York University

Surface Interpolation with Kriging and Inverse Distance Weighting

ESSE 4640 Lab #6

December 3, 2018

Daniel Berec
214327878

Table of Contents

Table of Figures

Table of Tables

Introduction

The purpose of this laboratory exercise is to explore the effects of Kriging and inverse distance weighting interpolation. There are three varieties of Kriging that will be explored: simple, ordinary, and universal. These methods are similar in application, however they vary slightly. The reference points were provided alongside the laboratory manual, as outline in ***Table 1***.

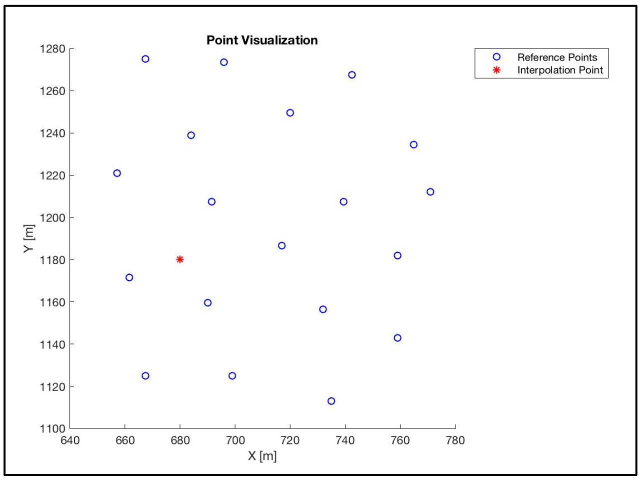

Figure 1. Point visualization.

Point #	X [m]	Y [m]	Z [m]
1	771	1212	697.2
2	735	1113	680.4
3	684	1239	691.2
4	690	1159.5	696
5	759	1182	693.6
6	720	1249.5	702
7	657	1221	697.2
8	739.5	1207.5	694.8
9	696	1273.5	693.6
10	661.5	1171.5	684
11	765	1234.5	702
12	732	1156.5	688.8
13	759	1143	685.2
14	699	1125	690
15	691.5	1207.5	705.6
16	667.5	1275	687.6
17	742.5	1267.5	698.4
18	667.5	1125	702
19	717	1186.5	690

Table 1. Reference point data.

Methodology

Step 1 → Creating the Semivariogram

As Kriging utilizes the relationship between point distances and their semivariances, a semivariogram is required. Firstly, the distances between each pair of points in the reference data was calculated:

$$d_{ij} = \sqrt{\left(X_i - X_j\right)^2 + \left(Y_i - Y_j\right)^2}$$

Second, the semivariances of each point pair was calculated using the formula:

$$\gamma_{ij} = \frac{1}{2}\left(Z_j - Z_i\right)^2$$

To create the semivariogram, the distance for each point pair was plotted against its respective semivariance, yielding the plot shown in ***Figure 2***.

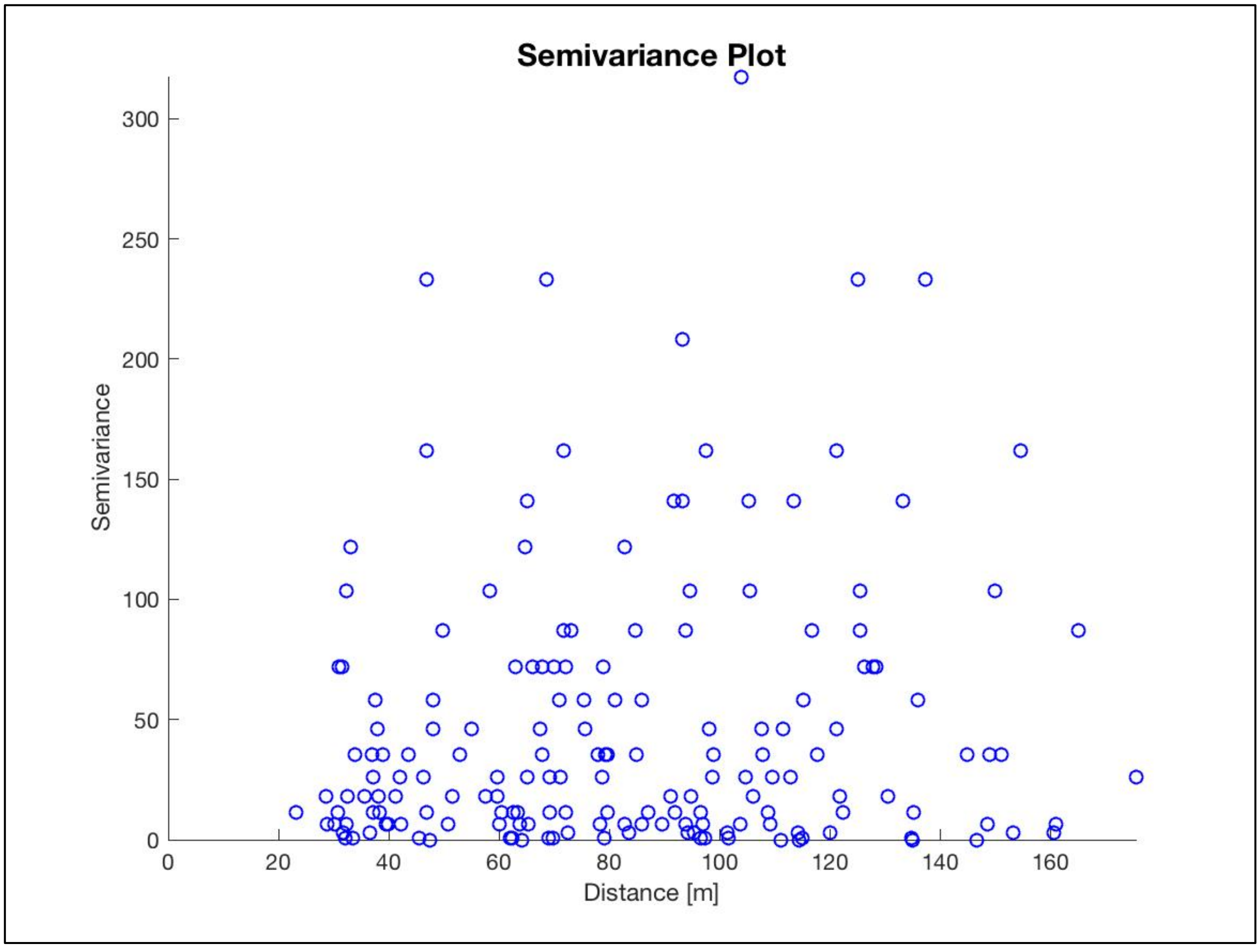

Figure 2. Semivariance plotted against distance for each point pair.

In order to create the semivariogram, the plot was broken down into 12 lags with a lag size of 5. This meant that the semivariogram would only consider the distances that were less than 60 meters. The average semivariance was calculated for each lag, leading to a matrix consisting of 12 points. Since the variogram was to be linear, the points were fit to a linear model and plotted overtop the existing semivariance plot. For the distances greater than 60 meters, the maximum value (the semivariance value at 60m) was continued for the full extent of the plot. This can be explained using the following formulas:

$$\gamma_d = md + n, \{d \epsilon R, 0 < d \leq 60\}$$

$$\gamma_d = \gamma_r, \{d \epsilon R, d > 60\}$$

where d is the distance, r is the range, n is the nugget, and m is the slope. The resulting semivariogram can be seen in ***Figure 2***.

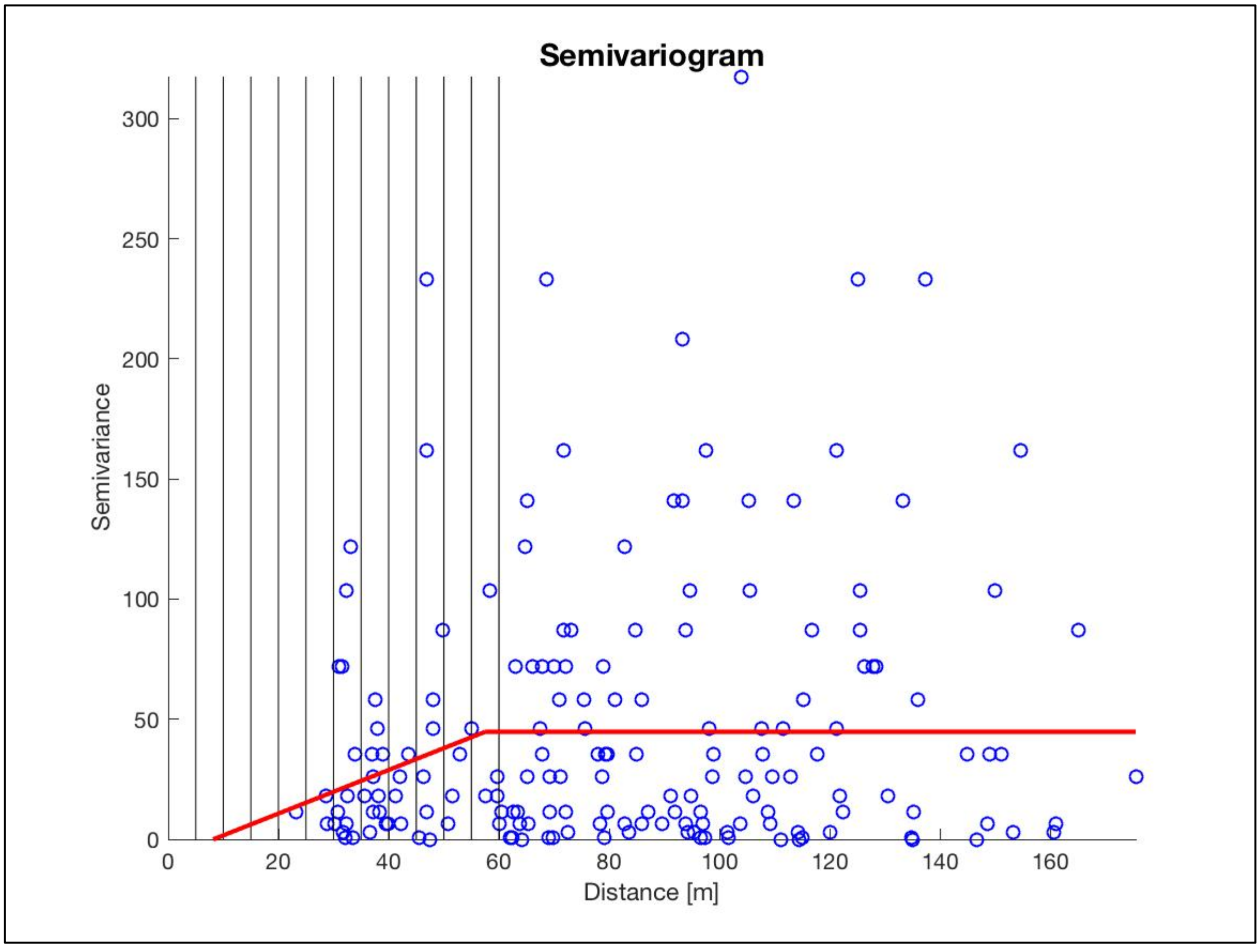

Figure 3. Semivariogram.

Step 2 → Interpolating Semivariance from Semivariogram

In order to determine the semivariance of the points relating to point P (γ_{Pn}), the distance between point P and each reference point was calculated and used in conjunction with the semivariogram to determine which semivariance corresponded to which distance. If the distance was determined to be greater than 60m, the maximum value of the semivariogram was used, as this portion of the semivariogram is horizontally flat (values do not change).

Step 3 → Simple Kriging

Once the semivariances of each (P, n) point pair have been determined, the Kriging process may commence. Firstly, the weighting of each reference point must be determined for use in the calculation of the estimated elevation of the interpolation point, P. This uses the formula:

$$G \cdot w = g \rightarrow \ w = G^{-1} \cdot g$$

where the G matrix is an $n \times n$ symmetric matrix consisting of the semivariances of each reference point pair, the g matrix is a $n \times 1$ matrix consisting of the semivariances γ_{Pi} determined in the previous step, and the w matrix is an $n \times 1$ matrix consisting of the weights of each reference point:

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \gamma_{nn} \end{bmatrix}^{-1} \begin{bmatrix} \gamma_{P1} \\ \gamma_{P2} \\ \vdots \\ \gamma_{Pn} \end{bmatrix}$$

Once the weights have been determined, the elevation of the interpolation point, P, can be calculated by the following:

$$Z_p = w_1 Z_1 + w_2 Z_2 + \cdots + w_n Z_n = \sum w_i Z_i$$

and the accuracy of the estimation can be assessed by considering the variance of the estimation error, which is calculated as:

$$var_e = \sigma_e^2 = w_1 \gamma_{P1} + w_2 \gamma_{P2} + \cdots w_n \gamma_{Pn} = \sum w_i \gamma_{Pi}$$

Step 4 → Ordinary Kriging

Ordinary Kriging follows roughly the same process as simple Kriging, except the condition is imposed that the summation of weights must equal to one:

$$\sum w_i = 1$$

This is achieved by altering the $G, g,$ and w matrices slightly:

$$\begin{bmatrix} w \\ k \end{bmatrix} = \begin{bmatrix} G & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} g \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ k \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} & 1 \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \gamma_{nn} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_{P1} \\ \gamma_{P2} \\ \vdots \\ \gamma_{Pn} \\ 1 \end{bmatrix}$$

where k is the Lagrange coefficient. Since the size of the weight matrix was increased by one, the dimensions of the other two matrices were also increased by adding a row/column of ones to the ends. This modification ensures that the summation of weight will be equal to one. The elevation of the interpolation point can then be calculated as in the simple Kriging method:

$$Z_p = w_1 Z_1 + w_2 Z_2 + \cdots + w_n Z_n = \sum w_i Z_i$$

The variance of the estimation error can also be calculated, this time considering the Lagrange coefficient:

$$var_e = \sigma_e^2 = w_1\gamma_{P1} + w_2\gamma_{P2} + \cdots w_n\gamma_{Pn} + k = \sum w_i\gamma_{Pi} + k$$

Step 5 → Universal Kriging

Similar to the last step, universal Kriging also follows roughly the same process as simple Kriging, with more slight modifications made to the matrices. This includes the addition of the coefficients of the linear trend, as well as the Lagrange coefficient.

$$\begin{bmatrix} w \\ k \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} G & 1 & X_i & Y_i \\ 1 & 0 & 0 & 0 \\ X_i & 0 & 0 & 0 \\ Y_i & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} g \\ 1 \\ X_P \\ Y_P \end{bmatrix} \rightarrow \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ k \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} & 1 & X_1 & Y_1 \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2n} & 1 & X_2 & Y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \gamma_{nn} & 1 & X_n & Y_n \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ X_1 & X_2 & \cdots & X_n & 0 & 0 & 0 \\ Y_1 & Y_2 & \cdots & Y_n & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_{P1} \\ \gamma_{P2} \\ \vdots \\ \gamma_{Pn} \\ 1 \\ X_P \\ Y_P \end{bmatrix}$$

These modifications impose the conditions that:

$$\sum w_i = 1 \qquad \sum w_i X_i = X_P \qquad \sum W_i Y_i = Y_P$$

As in the other cases, the elevation of the interpolation point can be calculated as:

$$Z_p = w_1 Z_1 + w_2 Z_2 + \cdots + w_n Z_n = \sum w_i Z_i$$

and the variance of estimation error can be calculated (this time considering the Lagrange coefficient as well as the linear trend coefficients) as:

$$var_e = \sigma_e^2 = w_1\gamma_{P1} + w_2\gamma_{P2} + \cdots w_n\gamma_{Pn} + k(a_1 X_P + a_2 Y_P) = \sum w_i\gamma_{Pi} + k(a_1 X_P + a_2 Y_P)$$

Step 6 → Inverse Distance Weighting

This method of interpolation was explored in laboratory #5, and therefore will not be explained in excessive detail. The weighting of each reference points is calculated by considering the distance between the reference point and the interpolation point:

$$w = \frac{1}{d^v}$$

where in this case, v was equal to two and five (two IDW methods). The elevation of the interpolation point can then be determined by the following:

$$Z_P = \frac{\sum w_i Z_i}{\sum w_i}$$

Results & Analysis

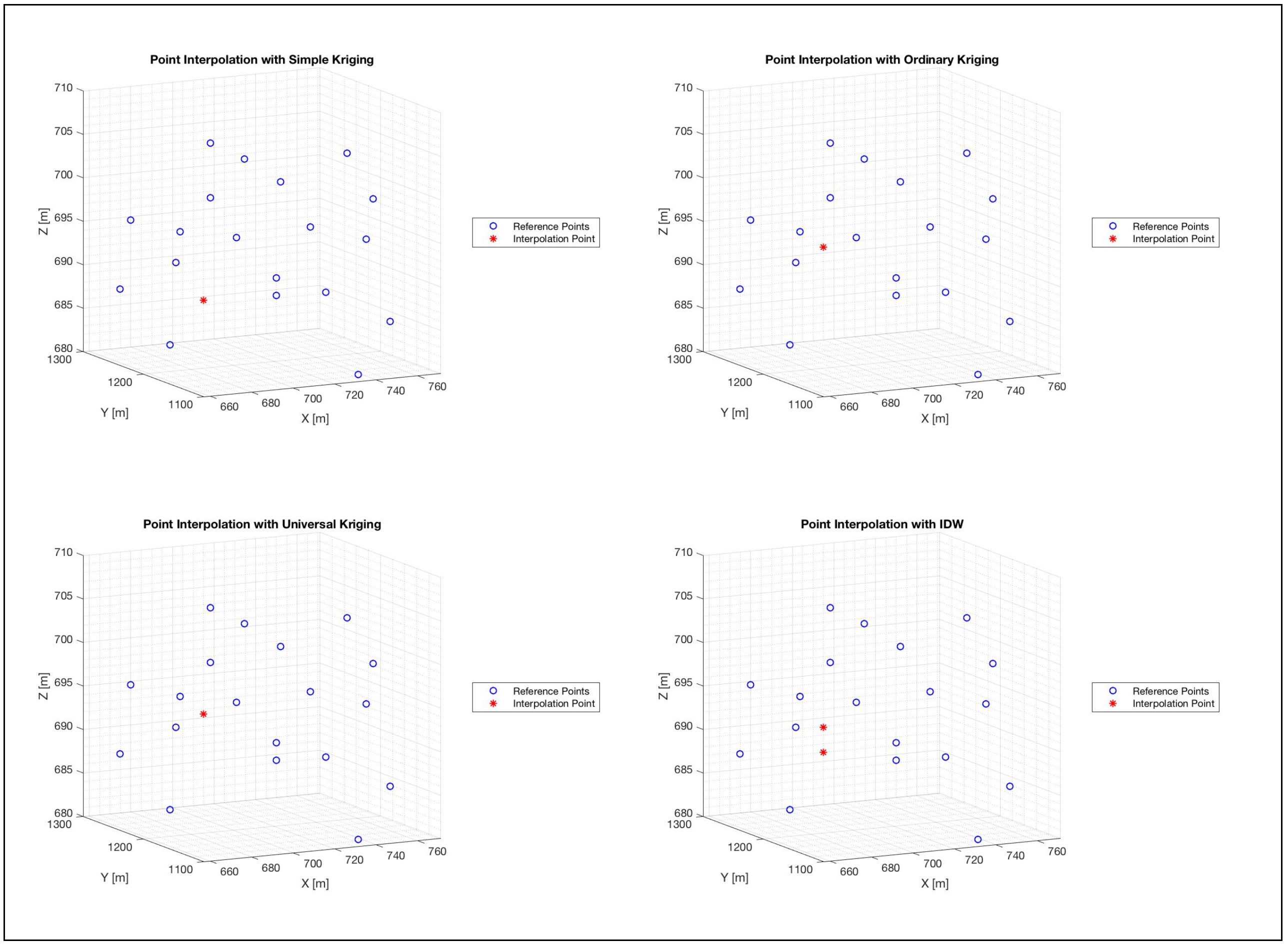

Figure 4. Interpolation results for each interpolation method.

Method	Elevation [m]	Variance Est. Error	Standard Deviation	Sum of Weights
Simple Kriging	688.475	8.956	2.992657682	0.99113
Ordinary Kriging	694.602	8.955	2.992490601	1
Universal Kriging	694.359	103.96	10.19607768	1
IDW 1	692.851	NA	NA	0.00915
IDW 2	689.962	NA	NA	0.000001

Table 2. Interpolation results.

The elevations obtained were roughly as expected. I did not expect the result of simple Kriging to be *that* much different than ordinary and universal Kriging, but it makes sense when analyzing the summation of weights. The summations of weights were pretty much as expected, with ordinary and universal Kriging both equal to one (as the condition states) and simple Kriging close to one. The weights for the IDW 2 method were very small due to the v-coefficient used to calculate them ($v = 5$), whereas the weights for the IDW 1 method were considerably larger ($v = 2$). Method 1 gives higher weighting (relative to method 2) to the reference points that are further from the interpolation point, as method 2 barely considers distant points at all in reference to closer points.

	Simple Kriging	Ordinary Kriging	Universal Kriging	IDW Method 1	IDW Method 2
w_1	0.063933	0.065294	0.060919	0.00010747	1.20E-10
w_2	0.038778	0.039134	0.048908	0.00013308	2.04E-10
w_3	-0.34274	-0.34247	-0.34187	0.00028596	1.38E-09
w_4	0.35926	0.3587	0.3563	0.0019222	1.62E-07
w_5	0.093373	0.093188	0.094169	0.00016013	3.24E-10
w_6	0.20398	0.20476	0.20742	0.00015551	3.02E-10
w_7	0.17001	0.17086	0.17174	0.00045249	4.36E-09
w_8	-0.36841	-0.36893	-0.3732	0.00023275	8.26E-10
w_9	0.034113	0.034112	0.027574	0.00011113	1.30E-10
w_{10}	0.40525	0.40635	0.41316	0.0024125	2.86E-07
w_{11}	0.093088	0.093146	0.090877	9.81E-05	9.53E-11
w_{12}	-0.25049	-0.25098	-0.25301	0.0003071	1.65E-09
w_{13}	0.070542	0.0718	0.073688	0.00013141	1.98E-10
w_{14}	-0.010072	-0.0090306	-0.0046652	0.00029533	1.50E-09
w_{15}	0.32228	0.32184	0.31717	0.0011255	4.25E-08
w_{16}	0.071678	0.072561	0.066096	0.00010892	1.24E-10
w_{17}	-0.15698	-0.15634	-0.1697	8.65E-05	6.96E-11
w_{18}	-0.14866	-0.14817	-0.13623	0.00031434	1.75E-09
w_{19}	0.34221	0.34417	0.35066	0.00070859	1.34E-08
k		-0.35002	10.566		
a_1			-0.0033802		
a_2			-0.007122		

Table 3. Reference point weights and coefficients.

The individual weights of points for the three Kriging methods are very close in comparison, as they should be. Interestingly, the weights of simple and ordinary Kriging are closer in value than the weights of ordinary and universal Kriging. This is likely the cause of the linear trend coefficients that are present in the universal Kriging calculation. The individual weights of the inverse distance weighting method are significantly smaller than the weights of the Kriging method, which was expected given the formulas used. However, I did not expect the weights of the second inverse distance weighting method to be as small as they are.

	Simple	Ordinary	Universal	IDW 1	IDW 2
Simple		6.127	5.884	4.376	1.487
Ordinary	-6.127		-0.243	-1.751	-4.64
Universal	-5.884	0.243		-1.508	-4.397
IDW 1	-4.376	1.751	1.508		-2.889
IDW 2	-1.487	4.64	4.397	2.889	

Table 4. Elevation differences in meters.

The elevations differences make sense, with ordinary and universal being the closest in comparison. The simply Kriging method has the largest difference, closely followed by the second inverse distance weighting method.

Discussion

Referring to the semivariogram in ***Figure 3***, the linear model appears to somewhat fit the semivariance data when the distance is less than 60 meters. There is a noticeable increase in the semivariances as the distance increases from 20 meters to 60 meters, which is echoed in the semivariogram model. As for the distances greater than 60 meters, it looks as though the average semivariance (if it were continued in lags of 5 meters) should decrease as the distance increases, which is not represented in the horizontal line. Overall I feel as though the model is a relatively appropriate representation of the correlation between semivariance and distance.

Conclusion

Kriging has proven to be a robust and accurate method of surface interpolation. The three Kriging methods explored are simple to implement, and provide generally better results than the inverse distance weighting method. This was a fitting exercise to end off the course, as this lab was challenging yet enjoyable nonetheless.

References

Armenakis, C. (2018). *Lecture 6 surface representation: Local surface interpolation using area and point-based methods* [PDF]. LE/ESSE 4640 Digital Terrain Modeling, York University.

Armenakis, C. (2018). *Lecture 7A: Kriging – A geospatial global interpolation method* [PDF]. LE/ESSE 4640 Digital Terrain Modeling, York University.

Armenakis, C. (2018). *Lecture 7B: Simple, ordinary, and universal Kriging* [PDF]. LE/ESSE 4640 Digital Terrain Modeling, York University.

Appendix

Matlab Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   ESSE 4640 Digital Terrain Modeling                    %
%              Lab 6 || Surface Interpolation with Kriging                %
%                       Daniel Berec || 214327878                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear; close all; clc;
format longg

% load data

choice = input(['Select interpolation method:\n(1) Simple Kriging\n(2)
Ordinary Kriging\n'...
    '(3) Universal Kriging\n(4) Inverse Distance Weighting\nSelection: ']);
clc;
if choice == 1
    disp('Selection: Simple Kriging')
    method = 'Simple Kriging';
elseif choice == 2
    disp('Selection: Ordinary Kriging')
    method = 'Ordinary Kriging';
elseif choice == 3
    disp('Selection: Universal Kriging')
    method = 'Universal Kriging';
elseif choice == 4
    disp('Selection: Inverse Distance Weighting')
    method = 'IDW';
else
    disp('Not a valid selection.')
    return
end

data = dlmread('ESSE4640_Lab_6_XYZ_F2018.txt','\t',[1 0 19 3]);
PX = 680; PY = 1180; % interpolation point
X = data(:,2); Y = data(:,3); Z = data(:,4);
numpoints = size(data,1);

figure(); hold on
plot(X,Y,'bo'); plot(PX,PY,'r*');
title('Point Visualization'); xlabel('X [m]'); ylabel('Y [m]');
legend('Reference Points','Interpolation
Point','Location','northeastoutside')
axis square


if choice == 1 || choice == 2 || choice == 3

count = 0;
for i = 1:numpoints
    for j = i+1:numpoints
        count = count+1;
```

```
        distance(count,1) = sqrt((X(i)-X(j))^2+(Y(i)-Y(j))^2);
        semivariance(count,1) = (1/2)*(Z(j)-Z(i))^2;
    end
end
figure()
hold on
plot(distance,semivariance,'bo')
xlabel('Distance [m]'); ylabel('Semivariance');
title('Semivariogram','fontsize',14)
axis([0 max(distance) 0 max(semivariance)])

numlags = 12;
lagsize = 5;
currdist = 0;
prevdist = 0;
numit = 0;
done = 0;
temp1 = -2.5;
while ~done
    currdist = currdist+lagsize;
    rangemin = prevdist; rangemax = currdist;
    tempvar = 0;
    int = 0;
    for i = 1:size(distance,1)
        if distance(i) > rangemin && distance(i) <= rangemax
            int = int+1;
            tempvar(int) = semivariance(i);
        end
    end
    temp1 = temp1+lagsize;
    average_semivar(numit+1,1:2) = [temp1 mean(tempvar)];

    prevdist = currdist;
    numit = numit+1;
    if numit == 12
        done = 1;
    end
    line([rangemax rangemax], ylim,'linewidth',0.5,'color','k')
end
p = polyfit(average_semivar(:,1),average_semivar(:,2),1);
pp = polyval(p,average_semivar(:,1));
plot(average_semivar(:,1),pp,'r','linewidth',2)
flatLine = [average_semivar(end,1) max(distance);
    pp(end) pp(end)];
plot(flatLine(1,:),flatLine(2,:),'r','linewidth',2)

for i = 1:numpoints
    Pdistance(i,1) = sqrt((PX-X(i))^2+(PY-Y(i))^2);
    if Pdistance(i) <= 60
        Ppercent(i,1) = Pdistance(i)/60;
    elseif Pdistance(i) > 60
        Ppercent(i,1) = 1;
    end
    Psemivar(i,1) = Ppercent(i)*pp(end);
end

G = ones(numpoints,numpoints);
```

```
count = 0;
for i = 1:numpoints
    for j = 1:numpoints
        if i ~= j
            count = count+1;
            dist = sqrt((X(j)-X(i))^2+(Y(j)-Y(i))^2);
            if dist <= 60
                next_percent = dist/60;
            elseif dist > 60
                next_percent = 1;
            end
            G(i,j) = next_percent*pp(end);
        end
    end
end

if choice == 2
    G(numpoints+1,:) = 1; G(:,numpoints+1) = 1; G(end,end) = 0;
    Psemivar(numpoints+1) = 1;
elseif choice == 3
    G(numpoints+1,:) = 1; G(:,numpoints+1) = 1; G(end,end) = 0;
    G(numpoints+2,1:numpoints) = X'; G(1:numpoints,numpoints+2) = X;
    G(numpoints+3,1:numpoints) = Y'; G(1:numpoints,numpoints+3) = Y;
    Psemivar(numpoints+1:numpoints+3) = [1 PX PY];
end
w = inv(G)*Psemivar;
ZP = sum(w(1:numpoints).*Z);
summation_of_weights = sum(w(1:numpoints));
if choice == 1
    var_estimation_error = sum(w(1:numpoints).*Psemivar(1:numpoints));
elseif choice == 2
    var_estimation_error = sum(w(1:numpoints).*Psemivar(1:numpoints)) +
w(end);
elseif choice == 3
    var_estimation_error = sum(w(1:numpoints).*Psemivar(1:numpoints)) + ...
        w(end-2)*(w(end-1)*PX+w(end)*PY);
end
fprintf('The estimated elevation at point P is: %.6f\n',ZP)
fprintf('The variance estimation error is: %.6f\n',var_estimation_error)
fprintf('The summation of weights is: %.6f\n',summation_of_weights)

elseif choice == 4 % inverse distance weighting
    for i = 1:numpoints
        distance = sqrt((PX-X(i))^2+(PY-Y(i))^2);
        weights(i,1) = 1/distance^2;
        weights(i,2) = 1/distance^5;
    end
    for i = 1:2
        ZP(i) = sum(weights(:,i).*Z)/sum(weights(:,i));
    end
    fprintf(['The estimated elevation at point P is\nMethod 1 (w=1/d^2):
%f'...
        '\nMethod 2 (w=1/d^5): %f\n'],ZP(1),ZP(2))
    fprintf('The summation of weights is\nMethod 1: %f\nMethod 2:
%f\n',sum(weights(:,1)),sum(weights(:,2)))
end

figure(); hold on
```

```
scatter3(X,Y,Z,'bo')
for i = 1:length(ZP)
    scatter3(PX,PY,ZP(i),'r*')
end
grid on; grid minor
title(sprintf('Point Interpolation with %s',method));
xlabel('X [m]'); ylabel('Y [m]'); zlabel('Z [m]');
legend('Reference Points','Interpolation Point','Location','eastoutside')
axis square
view(-27,11)
```