LAB # 5: Surface interpolation with Kriging and IDW

Submitted by: Ryan Nguyen (216334336)

Submitted to: Amir Nourbakhsh

Submitted on: November 19, 2021

Course Code: ESSE 4640

Methodology

1.1.1) Like previous labs, we will first have to import the .txt file for the data of the points. To find the semi-variance we will first have to find the distance between each unique pair of points except for point P. There are 20 points, the unique number of pairs is: . Which means there are 190 unique pairs and 190 distances to be calculating for.

Equation 1: Distance Equation

With our new set of distances that are below half of the maximum distance, we are to create histograms of our data. In this lab, I have chosen to do 5 bins. With our frequency and bins, we can compute the semi-variance.

Equation 2: Semi-Variance Equation

d is the distance interval (bin)

Where nd is the number of pairs within the distance interval (frequency)

zi and zj are the elevation points associated in computing the distance

1.1.2) This uses a different equation to compute the semi-variogram:

Where m is the slope

d is the point pair distance (in intervals of 5m)

n is the nugget

r is the range

1.1.3) The weight can be computed using the distance equation but **only for the x and y coordinates of the points**. We compute the distance between the points and compute its semi-variance, compute the distance between the point and z and its respective semi-variance, where the semi-variance follows the model in 1.1.2. The weights can be computed by inversing the distance between points matrix.

Equation 3: Weight Computation for Simple Kriging

1.1.4) The weights are used to compute the estimated elevation for point P.

Equation 4: Elevation Computation for Point P using Simple Kriging

1.1.5) The standard deviation and variance are computed using the distances from the points to point P.

Equation 5: Variation and Standard Deviation of Simple Kriging

1.1.6) The summation is the sum of all elements in the weight matrix computed in 1.1.3)

Equation 6: Summation of Weight Matrix of Simple Kriging

1.2.1) The weight matrix is computed differently for ordinary kriging as we add a Lagrange coefficient.

Equation 7: Weight Computation for Ordinary Kriging

1.2.2) The elevation is computed using [Equation 4].

1.2.3) The standard deviation and variance are computed with the addition of the Lagrange coefficient.

Equation 8: Variation and Standard Deviation of Ordinary Kriging

1.3.1) The weight matrix is computed with the addition of coefficients of a linear trend.

Equation 9: Weight Computation for Universal Kriging

1.3.2) The elevation is computed using [Equation 4].

1.3.3) The standard deviation and variance are computed with the addition of the coefficients of a linear trend.

Equation 10: Variation and Standard Deviation of Universal Kriging

2.1) Z of point P will be estimated using a given Inverse Distance Weighting method.

Equation 11: IDW Method for Estimated Elevation

where d is the distance with respect to point P

Results

1.1.1) This is the semi-variogram computed using 5 bin intervals:

Figure 1: Semi-Variogram of the Given Data

Chart, line chart

Description automatically generated

We can identify that the nugget is 1.4555, with a range of 20m, with a range of approximately ~13.6593. The trend of the graph shows that it is increasing until it reaches 15m, after that it turns somewhat constant. This tells us that points with a distance of less than 15m are more correlated to the elevations than those over 15m.

1.1.2) The semi-variance is modelled using the equations which produced a semi-variogram:

Figure 2: Semi-Variogram using Given Equations

Chart, line chart

Description automatically generated

Figure 3: Residuals Between [Figure 1] and [Figure 2]

Chart, line chart

Description automatically generated

Looking at both graphs, they look similar where [Figure 2] is shifted 5m to the left than [Figure 1]. [Figure 2] has a linear rise to 10m as it becomes somewhat constant after that whereas we see a rise to 15m in [Figure 1]. Based on the residuals, the model doesn’t fit the equation well as its residuals is near 0 between distances 12m and 14m of the 0 – 20m range.

1.1.3) Distance matrices can be replicated with code attached. The weight matrix for simple kriging:

w1 = 0.999999999999998

w2 = -4.32648998808764e-16

w3 = 2.11122409866149e-15

w4 = -3.37946401026779e-16

w5 = -2.29143125517059e-16

w6 = 6.83219315157561e-16

w7 = -1.35066875545345e-15

w8 = -1.77269417769273e-16

w9 = -1.58578340235988e-16

w10 = -4.18837226478682e-15

w11 = -3.55330699586796e-16

w12 = 1.10690569398955e-15

w13 = 6.56728846552711e-16

w14 = -2.56591644490218e-15

w15 = 5.96255577232533e-16

w16 = -1.58116225704339e-15

w17 = -1.47424999962187e-16

w18 = -2.08503229819323e-17

w19 = -2.51739086864740e-16

w20 = 2.74013455356341e-16

1.1.4) The estimated Z at Point P: 322.800 m

1.1.5) The standard deviation is: 9.1967, variance is: 3.0326.

1.1.6) The summation of the weight is: 1.

1.1.7) Looking at the weight matrix, all the reference points except for point 1, had a negligible influence on the elevation of Point P. Point 1 has the most impact on the elevation of Point P being almost 1. Which is why its elevation is identical to it.

1.2.1) Distance matrices can be replicated with code attached. The weight matrix for ordinary kriging:

w1 = 1

w2 = -3.05311331771918e-16

w3 = -3.33066907387547e-16

w4 = -1.11022302462516e-16

w5 = -2.56739074444567e-16

w6 = 4.99600361081320e-16

w7 = -2.22044604925031e-15

w8 = -3.93131707743244e-16

w9 = -4.82253126321552e-16

w10 = 4.44089209850063e-16

w11 = -7.21644966006352e-16

w12 = 6.66133814775094e-16

w13 = 0

w14 = 3.33066907387547e-16

w15 = -3.95516952522712e-16

w16 = -7.77156117237610e-16

w17 = 1.11022302462516e-16

w18 = -5.55111512312578e-16

w19 = 7.63278329429795e-16

w20 = -7.21644966006352e-16

k = -1.77635683940025e-15

1.2.2) The estimated Z at Point P: 322.800 m

1.2.3) The standard deviation is: 9.1967, variance is: 3.0326.

1.2.4) The summation of the weight is: 1.

1.2.5) This weight is very similar to the simple kriging method. Point 1 had a significant impact on the elevation of point P and produced identical results in elevation, standard deviation, and variance. The weights this time are a little different as point 13 produced a weight of 0. The Lagrange coefficient posed an insignificant addition to the weight matrix.

1.3.1) Distance matrices can be replicated with code attached. The weight matrix for universal kriging:

w1 = 0.732651578574233

w2 = 0.191498402036087

w3 = -0.225070244992413

w4 = 0.0890043327471675

w5 = 0.192964014520574

w6 = 0.562898549787001

w7 = -1.22270298419111

w8 = -0.836840730313661

w9 = -1.04305416130631

w10 = -0.394870129076069

w11 = 0.805805753638525

w12 = 0.580217396625155

w13 = 1.13346985223019

w14 = 0.715550295874826

w15 = -0.629287739889719

w16 = 0.479615613136528

w17 = -0.720125560903171

w18 = 0.169349404184104

w19 = -0.325562832682583

w20 = 0.744489190000596

k = -743.912850163001

a1 = 0.497166139570249

a2 = 0.713926887971546

1.3.2) The estimated Z at Point P: 313.8668 m

1.3.3) The standard deviation is: -4.662193611565118e+05, variance is: 0.000000000000000e+00 + 6.828025784635788e+02i.

1.3.4) The summation of the weight is: -741.7018.

1.3.5) The weight values are very strange but still produced an elevation that is different than the previous methods and reasonably within expectations for an elevation. The variance produced a value that isn’t a real number! Most of the weighting went into the Lagrange coefficient producing a value under -700. This big influence from the coefficient impacted the standard deviation, and thus, the variance. However, excluding the Lagrange coefficient and linear trends, the weights of the points seem to be very reasonable. There is less weighting on point 1, the elevation of weight 7 and 9 seems to be uninfluential to the elevation of point P with a weighting of ~ -1, and point 13 this time has a bigger influence on the elevation with a weighting of > 1.

2.1) The weight matrix using the IDW method:

w1 = 0.0118231260345233

w2 = 0.00761614623000755

w3 = 0.00436071864643293

w4 = 0.0363768643142953

w5 = 0.0201085863663781

w6 = 0.0257069408740356

w7 = 0.0844594594594613

w8 = 0.0118091639111953

w9 = 0.0140114894213256

w10 = 0.00564174894217204

w11 = 0.0357015351660124

w12 = 0.00856164383561651

w13 = 0.0132100396301188

w14 = 0.0561482313307125

w15 = 0.0379794910748191

w16 = 0.00920979922637686

w17 = 0.0195771339075961

w18 = 0.0136967538693330

w19 = 0.00838574423480084

w20 = 0.00755629439322954

The elevation at point P: 322.2055 m.

2.2) The weights computed here looks the most “normal” out of all the computed weights. No values are negative or against correlation of the elevation of Point P. The values are however, very small. Its sum is 0.4319, not even a full 1! Although, still produced a reasonable elevation.

Task C

This is a bar graph depicting the elevations of the points and the computed elevations for Point P. On the x-axis, there is a gap after point 20, which represents the computed elevations for Point P. x=22: simple kriging, x=23: ordinary kriging, x=24: universal kriging, x=25: IDW method.

Figure 4: Bar Graph of Elevations

Chart, bar chart

Description automatically generated

Figure 5: Bar Graph of Elevation Residuals for Simple Kriging

Chart, bar chart

Description automatically generated

Figure 6: Bar Graph of Elevation Residuals for Ordinary Kriging

Chart, bar chart

Description automatically generated

Figure 7: Bar Graph of Elevation Residuals for Universal Kriging

Chart, bar chart

Description automatically generated

Figure 8: Bar Graph of Elevation Residuals for IDW Method

Chart, bar chart

Description automatically generated

The graphs for simple and ordinary kriging are identical due to them producing the same elevation. Looking at the data, universal kriging had the highest residuals from the rest of the methods. This could be because the weights computed for universal kriging were erratic with weights exceeding over 1 and under -1 (as well as a Lagrange coefficient of -743!).

Resources

Jadidi, M. (2021). Lecture: A Geostatistical Interpolation Method – Kriging II, Week 8. Digital Terrain Modeling. Toronto, Ontario, Canada: York University

MATLAB Code

% Importing Data

t = ESSE4640Lab5XYZF2021;

t = table2array(t);

x = t(:, 2);

y = t(:, 3);

z = t(:, 4);

% Q111

dist = [];

%distance between all points, 20 points so there should be 190 unique pairs

for i=1:length(x)-1

for j=2:length(x)-1

dist(i,j) = distance(x(i),x(j),y(i),y(j),z(i),z(j));

end

j = j+1;

end

%reorganizing matrix so that we get 1 side of distance and replacing 0s

%with nan

dist = triu(dist,1);

dist(dist == 0) = NaN;

distmat = [];

zmat = [];

o = 1;

p = 2;

%FINDING Z ASSOCIATED TO DIST

for i=1:length(x)-1

for j=2:length(x)-1

if ~all(isnan(dist(i,j)))

distmat(o) = dist(i,j);

distmat(p) = dist(i,j);

zmat(o) = z(i);

zmat(p) = z(j);

o = o+2;

p = p+2;

end

end

j = j+1;

end

% sorting from least to greatest dist while keeping same z elevation for

% distance computation, but distance is recorded twice

[distmat,idx]=sort(distmat);

zmat=zmat(idx);

%reordered so that now we have values from 1 to 380 (380/2 is 190)

distmat(2:2:end) = [];

% HISTOGRAM

% histogram(distmat, 5)

% 5 bins: [0-5], [5-10], [10-15], [15-20], [20-25]

bin1 = zmat(1:20); %10 values in this bin

bin2 = zmat(21:146); %63

bin3 = zmat(147:278); %66

bin4 = zmat(279:364); %43

bin5 = zmat(365:380); %8

semi1 = semi(bin1, 10);

semi2 = semi(bin2, 63);

semi3 = semi(bin3, 66);

semi4 = semi(bin4, 43);

semi5 = semi(bin5, 8);

col = [0, 5, 10, 15, 20];

semimat1 = [semi1;semi2;semi3;semi4;semi5];

% figure(1)

% plot(col, semimat1)

% title('1.1.1) Semi-Variogram')

% ylabel('Semi-Variance')

% xlabel('Distance')

% Q112

coefficients = polyfit(col, semimat1, 1);

slope = coefficients(1);

nugget = 1.4555;

semi15 = slope.\*[5 10 15]' + nugget; %no nugget

bin = zmat(279:380);

semi20 = semi(bin, 61);

col = [0; 5; 10; 15; 20];

semimat2 = [semi15; semi20; semi20];

% figure(2)

% plot(col, semimat2)

% title('1.1.2) Semi-Variogram')

% ylabel('Semi-Variance')

% xlabel('Distance')

resid = semimat1-semimat2;

% figure(3)

% plot(col, resid)

% title('1.1.2) Residuals')

% ylabel('Semi-Variance')

% xlabel('Distance')

% Q113

for i=1:length(x)-1

for j=1:length(x)-1

dist(i,j) = distance2(x(i),x(j),y(i),y(j));

model(i,j) = semivariance2(dist(i,j),bin,61,nugget);

end

end

for i=1:length(x)-1

distp(i) = distance2(x(i),x(21),y(i),y(21));

semip(i) = semivariance2(dist(i), bin, 61, nugget);

end

semip = semip';

weight = model^-1\*semip;

% Q114

elevp = 0;

for i = 1:length(x)-1

s = weight(i)\*z(i);

elevp = elevp + s; %ELEVATION IS 322.8

end

% Q115

std = 0;

for i = 1:length(x)-1

s = weight(i)\*distp(i);

std = std + s;

end

var = sqrt(std);

function semivariance = semi(bin, numofp)

semivariance = 0;

for i = 1:length(bin)/2

counter = (1 / (2\*numofp))\*(bin(i\*2-1) - bin(i\*2))^2;

semivariance = counter + semivariance;

end

end

function semivariance = semivariance2(d,b,n,nugget)

semivariance = 0;

if d >= 0 && d < 15

semivariance = 0.5589\*d+nugget;

else

for i = 1:length(b)/2

counter = (1 / (2\*n))\*(b(i\*2-1) - b(i\*2))^2;

semivariance = counter + semivariance;

end

end

end

function distance = distance(x1,x2,y1,y2,z1,z2)

distance = sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2);

end

function distance2 = distance2(x1,x2,y1,y2)

distance2 = sqrt((x2-x1)^2+(y2-y1)^2);

end

% Importing Data

t = ESSE4640Lab5XYZF2021;

t = table2array(t);

x = t(:, 2);

y = t(:, 3);

z = t(:, 4);

%distance between all points, 20 points so there should be 190 unique pairs

for i=1:length(x)-1

for j=2:length(x)-1

dist(i,j) = distance(x(i),x(j),y(i),y(j),z(i),z(j));

end

j = j+1;

end

%reorganizing matrix so that we get 1 side of distance and replacing 0s

%with nan

dist = triu(dist,1);

dist(dist == 0) = NaN;

distmat = [];

zmat = [];

o = 1;

p = 2;

%FINDING Z ASSOCIATED TO DIST

for i=1:length(x)-1

for j=2:length(x)-1

if ~all(isnan(dist(i,j)))

distmat(o) = dist(i,j);

distmat(p) = dist(i,j);

zmat(o) = z(i);

zmat(p) = z(j);

o = o+2;

p = p+2;

end

end

j = j+1;

end

% sorting from least to greatest dist while keeping same z elevation for

% distance computation, but distance is recorded twice

[distmat,idx]=sort(distmat);

zmat=zmat(idx);

nugget = 1.4555;

bin = zmat(279:380);

% Q121

for i=1:length(x)-1

for j=1:length(x)-1

dist(i,j) = distance2(x(i),x(j),y(i),y(j));

model(i,j) = semivariance2(dist(i,j),bin,61,nugget);

end

end

for i=1:length(x)-1

distp(i) = distance2(x(i),x(21),y(i),y(21));

semip(i) = semivariance2(dist(i), bin, 61, nugget);

end

semip = semip';

row = ones(1, 20);

col = ones(20, 1);

col = [col; 0];

model = [model; row];

model = [model col];

semip = [semip; 1];

weight = model^-1\*semip;

% Q122

elevp = 0;

for i = 1:length(x)-1

s = weight(i)\*z(i);

elevp = elevp + s; %ELEVATION IS 322.8

end

% Q123

std = 0;

for i = 1:length(x)-1

s = weight(i)\*distp(i);

std = std + s;

end

std = std + weight(21);

var = sqrt(std);

% Q124

sumweight = sum(weight);

function semivariance = semi(bin, numofp)

semivariance = 0;

for i = 1:length(bin)/2

counter = (1 / (2\*numofp))\*(bin(i\*2-1) - bin(i\*2))^2;

semivariance = counter + semivariance;

end

end

function semivariance = semivariance2(d,b,n,nugget)

semivariance = 0;

if d >= 0 && d < 15

semivariance = 0.5589\*d+nugget;

else

for i = 1:length(b)/2

counter = (1 / (2\*n))\*(b(i\*2-1) - b(i\*2))^2;

semivariance = counter + semivariance;

end

end

end

function distance = distance(x1,x2,y1,y2,z1,z2)

distance = sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2);

end

function distance2 = distance2(x1,x2,y1,y2)

distance2 = sqrt((x2-x1)^2+(y2-y1)^2);

end

% Importing Data

t = ESSE4640Lab5XYZF2021;

t = table2array(t);

x = t(:, 2);

y = t(:, 3);

z = t(:, 4);

%distance between all points, 20 points so there should be 190 unique pairs

for i=1:length(x)-1

for j=2:length(x)-1

dist(i,j) = distance(x(i),x(j),y(i),y(j),z(i),z(j));

end

j = j+1;

end

%reorganizing matrix so that we get 1 side of distance and replacing 0s

%with nan

dist = triu(dist,1);

dist(dist == 0) = NaN;

distmat = [];

zmat = [];

o = 1;

p = 2;

%FINDING Z ASSOCIATED TO DIST

for i=1:length(x)-1

for j=2:length(x)-1

if ~all(isnan(dist(i,j)))

distmat(o) = dist(i,j);

distmat(p) = dist(i,j);

zmat(o) = z(i);

zmat(p) = z(j);

o = o+2;

p = p+2;

end

end

j = j+1;

end

% sorting from least to greatest dist while keeping same z elevation for

% distance computation, but distance is recorded twice

[distmat,idx]=sort(distmat);

zmat=zmat(idx);

nugget = 1.4555;

bin = zmat(279:380);

% Q131

for i=1:length(x)-1

for j=1:length(x)-1

dist(i,j) = distance2(x(i),x(j),y(i),y(j));

model(i,j) = semivariance2(dist(i,j),bin,61,nugget);

end

end

for i=1:length(x)-1

distp(i) = distance2(x(i),x(21),y(i),y(21));

semip(i) = semivariance2(dist(i), bin, 61, nugget);

end

semip = semip';

row = ones(1, 20);

col = ones(20, 1);

col = [col; 0];

model = [model; row];

model = [model col];

for i=1:length(x)-1

adder(1, i) = x(i);

adder(2, i) = y(i);

end

adder2 = adder';

zblock = zeros(2, 1);

zblock2 = zeros(3, 2);

adder = [adder zblock];

adder2 = [adder2; zblock2];

model = [model; adder];

model = [model adder2];

semip = [semip; 1; x(21); y(21)];

weight = model^-1\*semip;

% Q132

elevp = 0;

for i = 1:length(x)-1

s = weight(i)\*z(i);

elevp = elevp + s; %ELEVATION IS 313.8668

end

% Q133

std = 0;

for i = 1:length(x)-1

s = weight(i)\*distp(i);

std = std + s;

end

std = std + weight(21)\*(weight(22)\*x(21)+weight(22)\*y(21));

var = sqrt(std);

% Q 134

sumweight = sum(weight)

function semivariance = semi(bin, numofp)

semivariance = 0;

for i = 1:length(bin)/2

counter = (1 / (2\*numofp))\*(bin(i\*2-1) - bin(i\*2))^2;

semivariance = counter + semivariance;

end

end

function semivariance = semivariance2(d,b,n,nugget)

semivariance = 0;

if d >= 0 && d < 15

semivariance = 0.5589\*d+nugget;

else

for i = 1:length(b)/2

counter = (1 / (2\*n))\*(b(i\*2-1) - b(i\*2))^2;

semivariance = counter + semivariance;

end

end

end

function distance = distance(x1,x2,y1,y2,z1,z2)

distance = sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2);

end

function distance2 = distance2(x1,x2,y1,y2)

distance2 = sqrt((x2-x1)^2+(y2-y1)^2);

end

% Importing Data

t = ESSE4640Lab5XYZF2021;

t = table2array(t);

x = t(:, 2);

y = t(:, 3);

z = t(:, 4);

%distance between all points, 20 points so there should be 190 unique pairs

for i=1:length(x)-1

for j=2:length(x)-1

dist(i,j) = distance(x(i),x(j),y(i),y(j),z(i),z(j));

end

j = j+1;

end

%reorganizing matrix so that we get 1 side of distance and replacing 0s

%with nan

dist = triu(dist,1);

dist(dist == 0) = NaN;

distmat = [];

zmat = [];

o = 1;

p = 2;

%FINDING Z ASSOCIATED TO DIST

for i=1:length(x)-1

for j=2:length(x)-1

if ~all(isnan(dist(i,j)))

distmat(o) = dist(i,j);

distmat(p) = dist(i,j);

zmat(o) = z(i);

zmat(p) = z(j);

o = o+2;

p = p+2;

end

end

j = j+1;

end

% sorting from least to greatest dist while keeping same z elevation for

% distance computation, but distance is recorded twice

[distmat,idx]=sort(distmat);

zmat=zmat(idx);

nugget = 1.4555;

bin = zmat(279:380);

% Q131

for i=1:length(x)-1

for j=1:length(x)-1

dist(i,j) = distance2(x(i),x(j),y(i),y(j));

model(i,j) = semivariance2(dist(i,j),bin,61,nugget);

end

end

for i=1:length(x)-1

distp(i) = distance2(x(i),x(21),y(i),y(21));

semip(i) = semivariance2(dist(i), bin, 61, nugget);

end

semip = semip';

row = ones(1, 20);

col = ones(20, 1);

col = [col; 0];

model = [model; row];

model = [model col];

for i=1:length(x)-1

adder(1, i) = x(i);

adder(2, i) = y(i);

end

adder2 = adder';

zblock = zeros(2, 1);

zblock2 = zeros(3, 2);

adder = [adder zblock];

adder2 = [adder2; zblock2];

model = [model; adder];

model = [model adder2];

semip = [semip; 1; x(21); y(21)];

weight = model^-1\*semip;

% Q132

elevp = 0;

for i = 1:length(x)-1

s = weight(i)\*z(i);

elevp = elevp + s; %ELEVATION IS 313.8668

end

% Q133

std = 0;

for i = 1:length(x)-1

s = weight(i)\*distp(i);

std = std + s;

end

std = std + weight(21)\*(weight(22)\*x(21)+weight(22)\*y(21));

var = sqrt(std);

% Q 134

sumweight = sum(weight)

function semivariance = semi(bin, numofp)

semivariance = 0;

for i = 1:length(bin)/2

counter = (1 / (2\*numofp))\*(bin(i\*2-1) - bin(i\*2))^2;

semivariance = counter + semivariance;

end

end

function semivariance = semivariance2(d,b,n,nugget)

semivariance = 0;

if d >= 0 && d < 15

semivariance = 0.5589\*d+nugget;

else

for i = 1:length(b)/2

counter = (1 / (2\*n))\*(b(i\*2-1) - b(i\*2))^2;

semivariance = counter + semivariance;

end

end

end

function distance = distance(x1,x2,y1,y2,z1,z2)

distance = sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2);

end

function distance2 = distance2(x1,x2,y1,y2)

distance2 = sqrt((x2-x1)^2+(y2-y1)^2);

end