

Geostatistics for Earth Observation

Revealing Celtic Fields near Doorwerth with Ordinary Kriging

24 april 2006

Astrid J.M. Humme – 1065068
Aerospace engineering, TU Delft

Table of contents

1. Introduction	2
2. Variograms	4
<i>2.1 Raw and experimental variogram</i>	<i>4</i>
<i>2.2 Theoretical variograms</i>	<i>5</i>
3. Ordinary Kriging	8
<i>3.1 Covariance functions</i>	<i>8</i>
<i>3.2 Nugget effect</i>	<i>9</i>
<i>3.3 Results of Ordinary Kriging</i>	<i>10</i>
4. Conclusions and recommendations	13
Bibliography	14
Appendix A	15
Appendix B:	16
Appendix C	21
Appendix D	26

1. Introduction

A detailed elevation model of the Netherlands has been acquired with the use of Airborne Laser Altimetry and is managed by AHN ('Actueel Hoogtebestand Nederland'). The heights of a terrain can be measured while flying over the area and transmitting laser beams. These beams are scattered by the ground. A certain amount of the beam will be reflected back to the airplane and from the time span between transmission and receive, the distance to a certain point on the terrain can be calculated.

The absolute elevation relative to NAP is determined with the use of the position and location of the plane at a certain time of interest. The location of the airplane is determined via GPS (Global Positioning System). Next to a GPS in the airplane also one system is located on the ground at a maximum distance of 20 km from the airplane. Other parameters should be taken into account to determine an elevation as accurate as possible; these parameters are the attitude, speed and flight altitude of the airplane and the frequency of the laser pulses.

The accuracy of the laser altimetry technique depends strongly on the amount of vegetation and topography in the area. AHN states that for the accuracy of solid topography (such as roads and parking lots) as well as flat or soft topography (such as beaches and grass-fields) a standard deviation of 15 cm maximum applies, with a systematic error of 5 cm maximum. [1]

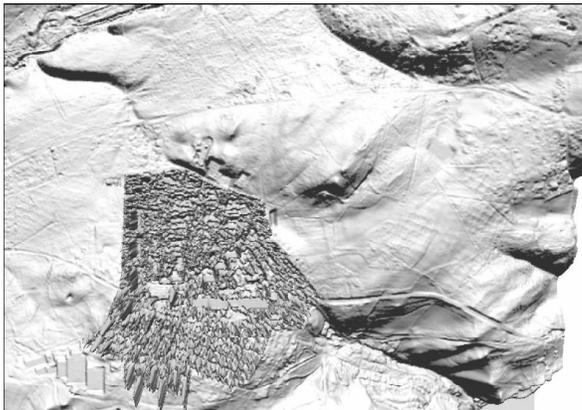


Figure 1-1: Celtic field visualized by Chris Sueur
(http://www.ahn.nl/docs/Sueur_GIN_2005.pdf)

This elevation model can be used for a large amount of applications. Archaeologist Chris Sueur from Vestigia bv, Archaeology and culture history, applied a technique based on illumination to visualise ancient Celtic fields, near Doorwerth, situated in the East of the Netherlands. Figure 1.1 shows the fields exposed by Sueur.

The purpose of this report is to find out whether the visualisation of these fields can be improved with the geostatistic method ordinary kriging. The method uses the available laser elevation data near the Dutch city Doorwerth to estimate a smooth elevation surface through the area of interest. The estimated surface will be subtracted from the dataset and result into the difference between the smooth surface and the dataset, which are small height differences like roads and most probably the Celtic fields.

Figure 1.2 is a plot in which the available heights from the dataset are scattered against the x- and y-coordinate presented in the Dutch 'RijksDriehoeksstelsel'. The colorbar displays the height in meters. During this report the units of the colorbars are in meters.

As can be seen in figure 1.2 when compared to figure 1.1 the fields are assumed to be at the northeast. Therefore a large part can be disposed from the original dataset; a

crop is made from the area of interest to reduce the amount of data points to make processing possible and where possible less time consuming. Figure 1.3 presents the 2-dimensional plot of the crop where the Celtic fields are situated. The total dataset contains 428.634 positions with corresponding heights and the crop contains 103.699 positions and heights. All calculations discussed are performed in the program Matlab.

This report will first discuss the selection of the theoretical model used for ordinary kriging based on the experimental variogram. Chapter 3 explains the concept of ordinary kriging and the results after processing the AHN dataset. The last chapter will give the conclusions and recommendations.

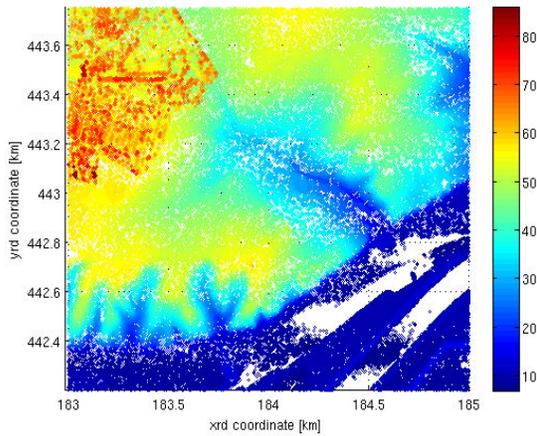


Figure 1-2: AHN dataset Doorwerth scattered

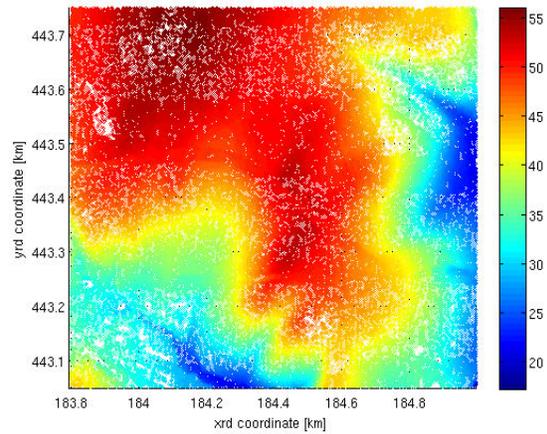


Figure 1-3: Crop of AHN dataset scattered

2. Variograms

A variogram characterizes the spatial variation in a dataset. Several variograms exist: the raw, experimental and theoretical variogram. The raw and experimental variograms are derived from the dataset and the theoretical variograms model the dataset. The linear, exponential, Gaussian and spherical models are the most often chosen theoretical variograms.

A model is needed when Ordinary Kriging is performed to define the parameters: variance, range and nugget. This model is the theoretical variogram, which resembles the experimental variogram best. [2]

2.1 Raw and experimental variogram

In order to construct the raw variogram, see the point cloud in figure 2.1, first the distances between all the points in the dataset have to be calculated:

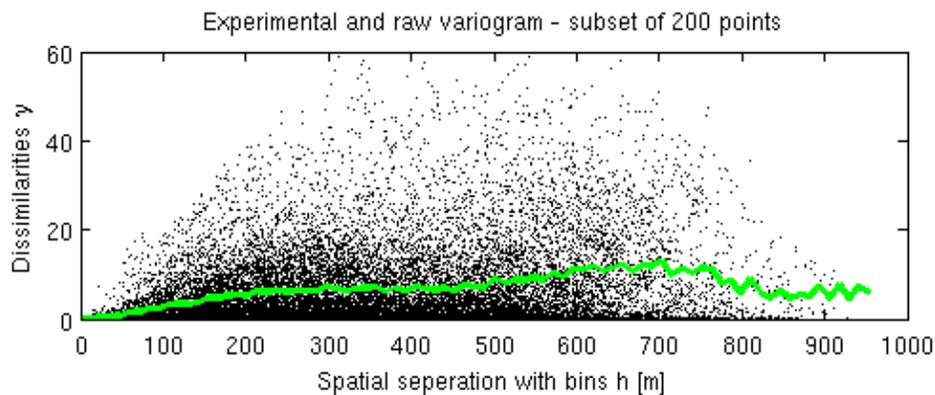
$$h_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2.1)$$

In which h_{ij} is the distance from point i to point j , x_i is the x-coordinate of point i and y_i is the y-coordinate of point i , the same holds for j .

On the y-axis the dissimilarities, γ , are displayed. These are the variances of the height, z :

$$\gamma_{ij} = \frac{(z_i - z_j)^2}{2} \quad (2.2)$$

To construct the experimental variogram, the distances calculated are grouped and averaged. A certain spatial separation h is chosen, which is the size of one bin in which the distances are grouped. Every bin has one dissimilarity value calculated by averaging all the dissimilarities of the distances that belong to that certain bin. The experimental variogram is a graph composed out of these average dissimilarities and can be found for the Doorwerth data in figure 2.1.



Figuur 2-1: Raw variogram (point cloud) and experimental variogram (graph).

Figure 2.1 is based on a crop of 200 points, distributed over the total area. For the calculation of the experimental variogram 200 points is not much compared to the 103.699 points available. Therefore a larger subset has to be chosen, which better represent the full dataset. The amount of points is limited by the calculation capacity of the used computer. The variograms computed with 200 points are only used for visualization of the raw variogram.

To be able to compare the differences after ordinary kriging, two dataset are used during this research: one of 700 and one of 4000 points. Ordinary kriging is used for estimation of a smooth surface based on the AHN dataset, it is expected that when the dataset is large, the estimated surface will resemble the measured surface better and therefore will give less clear height differences after subtraction of the estimated and the measured surfaces.

The points are chosen above the 45 m height, else the variogram would be influenced too much by the slopes of the ‘cliffs’ near Doorwerth and because it is expected that the fields are not situated on the slopes, the heights can be removed without consequences. Not removing these heights will result in a variogram, which does not average out, but will keep increasing. Fitting a reasonable variogram then can be hard when not impossible. The distances between these points are grouped into 80 bins, which results into one dissimilarity for every 12.5 m of spatial distance. A large amount of bins is not required, because a smooth surface has to be created with ordinary kriging and not a surface that resembles the dataset exactly.

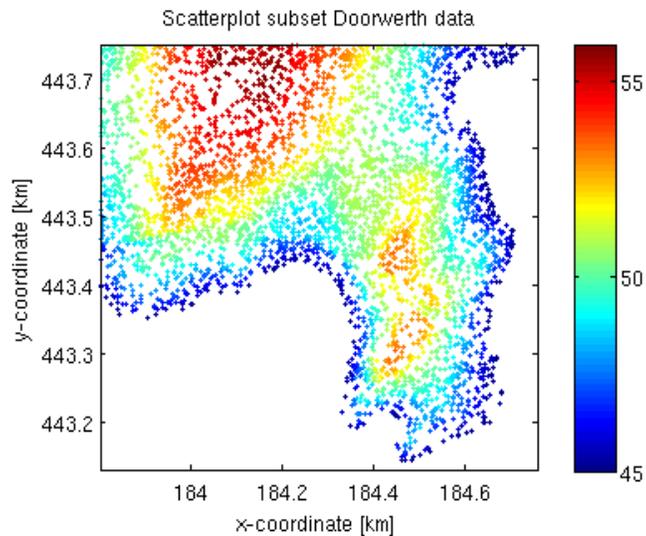


Figure 2-2: Scatterplot Doorwerth data - 4000 points above 45 m (NAP)

Figure 2.2 shows the scatterplot of the heights when subtracting the heights below 45 meter and using 4000 points.

2.2 Theoretical variograms

Now the best fitting theoretical variogram with respect to the experimental variogram is calculated. This variogram will be used for the ordinary kriging step. In Appendix B the Matlab code for determining the variograms can be found.

In a variogram a sill and range can be specified. The sill is the dissimilarity the variogram finally reaches. The range is the distance at which the sill is reached. The programs used to determine the best fitting theoretical variogram, calculate and return the maximal variance, σ^2 , and range, R . The maximal variance depends on the sill and the nugget. In which the nugget is, in this case, chosen to be the dissimilarity at 0 m spatial separation, which is most of the time the measurement inaccuracy. In the case of a subset of 4000 points the nugget is 7,17 cm. This nugget is based on the smallest

dissimilarity obtained from the experimental variogram and also corresponds to the measurement error specified by AHN.

Several theoretical variograms can be used to model the experimental variogram, see figure 2.3. The linear variogram can best be chosen when the experimental variogram does not reach a sill, i.e. is not stationary, this variant is not presented in figure 2.3.

The exponential variogram can best be used when the variogram reaches a sill, but curves all the way up. The spherical variogram can best be used when the experimental variogram starts out straight, bends over sharply and levels out. And the last variogram, the Gaussian variogram can be used when the experimental variogram starts out slow, then almost acts like the exponential variogram and bends a bit slower than it. This model also levels out.

The following formulas represent the different theoretical variograms: [2]

$$\gamma_{\text{expo}}(h) = \sigma^2 \left(1 - e^{-\frac{3h}{R}} \right) \quad (2.3)$$

$$\gamma_{\text{gauss}}(h) = \sigma^2 \left(1 - e^{-\frac{(3h)^2}{R^2}} \right) \quad (2.4)$$

$$\gamma_{\text{sph}}(h) = \sigma^2 \left(\frac{3h}{2R} - \frac{h^3}{2R^3} \right) \quad (2.5)$$

With a nugget of 7,17 cm the best fitting model will be the Gaussian model, see figure 2.3. Returned values for the variance and range were:

Model	Variance	Range [m]
Exponential	8.879	667
Gaussian	8.290	625
Spherical	8.500	650

Table 2.1: Variance and range for the three models required from a subset of 4000 points

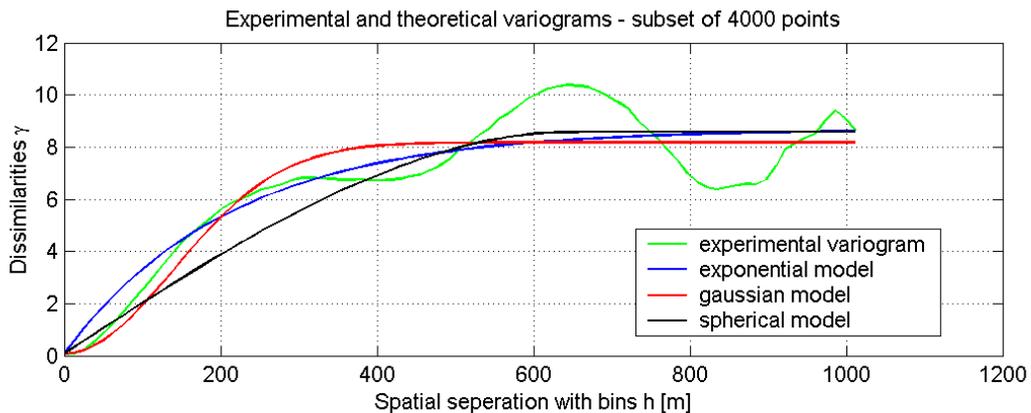


Figure 2-3: Experimental and theoretical variograms for the Doorwerth dataset

The program for best fitting spherical variogram does not compute the variogram as it should. When the sill is reached, the rest of the dissimilarities should stay at that

value. Therefore a function is used which does not calculate the best fitting model, but calculates the spherical model with the use of the sill and range specified by the user. From examining the experimental variogram the sill and range can be estimated. But clear is that even when the best fitting sill and range are chosen, still the Gaussian function corresponds the dataset best and therefore will be chosen as theoretical model for ordinary kriging.

As discussed in section 2.1 two dataset will be used for the visualization of the Celtic fields, in order to be able to compare the results. The variance and range for the best fitting theoretical models based on a subset of 700 points out of the total AHN dataset can be found in table 2.2. The nugget resulted into 4,70 cm.

Model	Variance	Range [m]
Exponential	8.227	625
Gaussian	7.698	556
Spherical	8.000	600

Table 2.2: Variance and range for the three models required from a subset of 700 points

3. Ordinary Kriging

Kriging is used to define a relatively smooth surface based on the chosen theoretical models. In chapter 2 the best fitting model is defined, which is the Gaussian, and variance, range and nugget are calculated. Ordinary kriging requires a covariance function. In this case the exponential, Gaussian and spherical covariance functions are used, and the parameters calculated in chapter 2 are inserted into these functions to get best results. Even though know is that the Gaussian function models the dataset best, three covariance functions are used to compare the results.

The calculated surface is subtracted from the available dataset consisting of the laser measurements and results into height residuals. When displaying these residuals the Celtic fields are expected to be seen.

Ordinary Kriging respects the measurements; the method assumes measurements are exact. Unfortunately they are not. The nugget expresses this error. When this nugget is filtered from the redundancy matrix, ordinary kriging will not respect the measurements anymore.

3.1 Covariance functions

Ordinary Kriging uses the covariance function, composed out of the parameters of the best fitting models determined in chapter 2, to adjust weights to the surrounding data points. Multiplying these weights, w_i , with the heights of those points, h_i , and adding the weighted heights, results into one value for the height, \hat{h}_0 , at the observed x and y coordinates, see formula 3.1. [3]

$$\hat{h}_0 = \sum_{i=1}^n w_i \cdot h_i \quad (3.1)$$

The weights are calculated by multiplying the inverse of the redundancy matrix, C_n^{-1} , with the proximity vector, d_n .

$$w_n = C_n^{-1} \cdot d_n \quad (3.2)$$

$$C_n = \begin{pmatrix} C_{11} & \cdots & C_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C_{n1} & \cdots & C_{nn} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}, \quad d_n = \begin{pmatrix} C_{10} \\ \vdots \\ C_{n0} \\ 1 \end{pmatrix}, \quad w_n = \begin{pmatrix} w_1 \\ \vdots \\ w_n \\ \lambda \end{pmatrix}$$

$$C_{ij} = Cov(\|p_i - p_j\|)$$

In which λ is the Lagrange multiplier and p_i denotes the x,y coordinates of the i -th point.

The value for C_{ij} when $i = j$, results into zero spatial distance and therefore only contains the variance.

The redundancy matrix and proximity vector are compiled with the use of covariance functions of each of the three models; experimental, Gaussian and spherical model.

$$COV_{\text{exp}_o}(h) = \sigma^2 e^{-\frac{3h}{R}} \quad (3.3)$$

$$COV_{\text{gauss}}(h) = \sigma^2 e^{-\frac{(3h)^2}{R^2}} \quad (3.4)$$

$$COV_{\text{spher}}(h) = \begin{cases} \sigma^2 \left(1 - \frac{3h}{2R} + \frac{h^3}{2R^3} \right), & h \leq R \\ 0 & h > R \end{cases} \quad (3.5)$$

Figure 3.1 shows the graphs of these covariance functions.

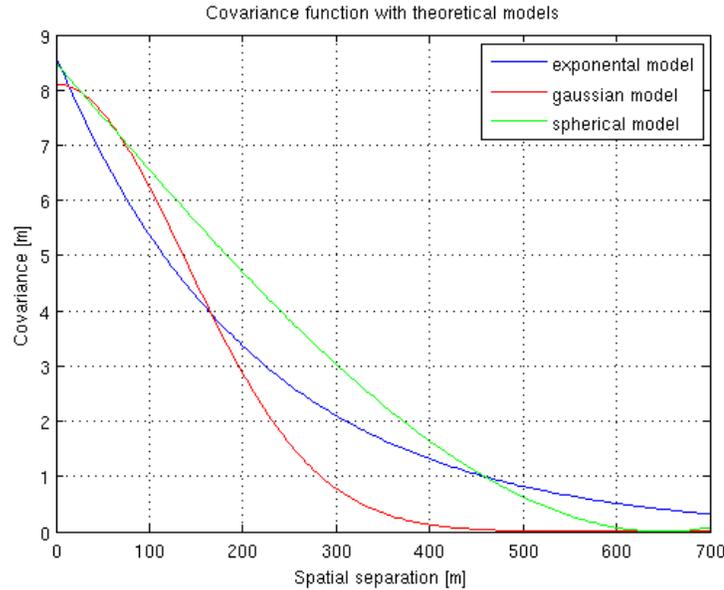


Figure 3-1: Theoretical covariance functions

3.2 Nugget effect

Still the nugget, that only effects the covariance function on the spatial distance of 0, has to be taken into account. Therefore addition of the nugget to the case where the covariance is calculated for the situations where $i = j$ in equation 3.2 is necessary. Like said in the introduction, ordinary kriging respects the measurements and therefore will return exactly the same heights at an observation location as the observation value itself. To visualize the Celtic fields, thus creating a smooth surface and subtracting it, it is not desirable to have residuals equal to zero. After filtering the nugget effect, the system can not respect the measurements. [3]

Filtering the nugget means that the nugget effect will still be present in the redundancy matrix, but is eliminated in the proximity vector. Formula 3.6 represents this filtering.

$$\begin{pmatrix} C_{11} + \text{nugget} & \cdots & C_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C_{n1} & \cdots & C_{nn} + \text{nugget} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} C_{10} \\ \vdots \\ C_{n0} \\ 1 \end{pmatrix} = \begin{pmatrix} w_1 \\ \vdots \\ w_n \\ \lambda \end{pmatrix} \quad (3.6)$$

3.3 Results of Ordinary Kriging

The best fitting model on the experimental variogram was the Gaussian model. To be able to compare differences also the exponential and spherical models are used for the calculation of a smooth surface based on the dataset from AHN and every model had its own range and variance. Also for comparison two different cases were observed. One case in which the range, nugget and variances were based on a subset of 700 points and one in which these parameters were based on a subset of 4000 points. This subset was randomly picked from the available AHN dataset of the region near Doorwerth.

After some trial and error, it became clear that the amount of nearest surrounding points that got a significant weight for the exponential and spherical model was 80 and for the Gaussian model this resulted into 175 surrounding points. It is obvious that the Gaussian model requires more surrounding points when keeping in mind that the variogram model starts out very slow, see figure 2.3.

These boundaries are chosen as such; because the summed weights assigned to the heights of the positions further away then the closest 175 (or 80 positions) is smaller then 0.001. This seems a reasonable number when comparing the percentage and the reaction of the points further then this boundary. The weights assigned to points further away (for the Gaussian function were) so small that many points then had to be selected, but all only got a weight of at most 1 promille.

The models used all the measurements available in the elevation dataset after cropping and eliminating the heights below 45 meter. The output of the ordinary

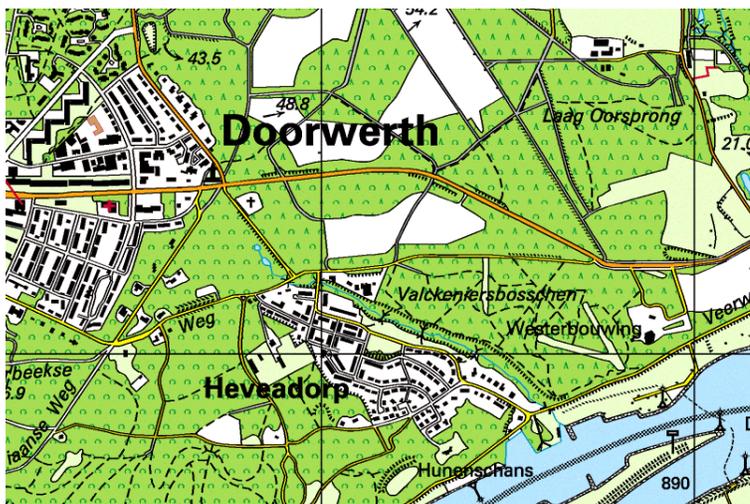


Figure 3-2: Topographic map of the Doorwerth surroundings

kriging are the heights for specified x and y coordinates, and to make life easy these coordinates were chosen to be the same as the coordinates of the dataset. Subtraction of the model from the dataset will be less difficult because no interpolation is needed to subtract.

The topographic map of the area near Doorwerth can be

found in figure 3.2. This map will be compared with the residual heights, which are the heights from the dataset minus the modeled heights, to find the Celtic fields.

The two scatterplots of the Gaussian model, one based on a subset of 700 points and one on 4000 points, can be found in figure 3.3 and 3.4. The other exponential and spherical scatter plots can be found in appendix A. These clearly did not result into the Celtic fields. As can be seen in figure A.1, the exponential plot based on a subset of 700 points did not result into anything and the other three plots only resulted into the roads present nowadays.

In appendix A also the interpolated surfaces can be found. These surfaces are created because they visualize the heights a bit better. All the calculations are done in the program Matlab. Information about the different interpolation methods, linear, cubic and nearest neighbor can be found in the help function of the program. [4] The interpolation presenting the Celtic fields best, are displayed in the appendix.

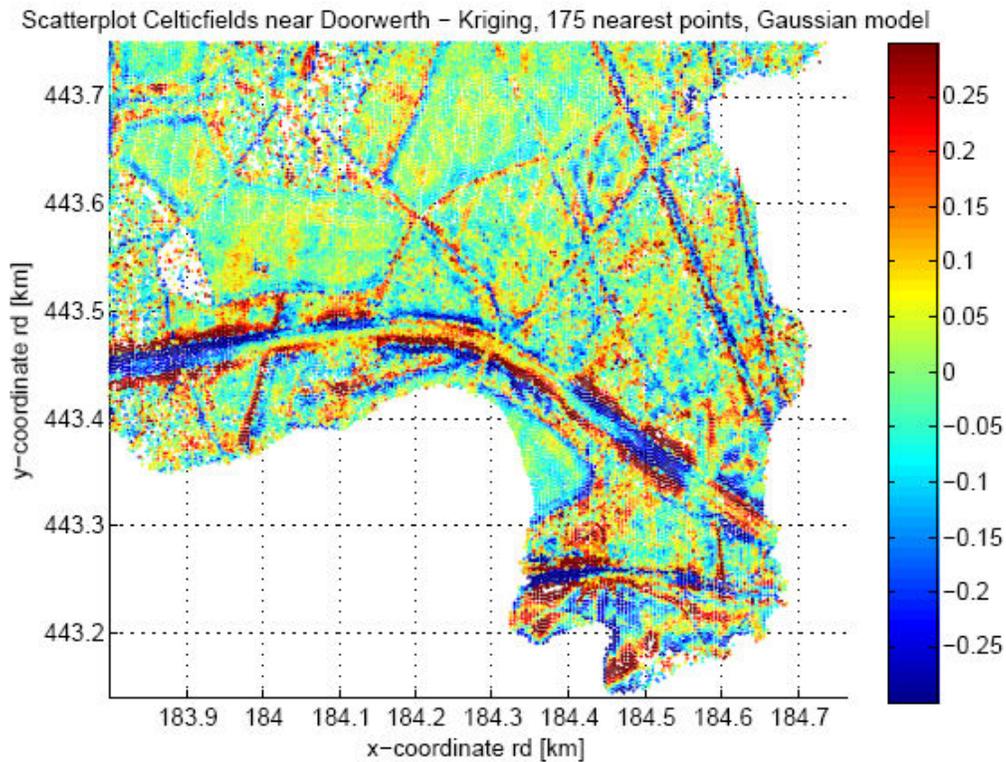


Figure 3-3 Scatterplot Celtic fields, Gaussian model, based on the subset of 700 points.

As can be seen from the figures 3.3 and 3.4 the Celtic fields can be distinguished in the scatterplots. The red and blue lines are mostly roads (see figure 3.2) and the yellow blocks are the Celtic fields. These fields are also present in the south of the area, which were not displayed in the figure of Chris Sueur. Therefore it can be concluded that the use of ordinary kriging indeed improves the visualization of the fields.

Also can be seen that the scatterplot made with the Gaussian model based on the parameters of the 700 points subset gives better results than the scatterplot based on

the subset of 4000 points. This can be due to the fact that the model which uses the parameters based on the subset of 4000 points models the surface too well and therefore does create a surface which is not smooth enough to clearly distinguish the Celtic fields e.g. in the south below the main road.

Weird though is that the exponential model based on the subset of 700 points is worse than the model based on the subset of 4000 points. (Appendix A)

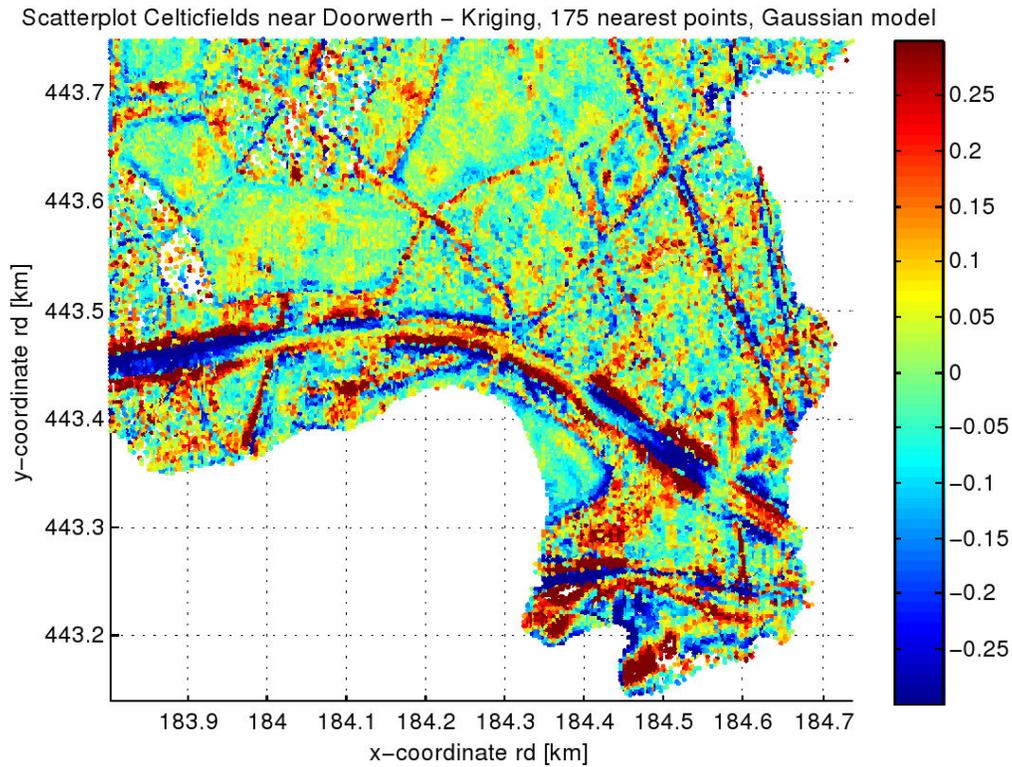


Figure 3-4: Scatterplot Celtic fields, Gaussian model, based on the subset of 4000 points.

4. Conclusions and recommendations

After comparison of the experimental variogram with the theoretical variograms, the Gaussian model best fits the experimental variogram. This model is used for the ordinary kriging step. For comparison also the spherical and exponential models are processed, all three for a variance, nugget and range derived from a subset of 700 and one of 4000 data points.

The Gaussian model resulted into a clear visualisation of the Celtic fields, both for the models build up with the range, sill and nugget found from best fitting Gaussian variogram of the subset of 700 and 4000 points derived from the AHN elevation dataset. It can be seen that the Celtic fields are better visualized in the scatter plot based on the best fitting parameters for the 700 points subset, probably because the surface is smoother for this model.

The spherical and exponential do not result in clear Celtic fields.

Recommended is a research for better visualization. Smaller scales can be used on the z-axis to distinguish smaller differences and/or parameters based on smaller datasets can be used.

Bibliography

1. AHN (Actueel Hoogtebestand Nederland), www.ahn.nl, last visited at 05-04-2006.
2. P.K. Kitanidis , *Introduction to Geostatistics*, applications in hydrogeology, Cambridge University Press, 1997
3. R. Lindenberg, *Lecture sheets 2005, AE4-E04, Geostatistics for Earth Observation*.
4. Matlab website, www.mathworks.com, last visited at 24-04-2006

Appendix A

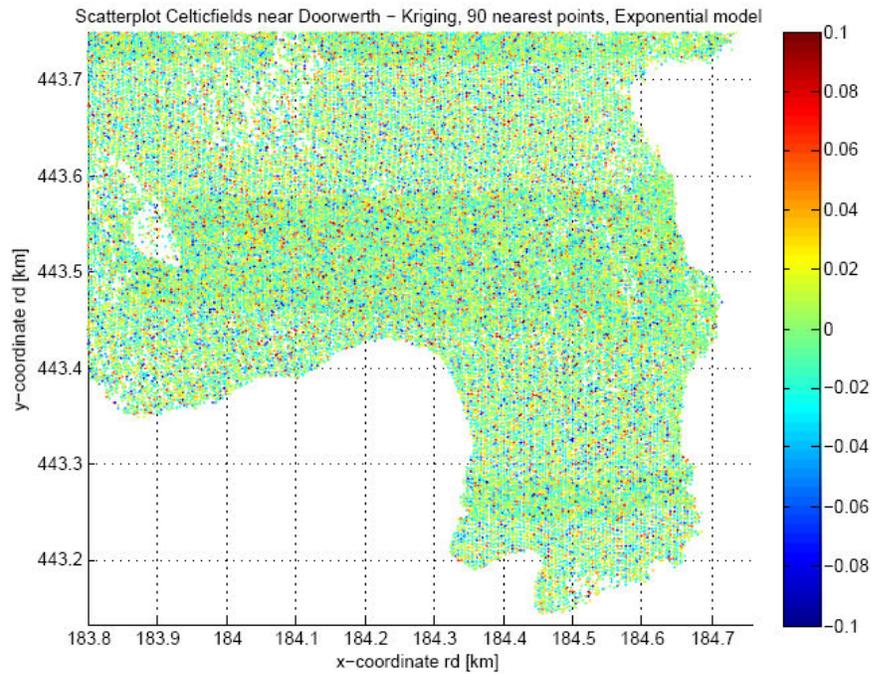


Figure A-1: Scatterplot Celtic fields, exponential model, based on the subset of 700 points.

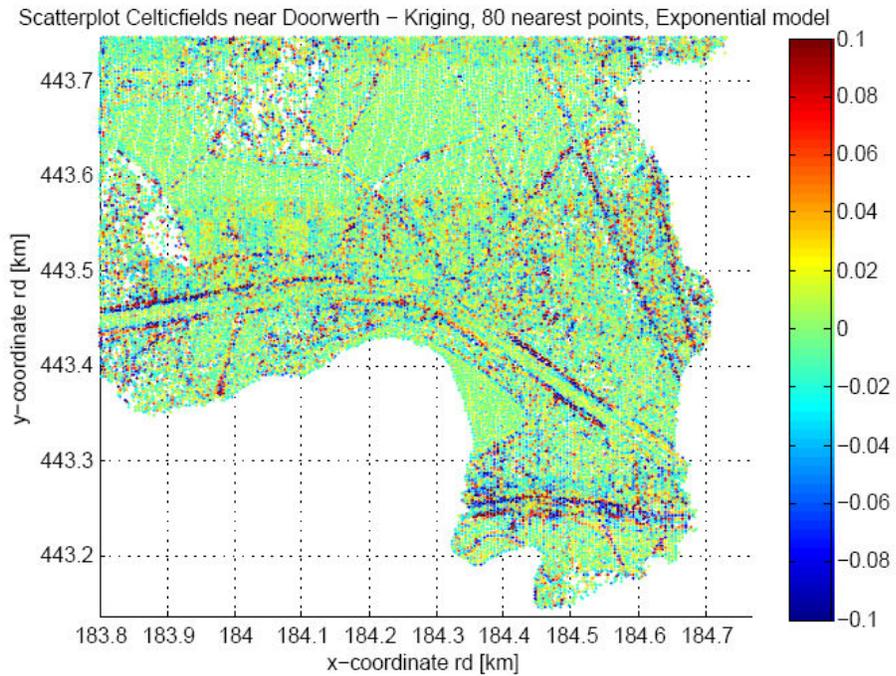


Figure A-2: Scatterplot Celtic fields, exponential model, based on the subset of 4000 points

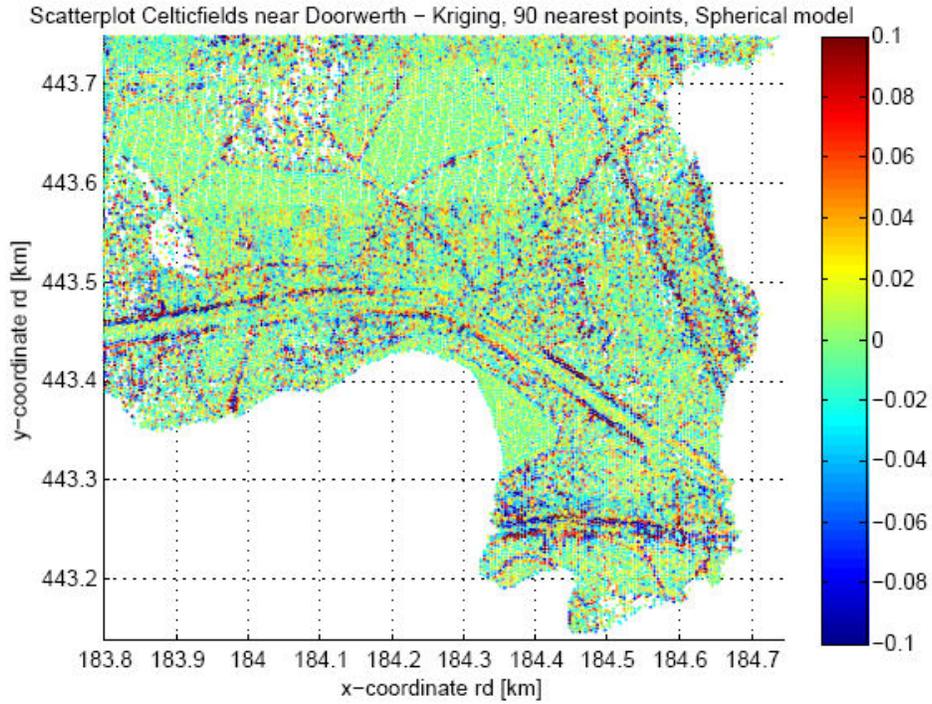


Figure A-3: Scatterplot Celtic fields, spherical model, based on the subset of 700 points.

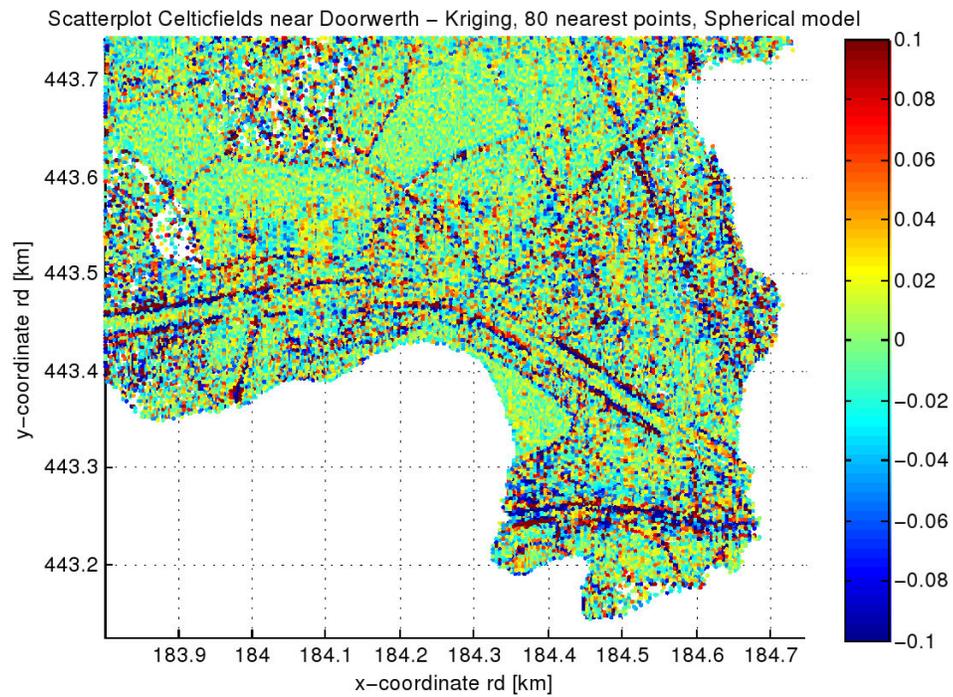


Figure A-4: Scatterplot Celtic fields, spherical model, based on the subset of 4000 points.

Interpolated surfaces:

Residual height after OK kriging with the exponential model - cubic interpolated surface

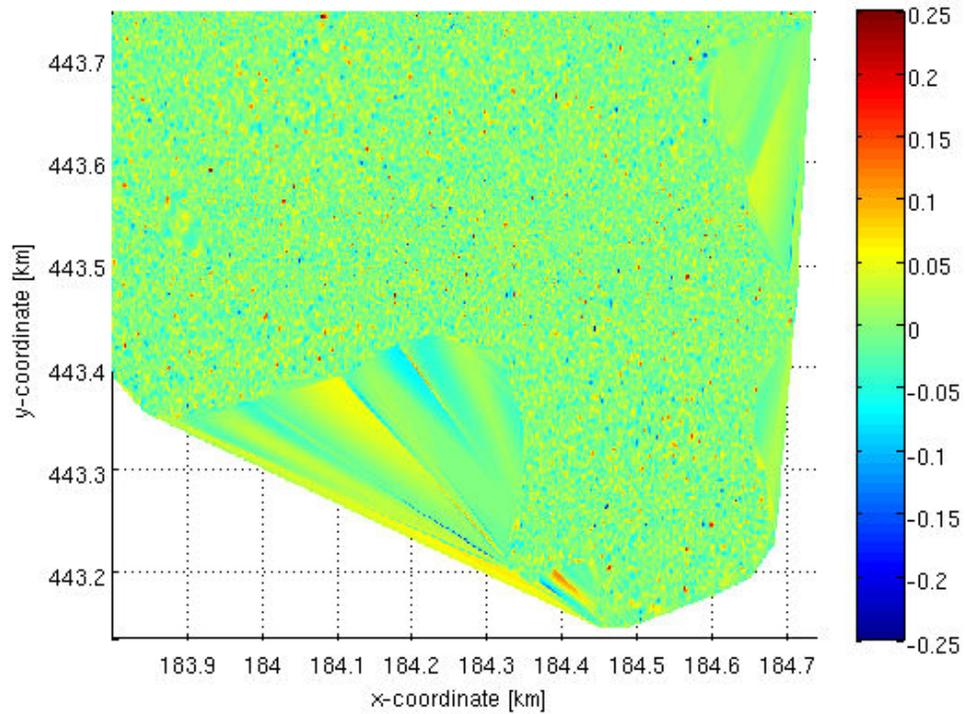


Figure A-5: Cubic interpolated Celtic fields, based on the subset of 700 points.

Residual height after OK kriging with the exponential model - linear interpolated surface

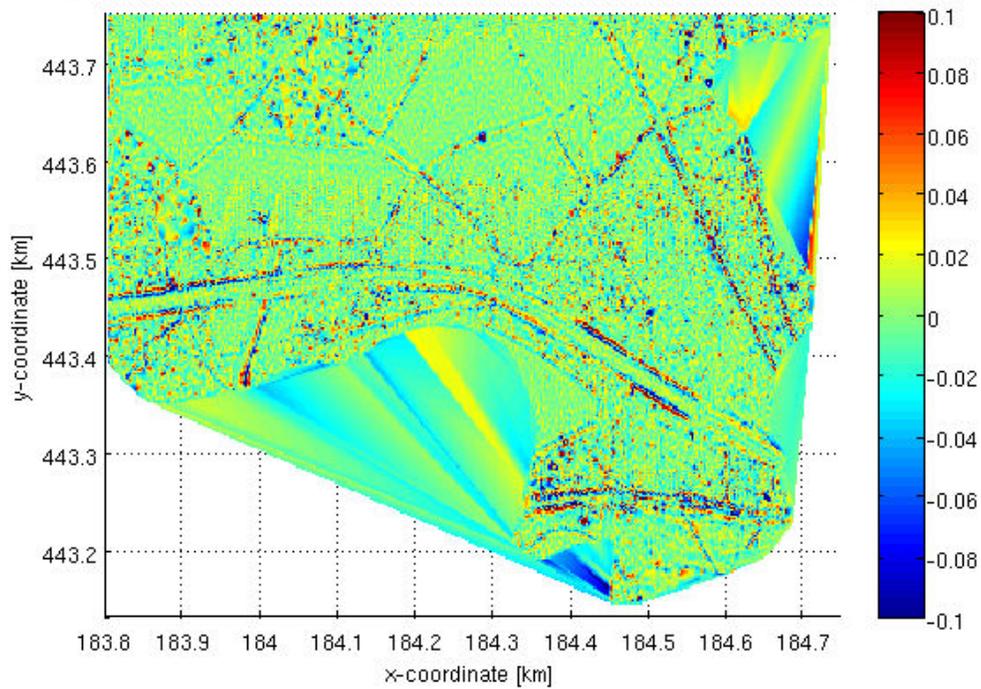


Figure A-6: Cubic interpolated Celtic fields, based on the subset of 4000 points.

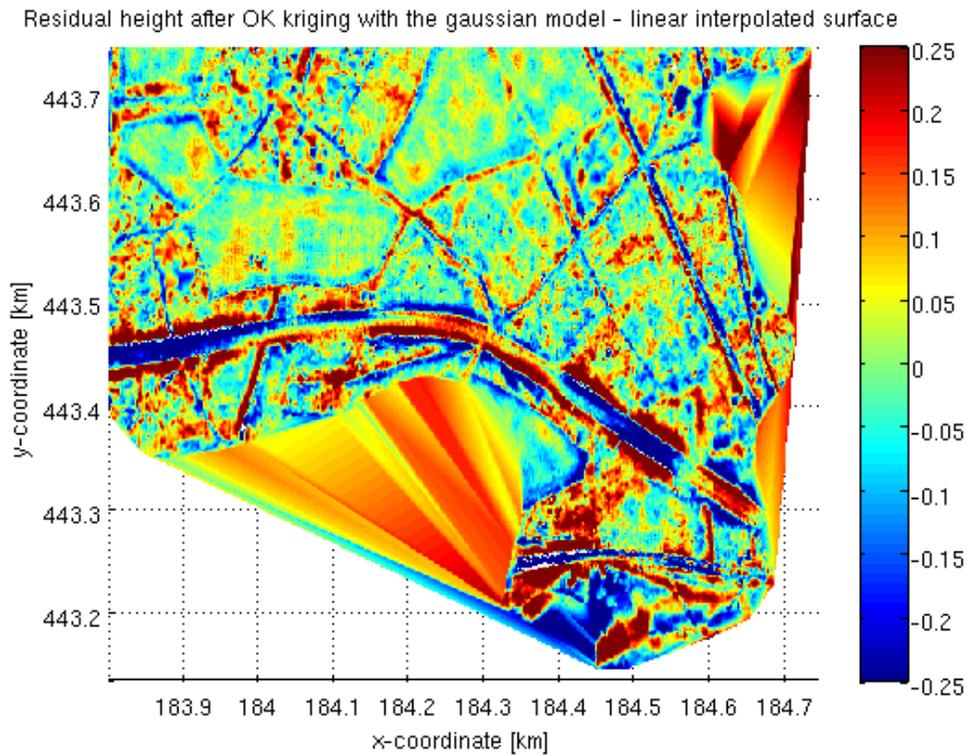


Figure A-7: Gaussian interpolated Celtic fields, based on the subset f 700 points

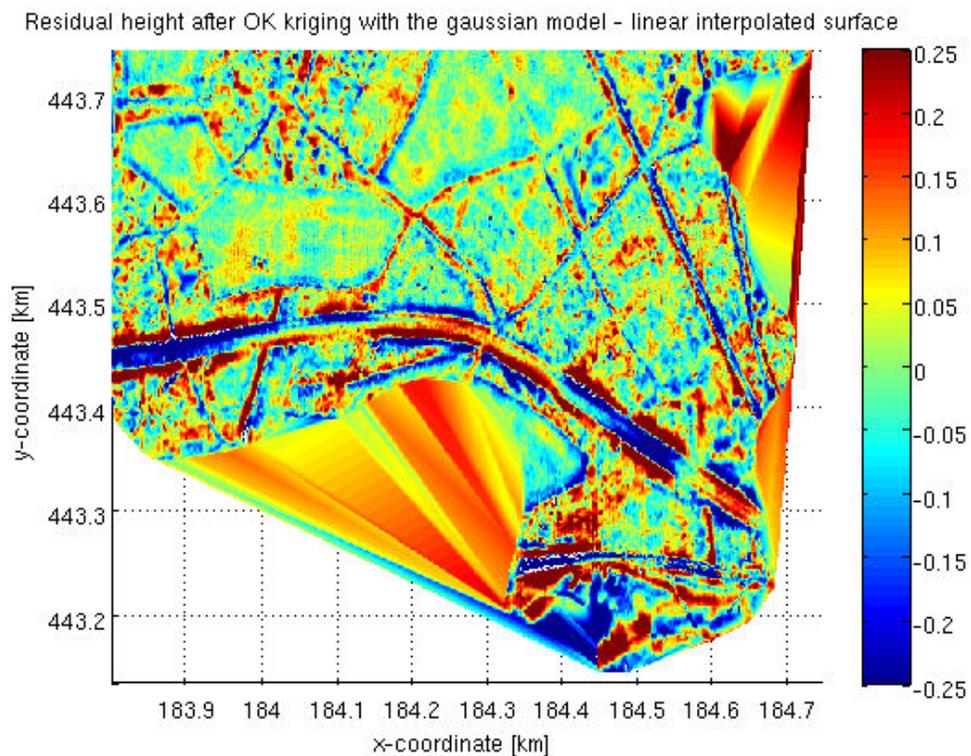


Figure A-8: Gaussian interpolated Celtic fields, based on the subset f 4000 points

Residual height after OK kriging with the spherical model - nearest interpolated surface

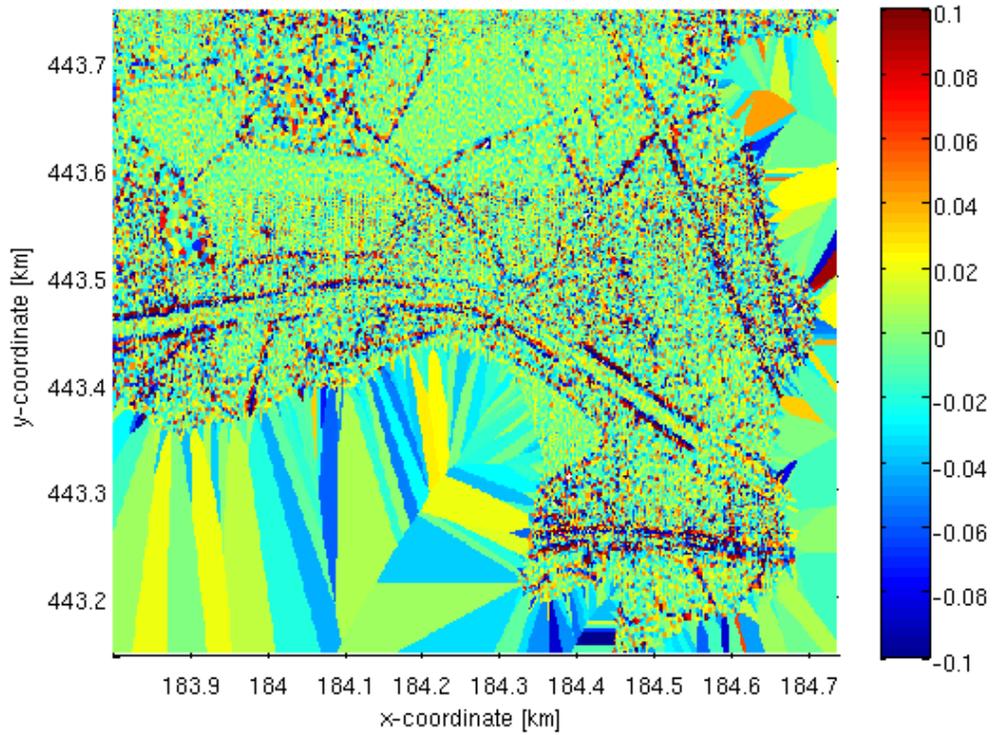


Figure A-9: Spherical interpolated Celtic fields, based on the subset f 700 points

Residual height after OK kriging with the spherical model - cubic interpolated surface

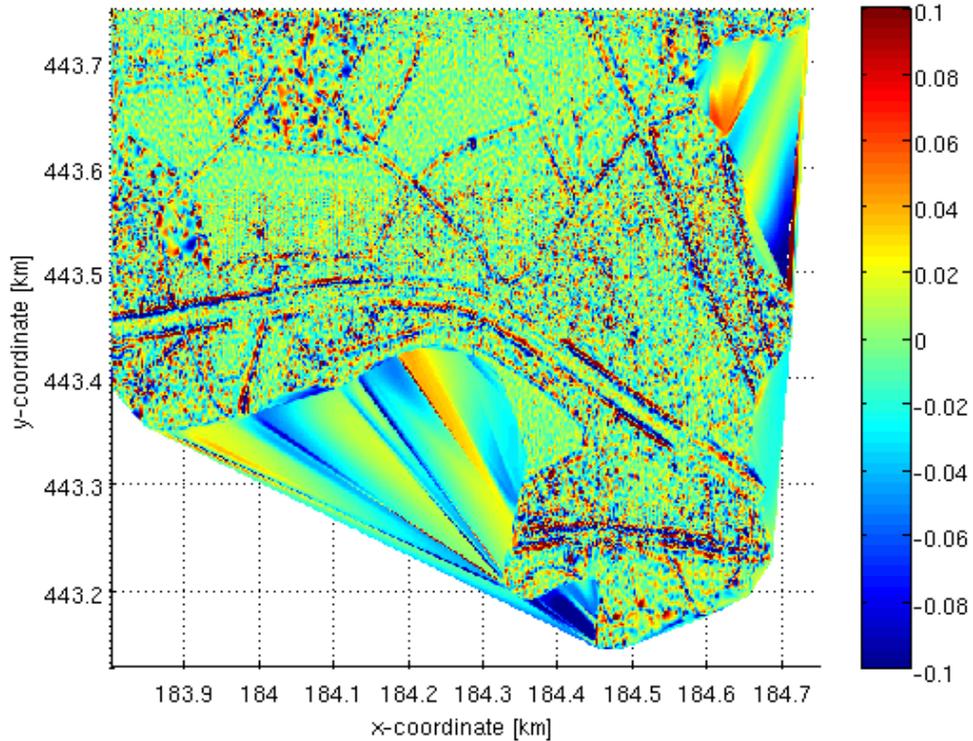


Figure A-10: Spherical interpolated Celtic fields, based on the subset f 4000 points

Appendix B

```
clear all
close all
hold off

%Open the data
load crop_doorwerth

%create a subset -> Random data selector with N the amount of data
points
N = 8000;
[r,c] = size(XYZc);
XYZs = XYZc(randperm(r),:);
XYZs = XYZs(1:N,:);
[r,c] = size(XYZs);

%create a scatterplot
figure(1)
scatter3(XYZs(:,1),XYZs(:,2),XYZs(:,3),5,XYZs(:,3))
colorbar;
view(2);
xlabel('xrd [km]')
ylabel('yrd [km]')

% extrema
extr = zeros(4,1);
extr(1,1) = min(XYZ(:,3));
extr(2,1) = max(XYZ(:,3));
extr(3,1) = mean(XYZ(:,3));
extr(4,1) = var(XYZ(:,3));
save min_max_mean_var_doorwerth_crop extr
```

Appendix C

```
clear all
close all
hold off

%Open the data
load -ascii crop_doorwerth
XYZc = crop_doorwerth;

clear crop_doorwerth

%-----create a subset -> Random data selector with N the
amount of data points-----

%assume fields are not on the slopes, remove slope values
I = find(XYZc(:,3) >= 45);
XYZc = XYZc(I,:);
%create subset
N = 200;
[r,c] = size(XYZc);
XYZs = XYZc(randperm(r),:);           % 1 point 7 m^2
XYZs = XYZs(1:N,:);
[r,c] = size(XYZs);

%scatterplot subset
figure(1)
scatter(XYZs(:,1), XYZs(:,2), 10, XYZs(:,3), 'filled');
title('Scatterplot subset Doorwerth data')
xlabel('x-coordinate [m]')
ylabel('y-coordinate [m]')
colorbar;
box on

%-----raw variogram-----
-----

%computation of dissimilarities against spacial seperation
for i = 1:r
    for j = 1:r
        h(i,j) = sqrt( (XYZs(i,1)-XYZs(j,1))^2 + (XYZs(i,2)-
XYZs(j,2))^2 );
        gamma(i,j) = (1/2)*( (XYZs(i,3)-XYZs(j,3))^2 );
    end
end

[hr, hc] = size(h);
h_gam = [ reshape(h,1, (hr*hc)); reshape(gamma,1, (hr*hc)) ];
I = find(h_gam(1,:) > 0);
h_gam = h_gam(:,I);
h_gam2 = h_gam;
maxh_gam = max(h_gam2(1,:));

%clear unneeded variables
clear h gamma i j r c N hr hc I XYZc

%-----experimental variogram-----
-----
```

```

nbin = 80; %nbin number of bins

for i=1:nbin
    %find the columnindexes for distances within a bin and create the
matrix
    hi = find( i*maxh_gam/nbin >= h_gam2(1,:));
    h_gami = h_gam2(:,hi);

    %create a matrix with the distance and gamma values for the bins
    [r,c] = size(h_gami);
    h_gam_ev(i,:) = [ i*maxh_gam/nbin, sum(h_gami(2,:)/c) ];
    %remove the distances already used during this loop
    hi = find(h_gam2(1,:) > i*maxh_gam/nbin);
    h_gam2 = h_gam2(:,hi);
end

%plot the bars and raw variogram
figure(2)
subplot(2,1,1)
Bar(h_gam_ev(:,1)-maxh_gam/(2*nbin), h_gam_ev(:,2),1,'w');
hold on
%scatter(h_gam(1,:),h_gam(2,:),1,'k','filled');
title('Raw variogram and bars - subset of 200 points')
xlabel('Spatial separation with bins h [m]')
ylabel('Dissimilarities \gamma')
grid on

%calculate nugget
nugget = min(h_gam_ev(:,2))
% save nugget nugget;

%introduce nugget into matrix
h_gam_ev(nbin+1,1) = 0;
h_gam_ev(nbin+1,2) = nugget;
[q I] = sort(h_gam_ev);
h_gam_ev = h_gam_ev(I(:,1),:);

% plot the experimental and raw variogram
subplot(2,1,2)
scatter(h_gam(1,:),h_gam(2,:),1,'k','filled');
hold on
plot(h_gam_ev(:,1), h_gam_ev(:,2),'g','LineWidth',3);
title('Experimental and raw variogram - subset of 200 points')
xlabel('Spatial separation with bins h [m]')
ylabel('Dissimilarities \gamma')
box on

%clear variables
clear c r hi i h_gam2 h_gami h_gam q

%----- To fit an theoretical variogram to the data, copied-----
-----

%specify needed variables for functions
xdata = h_gam_ev(:,1)';
ydata = h_gam_ev(:,2)';

%-----method 1-----
--

%compile wegschrijf matrices

```

```

sse_total = zeros(1,3);
theo_vario = zeros(3,nbin+1);
estimates_total = zeros(3,2);

% Exponential
[estimates, model] = expomodel_new(xdata,ydata);
[sse, FittedCurve] = model(estimates);
estimates_total(1,:) = estimates;
sse_total(1,1) = sse;
theo_vario(1,:) = FittedCurve;

% gaussian function
[estimates, model] = gaussmodel_new(xdata,ydata);
[sse, FittedCurve] = model(estimates);
estimates_total(2,:) = [estimates(1,1) -estimates(1,2)];
sse_total(1,2) = sse;
theo_vario(2,:) = FittedCurve;

% spherical function
[estimates, model] = sphermodel_new(xdata,ydata);
[sse, FittedCurve] = model(estimates);
estimates_total(3,:) = [8.5 650];
sse_total(1,3) = sse;
theo_vario(3,:) = FittedCurve;

% save estimates estimates_total

%plot all the variograms
figure(3)
subplot(2,1,1);
plot(h_gam_ev(:,1),h_gam_ev(:,2),'g','LineWidth',1)
hold on
plot(xdata, theo_vario(1,:), 'b','LineWidth',1)
plot(xdata, theo_vario(2,:), 'r','LineWidth',1)
plot(xdata, theo_vario(3,:), 'k','LineWidth',1)
grid on
box on
legend('experimental variogram', 'exponential model','gaussian
model', 'spherical model')
title('Experimental and theoretical variograms - subset of 700
points')
xlabel('Spatial separation with bins h [m]')
ylabel('Dissimilarities \gamma')

clear FittedCurve R estimates model nbin sill sse ydata

%computing variograms
vare = estimates_total(1,1);
Re = 1/estimates_total(1,2);
varg = estimates_total(2,1);
Rg = -1/estimates_total(2,2);
var = 8.5;
R = 650;
gammaexpo = vare*(1-exp(-(3*xdata./Re)))+nugget;
gammaauss = var*(1-exp(-((3*xdata/Rg).^2))+nugget;
gammaesper = var*(3*xdata/(2*R)-xdata.^3/(2*R.^3))+nugget; %
Geldt alleen voor h<R anders variancie
It = find(xdata > R);
if It > 0
    gammaspher(:,It) = ones(1,size(It,2))*(var+nugget);

```

```

end

% Plotting the theoretical variogram
figure(3)
% subplot(2,1,2);
plot(h_gam_ev(:,1),h_gam_ev(:,2),'g','LineWidth',1);
hold on
plot(xdata, gammaexpo, 'b','LineWidth',1)
plot(xdata, gammagauss, 'r','LineWidth',1)
plot(xdata, gammaspher, 'k','LineWidth',1)
legend('experimental variogram', 'exponential model','gaussian
model', 'spherical model')
title('Experimental and theoretical variograms - subset of 4000
points')
xlabel('Spatial seperation with bins h [m]')
ylabel('Dissimilarities \gamma')
grid on
box on

function [estimates, model] = expomodel(xdata, ydata)           %output =
funcnienaam(input)
% Call fminsearch with a random starting point.
%start_point = rand(1, 2);
start_point = [xdata(2),ydata(2)]
model = @expfun;
estimates = fminsearch(model, start_point);
% expfun accepts curve parameters as inputs, and outputs sse,
% the sum of squares error for  $A * \exp(-\lambda * xdata) - ydata$ ,
% and the FittedCurve. FMINSEARCH only needs sse, but we want to
% plot the FittedCurve at the end.
    function [sse, FittedCurve] = expfun(params)
        A = params(1);
        lambda = params(2);
        FittedCurve = A .* (1-exp(-lambda * 3*xdata)) ;           %A =
var; lambda = 1/R; xdata = h
        ErrorVector = FittedCurve - ydata;                       %ydata =
experimental variogram
        sse = sum(ErrorVector .^ 2);

        dlmwrite('expofit_ALsse.txt',{A,lambda,sse});
        dlmwrite('expofit_FC.txt',FittedCurve);
        dlmwrite('expofit_EV.txt',ErrorVector);
    end
end

function [estimates, model] = gaussmodel(xdata, ydata)
% Call fminsearch with a random starting point.
% start_point = rand(1, 2)
start_point = [xdata(2),ydata(2)]
model = @gaussfun;
estimates = fminsearch(model, start_point);
% expfun accepts curve parameters as inputs, and outputs sse,
% the sum of squares error for  $A * \exp(-\lambda * xdata) - ydata$ ,
% and the FittedCurve. FMINSEARCH only needs sse, but we want to
% plot the FittedCurve at the end.
    function [sse, FittedCurve] = gaussfun(params)
        A = params(1);
        lambda = params(2);
        FittedCurve = A .* (1-exp(-(lambda .^2 * (3*xdata).^2))) ;
        ErrorVector = FittedCurve - ydata;
        sse = sum(ErrorVector .^ 2);
    end
end

```

```

        dlmwrite('gaussfit_ALsse.txt',{A,lambda,sse});
        dlmwrite('gaussfit_FC.txt',FittedCurve');
        dlmwrite('gaussfit_EV.txt',ErrorVector');
    end
end

function [estimates, model] = sphermodel(xdata, ydata)
% Call fminsearch with a random starting point.
%start_point = rand(1, 2);
start_point = [xdata(2),ydata(2)]
model = @spherfun;
estimates = fminsearch(model, start_point);
% expfun accepts curve parameters as inputs, and outputs sse,
% the sum of squares error for A * exp(-lambda * xdata) - ydata,
% and the FittedCurve. FMINSEARCH only needs sse, but we want to
% plot the FittedCurve at the end.
    function [sse, FittedCurve] = spherfun(params)
        A = params(1);
        lambda = params(2);
        FittedCurve = A.*((1/2)*(3*xdata*lambda)-(xdata.^3
*(lambda.^3)));
        ErrorVector = FittedCurve - ydata;
        sse = sum(ErrorVector .^ 2);

        dlmwrite('spherfit_ALsse.txt',{A,lambda,sse});
        dlmwrite('spherfit_FC.txt',FittedCurve');
        dlmwrite('spherfit_EV.txt',ErrorVector');
    end
end

```

Appendix D

```

clear all
% warning off

%Open the data
load crop_doorwerth
XYZc = crop_doorwerth;
load nugget
load estimates
estimates_total = estimates;

clear crop_doorwerth estimates

% loading parameters
e = nugget; %nugget
Re = 1/(estimates_total(1,2)); %range for
best fitting exponential model
vare = estimates_total(1,1); %variance for
best fitting exponential model
Rg = 1/(estimates_total(2,2)); %range for
best fitting gaussian model
varg = estimates_total(2,1); %variance for
best fitting gaussian model
Rs = 1/(0.0016); %range for
spherical model
vars = 7.8; %variance for
spherical model

%-----Ordinary Kriging-----
%-----

%assume fields are not on the slopes, remove slope values
I = find(XYZc(:,3) >= 45); %find values
higher then 45 m, to get a better model
XYZc = XYZc(I,:); %eliminate
points lower then 45 m
%create subset
[r,c] = size(XYZc);
M = 20050; %amount
of points in the subset (r = total dataset)
%XYZ = XYZc(randperm(r),:); %shuffle the
datapoints
XYZ = XYZc(20001:M,:); %create
the dataset with M points

%minimum maximum mean
xmin = min(XYZ(:,1));
xmax = max(XYZ(:,1));
ymin = min(XYZ(:,2));
ymax = max(XYZ(:,2));

mean = mean(XYZ);

clear I gammagauss h c X Y

```

```

%-----only use the N closest points-----
%-----
N = 175; %amount of
points in the neighborhood used for weight calculation

%for every point of the subset calculate the distance to all the
points and
%sort,choose the N closest points
for l = 1:50
    l
    XYZs = sortrows([XYZc, sqrt((XYZc(:,1)-XYZ(l,1)).^2 + (XYZc(:,2)-
XYZ(l,2)).^2)],4);
    XYZs = XYZs(1:N, 1:3);

%Creating the redundacy matrices
Cns = ones(N+1);
Cng = ones(N+1);
ne = ones(N+1);

for i=1:N
    for j=1:N
        if i==j
            Cns(i,j) = vars+nugget;
            Cng(i,j) = varg+nugget;
            Cne(i,j) = vare+nugget;
        else
            h = sqrt( (XYZs(i,1)-XYZs(j,1))^2 + (XYZs(i,2)-
XYZs(j,2))^2 );
            Cng(i,j) = (varg)*exp(-(3*h/Rg)^2);
            Cne(i,j) = (vare)*exp(-3*h/Re);
            if h > Rs;
                Cns(i,j) = 0;
            else
                Cns(i,j) = (vars)*(1 - (3*h)/(2*Rs) +
(h^3)/(2*(Rs^3)) );
            end
        end
    end
end

Cns(N+1,N+1)=0;
Cng(N+1,N+1)=0;
Cne(N+1,N+1)=0;

% Creating the proximity vectors
ht = sqrt( (XYZs(:,1)-XYZ(1,1)).^2+(XYZs(:,2)-XYZ(1,2)).^2 );

dng1(:,1) = [(varg)*exp(-((3*ht/Rg).^2));1];
if dng1(1,1) %due to the way
the subset is created, they first one is always the point we are
looking at
    dng1(1,1) = varg + nugget;
end

dng2(:,1) = [(varg)*exp(-((3*ht/Rg).^2));1];
if dng2(1,1)
    dng2(1,1) = varg;
end

```

```

dne1(:,1) = [(vare)*exp(-3*ht/Re);1];
if dne1(1,1)
    dne1(1,1) = vare + nugget;
end

dne2(:,1) = [(vare)*exp(-3*ht/Re);1];
if dne2(1,1)
    dne2(1,1) = vare;
end

dns1(:,1) = [(vars)*(1-(3*ht)/(2*Rs)+(ht.^3)/(2*(Rs^3)));1];
It = find(ht > Rs);
dns1(It,1) = 0;
if dns1(1,1)
    dns1(1,1) = vars + nugget;
end

dns2(:,1) = [(vars)*(1-(3*ht)/(2*Rs)+(ht.^3)/(2*(Rs^3)));1];
dns2(It,1) = 0;
if dns2(1,1)
    dns2(1,1) = vars;
end

Wns1(:,1) = inv(Cns)*dns1(:,1);           % Weight vector
Wns2(:,1) = inv(Cns)*dns2(:,1);           % Weight vector
Wng1(:,1) = inv(Cng)*dng1(:,1);           % Weight vector
Wng2(:,1) = inv(Cng)*dng2(:,1);           % Weight vector
Wne1(:,1) = inv(Cne)*dne1(:,1);           % Weight vector
Wne2(:,1) = inv(Cne)*dne2(:,1);           % Weight vector

clear i j r c I h ht It

% Determining the height of the datapoints and estimated variances;
Zs1(1,1) = Wns1(1:N,1)'*XYZs(:,3);
Zg1(1,1) = Wng1(1:N,1)'*XYZs(:,3);
Ze1(1,1) = Wne1(1:N,1)'*XYZs(:,3);
Zs2(1,1) = Wns2(1:N,1)'*XYZs(:,3);
Zg2(1,1) = Wng2(1:N,1)'*XYZs(:,3);
Ze2(1,1) = Wne2(1:N,1)'*XYZs(:,3);

Evars1(1,1) = vars - diag(Wns1(:,1)'*dns1(:,1));
Evars2(1,1) = vars - diag(Wns2(:,1)'*dns2(:,1));
Evarg1(1,1) = varg - diag(Wng1(:,1)'*dng1(:,1));
Evarg2(1,1) = varg - diag(Wng2(:,1)'*dng2(:,1));
Evare1(1,1) = vare - diag(Wne1(:,1)'*dne1(:,1));
Evare2(1,1) = vare - diag(Wne2(:,1)'*dne2(:,1));

if Evars1(1,1) < 0.000000001
    Evars1(1,1) = 0;
end
if Evars2(1,1) < 0.000000001
    Evars2(1,1) = 0;
end
if Evarg1(1,1) < 0.000000001
    Evarg1(1,1) = 0;
end
if Evarg2(1,1) < 0.000000001
    Evarg2(1,1) = 0;
end
if Evare1(1,1) < 0.000000001
    Evare1(1,1) = 0;
end

```

```

end
if Evare2(1,1) < 0.000000001
    Evare2(1,1) = 0;
end

end

%check weights without nugget
Wns1(1,1)
Wns1(1,1)
Wng1(1,1)
Wng1(1,1)
Wne1(1,1)
Wne1(1,1)

%sum of weights should be one
for i = 1 :10000
    Wts1(i) = sum(Wns1(1:N,i));
    Wtg1(i) = sum(Wng1(1:N,i));
    Wte1(i) = sum(Wne1(1:N,i));
    Wts2(i) = sum(Wns2(1:N,i));
    Wtg2(i) = sum(Wng2(1:N,i));
    Wte2(i) = sum(Wne2(1:N,i));
end

clear M N vars varg vare Re Rg Rs e l Cne Cng Cns dne1 dne2 dng1 dng2
dns1 dns2 i t t1 t2 t3 dt mean ans

%-----calculation of residuals between model and
observations-----

Zg2res = XYZ(:,3) - Zg2;
Ze2res = XYZ(:,3) - Ze2;
Zs2res = XYZ(:,3) - Zs2;

XYZg = [XYZ(:,1) XYZ(:,2) Zg2res];
XYZe = [XYZ(:,1) XYZ(:,2) Ze2res];
XYZs = [XYZ(:,1) XYZ(:,2) Zs2res];

```