{n} w_i Z_i}{\sum_{i=1}^{n} w_i},$$

where i is the index denoting the selected data points.

Figure 2 shows the DEM that was created using the first weighting scheme. Figure 3 shows the DEM that was created using the second weighting scheme. Figure 4 shows the DEM that was created using the third weighting scheme.

Question 3: The Nearest Neighbour Elevation Assignment Method

The Nearest Neighbour Elevation Assignment method, the elevation of the nearest data point is assigned to each grid point. The distances between each data point and grid point were, first, indexed based on the minimum distance from each grid point. The index was then used to assign the grid points an elevation. Table __ shows

Question 4: Statistics and the Differences Between Both Methods

The mean, standard deviations, root mean square errors, minimum, maximum and range of the results using the built-in functions.

	NN and $w_i = \frac{1}{d}$	NN and $w_i = \frac{1}{d^2}$	NN and $w_i = \frac{1}{d^5}$
Mean	0.5442	0.2944	0.0377
Standard Deviations	2.3424	1.8017	1.0667
Root Mean Square Errors	2.4041	1.8250	1.0670
Minimum	−7.0777	−6.6649	−5.9331
Maximum	13.6059	13.1265	11.1095
Range	20.6836	19.7914	17.0427

Question 5: Generating the DEMs

A matrix was created wherein the elevations were matched to their corresponding x and y grid points' coordinates. This was then inputted into the "griddata" function along with the x and y grid points' coordinates and a label describing that data as cubic. The result was then input into the "mesh" function with the x and y grid points' coordinates. This process was repeated for the results of the NN method and all three of the IDW weighting schemes.

Results and Discussion

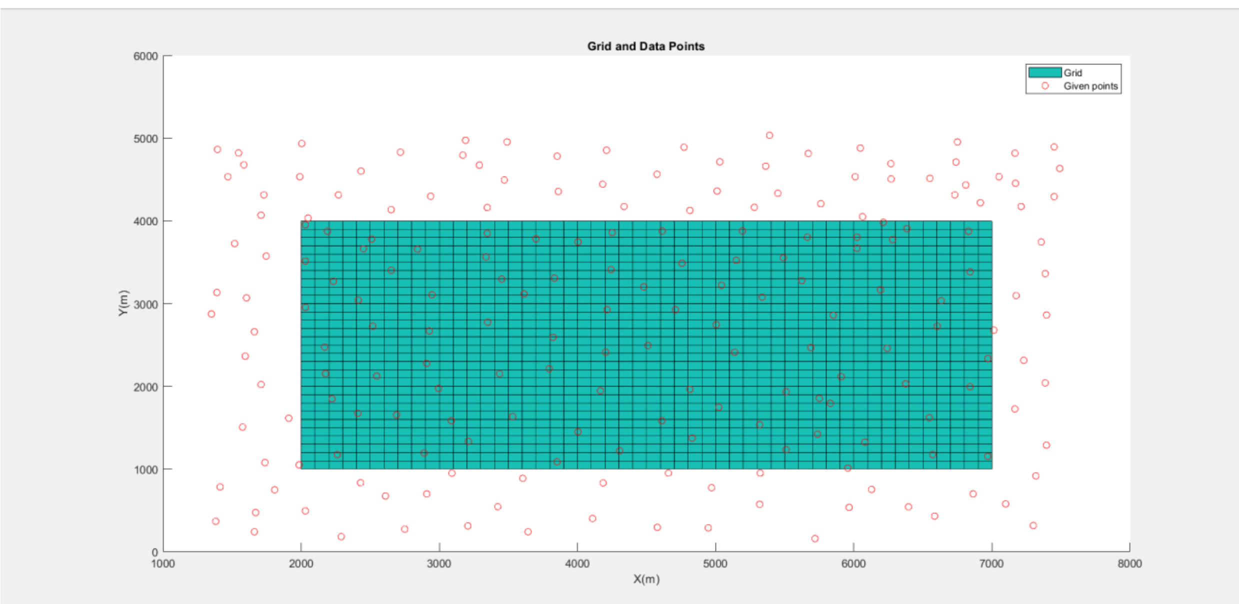

Figure 1: Grid and Data Points

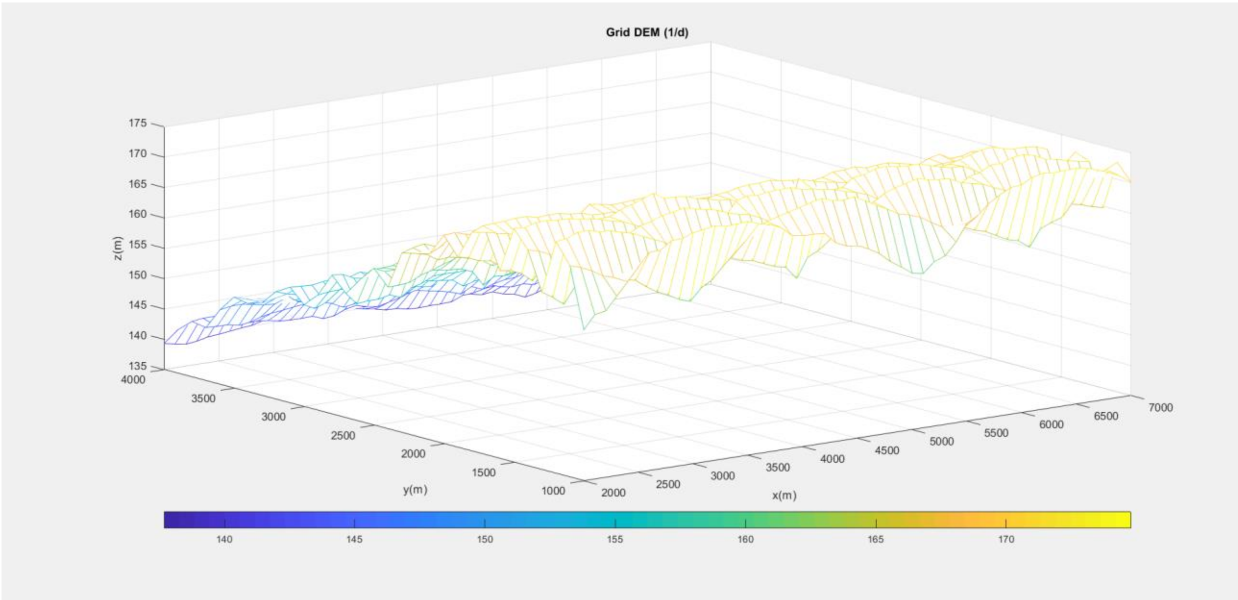

Figure 2: DEM using IDW method and 1/d weight

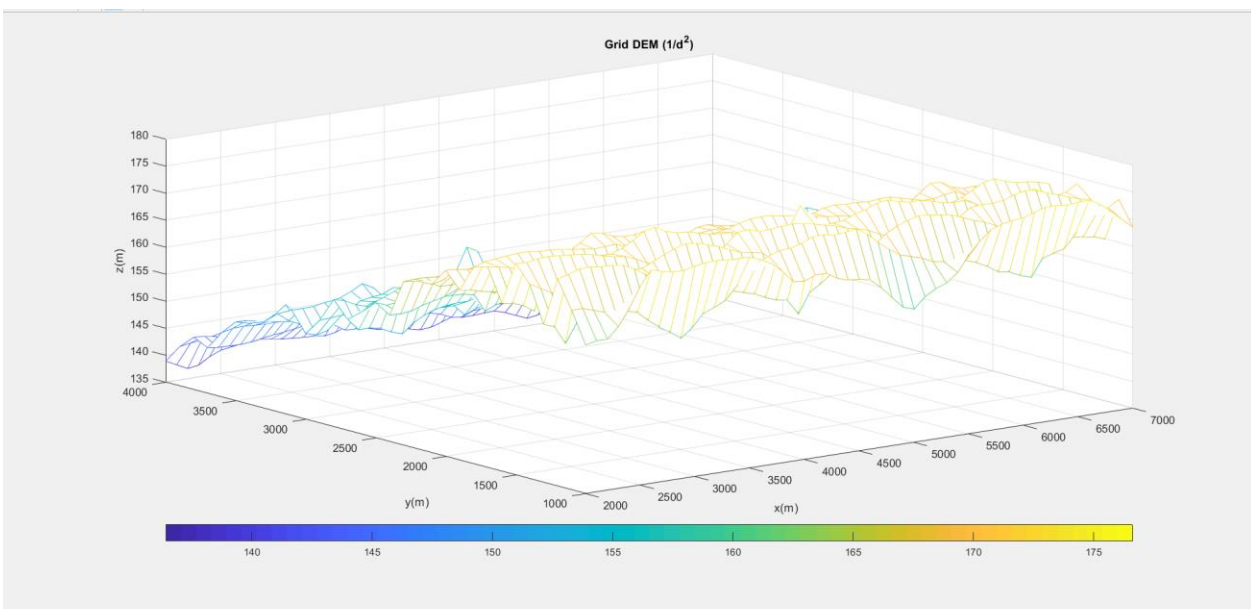

Figure 3: DEM using IDW method and 1/d^2 weight

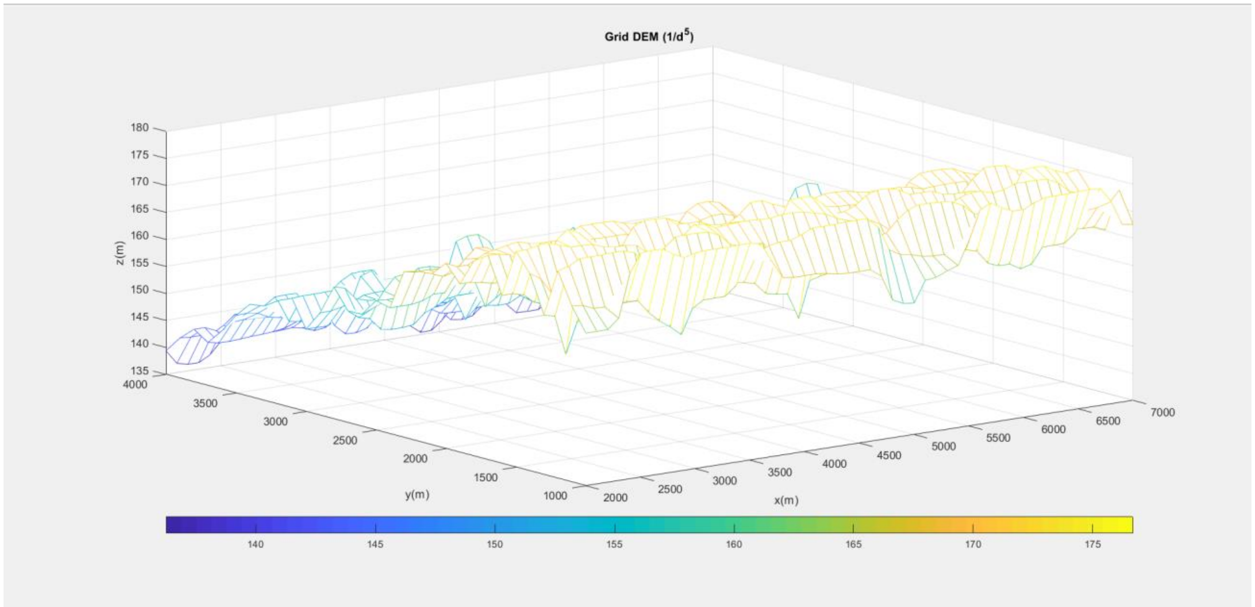

Figure 4: DEM using IDW method and $1/d^5$ weight

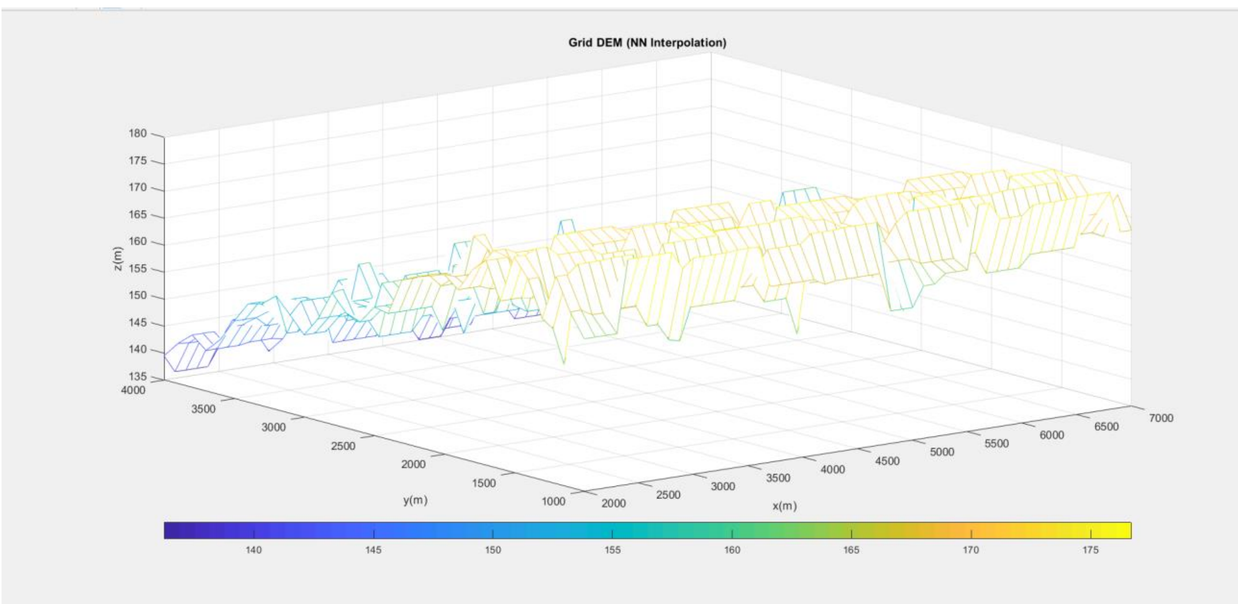

Figure 5: DEM using NN method

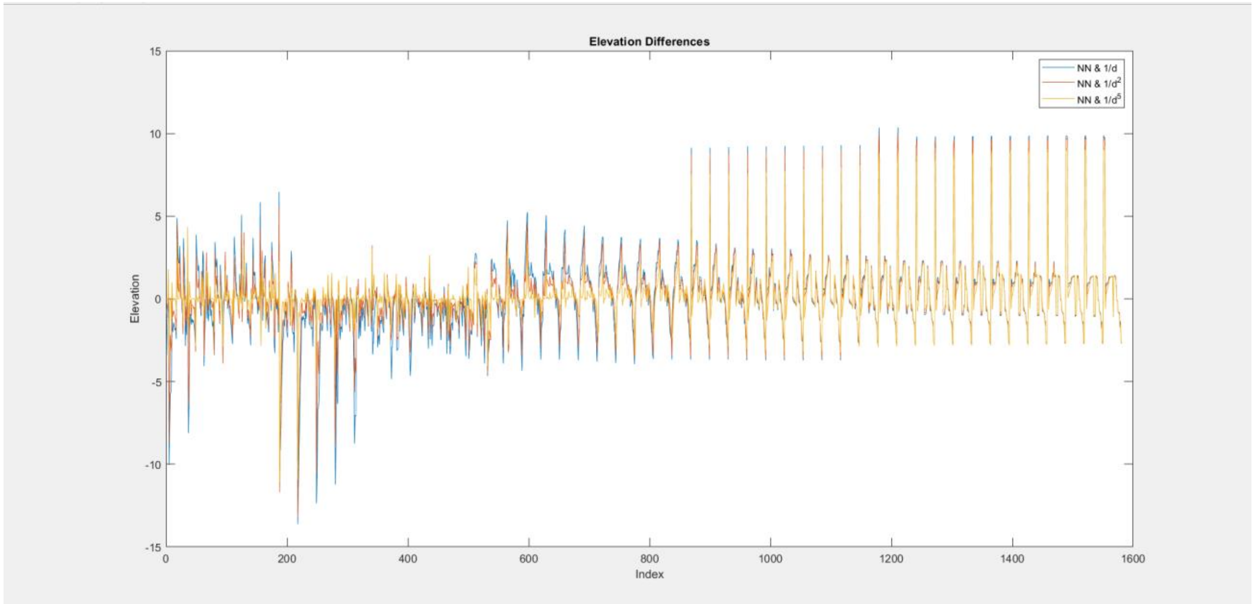

Figure 6: Elevation Differences

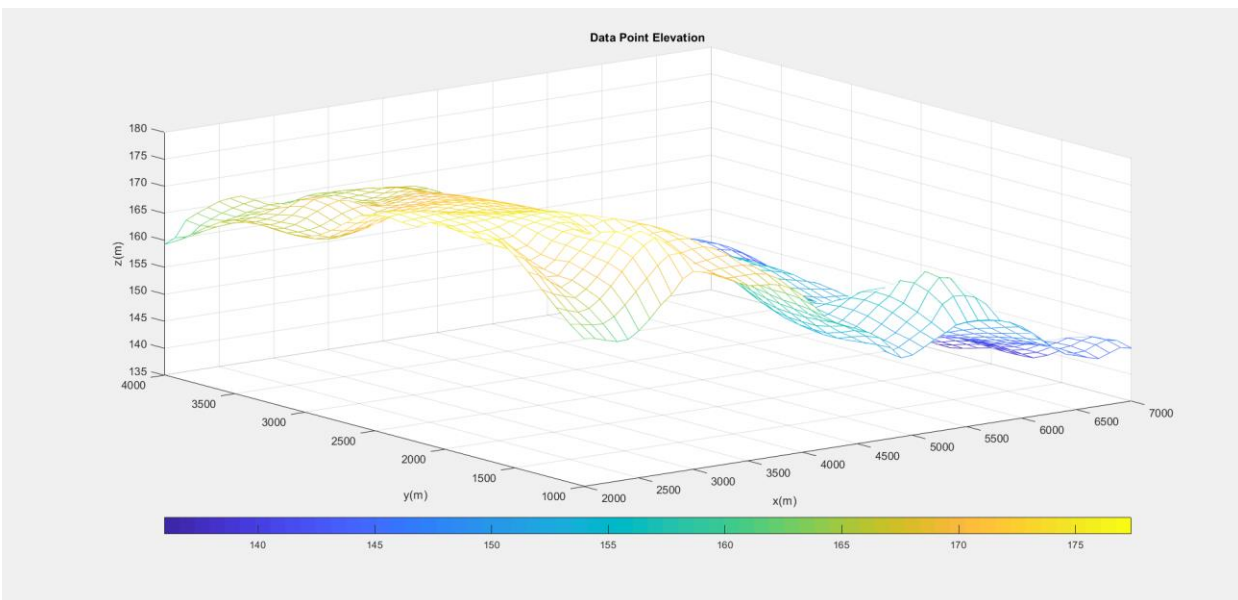

Figure 7: DEM using the Datapoints

Of the three weighting schemes used to generate the DEMs with the third weighting scheme ($\frac{1}{d}$) had the smoothest surface. The DEM created using the NN method has sharp peaks and valleys where as the DEMs generated using the IDW method has smoother peaks and valleys. The DEM created using the first weighting scheme ($\frac{1}{d}$) for the IDW method was the most similar to the DEM generated using the NN method. The IDW method using the third weighting scheme ($\frac{1}{d}$) is the best estimation of the elevation of the grid points because the furthest points have a very small weight and therefore have minimal contribution to the result.

Conclusion

This lab aided in further understanding the application local interpolation methods that were taught in the lectures and the differences in the DEMs that they produced. The results were satisfactory and while the graphs were looked similar to one another, they could be improved further.

References

2021. *ESSE 4640: Lab 4 Manual.*

Jadidi, Mojgan. 2021. "ESSE 4640 Digital Terrain Modelling: Lecture 6." Lecture Slides.

MATHWORKS. Documentation. https://www.mathworks.com

Appendix

```
%% ESSE4640 DTM Lab 4
clc
clear all
format long
ele = [];
ele2 = [];
ele5 = [];
elenn = [];

%% 1. Read Data and Generate a Grid
% read file
%data = xlsread('lab4esse4640');
%%Initial (Given) Coordinates
x = xlsread('lab4esse4640',1,'A2:A200') ;
y = xlsread('lab4esse4640',1,'B2:B200') ;
z = xlsread('lab4esse4640',1,'C2:C200') ;

% Generate a 100m Grid
LLX=2000; LLY=1000; URX=7000; URY=4000;

xgrd = LLX:100:URX; ygrd = LLY:100:URY;

[Xgrd,Ygrd] = meshgrid(xgrd,ygrd);
figure(1)
hold on
Z0 = Xgrd.*0 + Ygrd.*0;
surf(Xgrd,Ygrd,Z0);
scatter(x,y,'R');
legend('Grid','Given points');
title('Grid and Data Points');
xlabel('X(m)');
ylabel('Y(m)');
hold off

%% 2. Elevations at the Grid Point Using IDW
Method
% Weights
% 1/D
xgrd_size = size(xgrd) .* size(ygrd)
for i=1:xgrd_size(2)
  for j= 1:size(x)
    D(j,i) = sqrt((Xgrd(i)-x(j))^2+(Ygrd(i)-y(j))^2);
  end
end
d1 = D;
for i=1:xgrd_size(2)
  for counter = 1:15
    NN = min(d1(:,i));
    idx = find(d1(:,i) == NN);
    num(counter,i)=((1/NN)*z(idx));
    Wgt(counter,i) = (1/NN);
    d1(idx,i) = 100000000;
  end
end

for i=1:xgrd_size(2)
  ele(i) = sum(num(:,i))/sum(Wgt(:,i));
end

d2 = D;
```

```
% Weight 1/D^2
for i=1:xgrd_size(2)
   for counter = 1:15
      NN = min(d2(:,i));
      idx = find(d2(:,i) == NN);
      num2(counter,i)=((1/NN^2)*z(idx));
      Wgt2(counter,i) = (1/NN^2);
      d2(idx,i) = 100000000;
   end
end

for i = 1:xgrd_size(2)
   ele2(i) = sum(num2(:,i))/sum(Wgt2(:,i));
end

d5 = D;
% Weight 1/D^5
for i=1:xgrd_size(2)
   for counter = 1:15
      NN = min(d5(:,i));
      idx = find(d5(:,i) == NN);
      num5(counter,i)=((1/NN^5)*z(idx));
      Wgt5(counter,i) = (1/NN^5);
      d5(idx,i) = 100000000;
   end
end

for i = 1:xgrd_size(2)
   ele5(i) = sum(num5(:,i))/sum(Wgt5(:,i));
end

%% 3. Elevations at the Grid Point Using NN
Method
for i = 1:xgrd_size(2)
   for j = 1:15

   end
end
for i = 1:xgrd_size(2)
    for j= 1:size(x)
       Dnn(i,j) = sqrt((Xgrd(i)-x(j))^2+(Ygrd(i)-
y(j))^2);
    end
    NN = min(Dnn(i,:));
    idx = find(Dnn(i,:) == NN);
    elenn(i) = z(idx);
end

%% 4. Stats
for i = 1:xgrd_size(2)
    dle1(i) = elenn(i)- ele(i);
    RMSE1(i) = (elenn(i)- ele(i))^2;
end
 meanE1= mean(dle1);
 minE1 = min(dle1);
 maxE1 = max(dle1);
 stdeE1 = std(dle1);
 rmseE1 = sqrt(mean(RMSE1));
 rangeE1 = range(dle1);

% NN and 1/D^2
for i = 1:xgrd_size(2)
    dle2(i) = elenn(i)- ele2(i);
    RMSE2(i) = (elenn(i)- ele2(i))^2;
end
 meanE2= mean(dle2);
 minE2 = min(dle2);
 maxE2 = max(dle2);
 stdeE2 = std(dle2);
 rmseE2 = sqrt(mean(RMSE2));
 rangeE2 = range(dle2);

% NN and 1/D^5
for i = 1:xgrd_size(2)
    dle5(i) = elenn(i)- ele5(i);
    RMSE5(i) = (elenn(i)- ele5(i))^2;
end
 meanE5= mean(dle5);
 minE5 = min(dle5);
 maxE5 = max(dle5);
 stdeE5 = std(dle5);
 rmseE5 = sqrt(mean(RMSE5));
 rangeE5 = range(dle5);


%% 5. Graphs
%D1
figure;
zgrd=reshape(ele,51,31).';
Zgrd =
griddata(xgrd,ygrd,zgrd,Xgrd,Ygrd,'cubic');
%figure (2)
mesh(Xgrd,Ygrd,Zgrd);
title('Grid DEM (1/d)')
xlabel('x(m)')
ylabel('y(m)')
```

```
zlabel('z(m)')
colorbar('southoutside')
figure;

%D2
zgrd=reshape(ele2,51,31).';
Zgrd =
griddata(xgrd,ygrd,zgrd,Xgrd,Ygrd,'cubic');
%figure (2)
mesh(Xgrd,Ygrd,Zgrd);
title('Grid DEM (1/d^2)')
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
colorbar('southoutside')
figure;

%D5
zgrd=reshape(ele5,51,31).';
Zgrd =
griddata(xgrd,ygrd,zgrd,Xgrd,Ygrd,'cubic');
%figure (2)
mesh(Xgrd,Ygrd,Zgrd);
title('Grid DEM (1/d^5)')
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
colorbar('southoutside')
figure;

%NN
zgrd=reshape(elenn,51,31).';
Zgrd =
griddata(xgrd,ygrd,zgrd,Xgrd,Ygrd,'cubic');
%figure (2)
mesh(Xgrd,Ygrd,Zgrd);
title('Grid DEM (NN Interpolation)')
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
colorbar('southoutside')
figure;
%% plot Data Points
Zgrd = griddata(x,y,z,Xgrd,Ygrd,'cubic');
mesh(Xgrd,Ygrd,Zgrd);
title('Data Point Elevation')
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
colorbar('southoutside')
figure;
%% Elevation Difference
for i = 1:1581
   index(i) = i;
end
plot(index,(ele-elenn))
hold on
plot(index,(ele2-elenn))
plot(index,(ele5-elenn))
title('Elevation Differences')
xlabel('Index')
ylabel('Elevation')
legend('NN & 1/d', 'NN & 1/d^2','NN & 1/d^5')
```