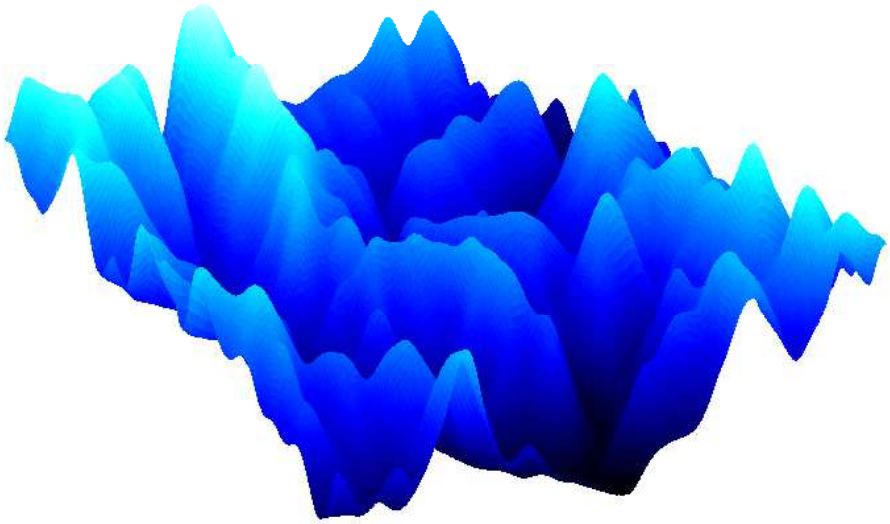


Reconstruction of Seabed Topography using Sonar and GPS data

Master's Thesis in Complex Adaptive Systems

TOBIAS DANNERSTEDT

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2014

MASTER'S THESIS EX032/2014

Reconstruction of Seabed Topography using Sonar and GPS data

Tobias Dannerstedt

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2014

Reconstruction of Seabed Topography using Sonar and GPS data

Master's Thesis EX032/2014
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Tel. +46-(0)31 772 1000

Cover:
Elevation data from a mountain area in Canada used to simulate depth data [1].

Chalmers Reproservice
Gothenburg, Sweden 2014

Abstract

Today's nautical charts are often based on inaccurate data, some that can originate as far back as the late 1800's. New measurements are rarely done and when they are, they are done with trained staff at a high cost. The purpose of this project is to use the data from existing sonar and GPS-units on recreational boats to interpolate a profile of the seabed. The project was carried out in collaboration with another project where the focus was to develop a prototype for storing and sending the measured data to a server. Due to lack of time no real data was collected and instead elevation data from a mountain area was used.

Three methods for spatial interpolation were implemented, Inverse Distance Weighting (IDW), Ordinary Kriging (OK) and Regularized Spline with Tension (RST). The parameters for the algorithms were optimized for the given data set and the methods were compared with respect to the interpolation error, the error propagation and the runtime. An indicator of the quality of the output was also calculated as a function of the distance to the closest known point.

The data set was divided into four different training sets with different densities as well as a validation set used for validating the results. In terms of interpolation error, IDW had the lowest error at low densities of input data while RST had the lowest error for higher densities. When comparing the error propagation IDW performed best for all densities except for the highest where RST performed best. The parameters were however only optimized for low interpolation error and the result from the error propagation can be improved. The runtime for IDW and OK was very similar, which was expected since the methods are based on the same principle. The runtime for RST was much lower and scaled better when the number of data points used was increased.

Contents

1 Introduction

Data from bathymetry, the study of underwater depth of lakes or ocean floors, is important in many aspects. One of the most widespread use of the data is for producing navigation products such as nautical charts. The data is also an important source for many Earth sciences. It can provide information about the effects of climate changes and how, for example a natural disaster such as a tsunami, will impact the ocean and coastline. Other applications is the use of bathymetric maps to help determine where fish and other sea life feed, live and breed. [2]

1.1 Background

The depth information on today's nautical charts in Sweden, but also in other countries, are often inaccurate [3]. This is due to the fact that the data used often is taken from measurements done in the late 1800's using contemporary methods. With the increasingly widespread use of chart plotters (a marine GPS unit which contains a nautical chart), especially in recreational boats, this has become a big problem. People rely blindly on their chartplotter and are ignoring the safety distance. New survey of the sea depth are done regularly at the major sea lanes by the Swedish Maritime Administration (*Sjöfartsverket*) with trained staff at a high cost. At smaller lanes however new measurements are rarely done and large errors can exist.

Today, modern boats of reasonably sizes are often equipped with a sonar. These often have standardized interfaces, e.g. NMEA 801, for connection to the network. By connecting a device with NMEA-interface, GPS, GPRS connectivity, memory and processing to the sonar, the depth information can be stored in memory and at a suitable time be transferred to a server at the agency or organization responsible for the data.

The purpose of this project is to analyse synthetic depth data, generated from elevation data, and create profiles of the seabed which in turn can be used for creating nautical charts. The input data consists of a large set of data points where each data point consists of information about the depth, position and time of measurement. Parallel with this project, another project is carried out which main goal is to collect data.

1.2 Aim

The aim of this thesis is to implement different methods for spatial interpolation. These methods are to be compared with respect to the error in the output as well as the error propagation. Furthermore, the runtime of the algorithms and how they scale with the size of the input data will be analysed. A system for handling changing water level will also be developed.

1.3 Limitations

The collection of input data and where the error in the input data arises from will not be considered. Automated optimization of the parameters in the algorithms will not be focused on. The output will be depth data and the creation of a nautical chart and a good visualisation will not be considered.

2 Theory

This chapter aims to give the theoretical framework and knowledge that is used throughout this thesis. An introduction to geometric theory of the earth is given at first. The theory of different interpolation methods follows. Finally a short introductions to genetic algorithms as well as cross-validation is given which are two techniques that will be used.

2.1 Earth Geometry

The earth is usually approximated as an ellipsoid whose short axis stretches from the center of the earth to the north/south pole while the long axis stretches from the center to the equator. Below the coordinate system used as well as algorithms for distance measurements will be explained.

2.1.1 Coordinate System

To specify a point on the surface of the Earth one uses a geographic coordinate system where the most common type is to use the three coordinates *latitude*, *longitude* and *elevation*, see Figure 2.1(a).

The latitude, denoted ϕ, of a point is the angle between the equatorial plane and the straight line through the given point and normal to the surface of a reference ellipsoid approximating the shape of the Earth, see Figure 2.1(b). One consequence of using a line normal to the surface is that it do not pass through the center of the Earth except at the poles and the equator.

The longitude, denoted λ, of a point is the angle, east or west, from a reference meridian to another meridian going through the point. A meridian is half a great ellipse (an ellipse passing through two points on a spheroid and having the same center as that of the spheroid) and the reference meridian is set to be the meridian passing through the Royal Observatory in Greenwich, London.

There are three main formats when denoting the latitude or longitude:

- "Degrees, minutes, seconds": 40° 26’ 46"
- "Degrees, decimal minutes": 40° 26.767’

- "Decimal degrees": 40.446°

where there are 60 seconds in a minute and 60 minutes in a degree. To distinguish between latitude and longitude one usually puts N or S for *north* or *south* after the latitude and W or E for *west* or *east* after the longitude. Throughout this report the "decimal degrees" format will be used.

2.1.2 Distance Measure

When performing spatial analysis there is need of distance measurement between two points, given their latitude and longitude. To know the distance the radius is needed, but the radius of the earth is not constant. The equatorial radius, R_e, is approximately 6378.16 km while the polar radius, R_p, is approximately 6356.78 km [4]. To simplify the calculations the Earth is usually approximated as a perfect sphere instead, using a weighted mean for the radius, namely $R = \frac{2R_e+R_p}{3}$.

Instead of using the mean radius, the real radius can be calculated for any given latitude. Figure 2.1(b) shows an exaggerated cut of the earth in the shape of a quarter of an ellipse. In this figure two angles are shown. The first angle, θ, is the angle between a straight line, denoted R in the figure, between the center of the earth and a given point on the surface. The second angle, ϕ, is the angle between a line normal to the surface at the given point and the equatorial plane. This line is denoted f in the figure and the angle ϕ is equal to the latitude of the given point.

The ellipse can be parametrised using the angle θ with $(R_e \cos\theta,\ R_p \sin\theta)$ and the radius can be computed using the Pythagorean theorem

$$R^2 = R_e^2 \cos^2\theta + R_p^2 \sin^2\theta \tag{2.1}$$

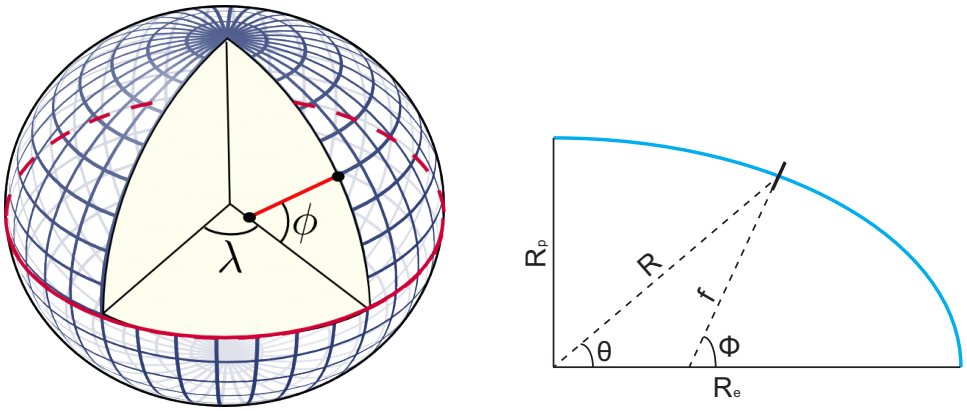
Given the latitude, ϕ, the radius R needs to be calculated. The first step is to find a relationship between ϕ and θ. The tangent of the surface is $(-R_e \sin\theta,\ R_p \cos\theta)$ from the parametrisation and the line f will be perpendicular to this, i.e. parallel with the normal. The normal can be calculated by rotating the tangent 90 degrees which gives $(R_p \cos\theta,\ R_e \sin\theta)$. From this the angle ϕ can be expressed by

$$\tan\phi = \frac{R_e \sin\theta}{R_p \cos\theta} \Rightarrow \frac{R_p}{R_e}\tan\phi = \frac{\sin\theta}{\cos\theta} = \tan\theta \tag{2.2}$$

This is used to substitute θ in (2.1) by

$$\cos^2\theta = \frac{1}{1+\tan^2\theta} = \frac{1}{1+\frac{R_p^2}{R_e^2}\tan^2\phi} = \frac{R_e^2}{R_e^2 + R_p^2\tan^2\phi} = \frac{R_e^2\cos^2\phi}{R_e^2\cos^2\phi + R_p^2\sin^2\phi} \tag{2.3}$$

$$\sin^2\theta = 1 - \cos^2\theta = \frac{R_p^2\sin^2\phi}{R_e^2\cos^2\phi + R_p^2\sin^2\phi} \tag{2.4}$$



(a) (b)

Figure 2.1: (a): The definition of latitude, ϕ, and longitude, λ. *By Peter Mercator via Wikimedia Commons.* (b): An exaggerated image of the earth as an ellipse. ϕ represents the latitude, R_e and R_p the equatorial and polar radius. R represents the radius at the given point.

and

$$R^2 = \frac{R_e^4 \cos^2\phi + R_p^4 \sin^2\phi}{R_e^2 \cos^2\phi + R_p^2 \sin^2\phi} \Rightarrow R = \sqrt{\frac{(R_e^2\cos\phi)^2 + (R_p^2\sin\phi)^2}{(R_e\cos\phi)^2 + (R_p\sin\phi)^2}} \tag{2.5}$$

which gives the radius as a function of the latitude.

When calculating the distance of two points on the surface of the earth the *Great Circle Distance* is usually used. However, when the distances are small (the points are less than 1km apart) rounding errors usually occur due to the precision in the computer. Therefore the *Haversine formula* is better [5] and will be used in this thesis. Using Haversine, the distance between two points is given by

$$d = 2R\arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2-\phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \tag{2.6}$$

where R is the radius and $\phi_{1,2}$ is the latitudes and $\lambda_{1,2}$ is the longitudes. Due to the fact the the radius of curvature of the earth is not constant the Haversine can have an error up to $\pm 0.5\%$ [5]. But an accuracy in the distance of $\pm 0.5\%$ is more than enough for this application and a more accurate method will be less computer efficient.

2.2 Interpolation Methods

Spatial interpolation estimates values at unobserved locations in an area covered by existing observation, called control points. Given N values of a studied phenomenon, z_j, at points $\mathbf{x_j}$ for $j = 1, \ldots, N$ one wants to find a function $Z(\mathbf{x})$ that fulfils

$$Z(\mathbf{x_j}) = z_j \tag{2.7}$$

for $j = 1, \ldots, N$ [6][7].

Spatial interpolation methods can either be global or local. Global interpolation uses all available control points while local interpolation only uses a sample of the control points in a local area.

Another way to classify spatial interpolation methods is exact and inexact interpolation. Exact interpolation predicts a value at the control points that are the same as the observed values while the inexact interpolation may predict values for the control points that differ from the observed values.

A third classification is deterministic interpolation as well as stochastic interpolation. In deterministic interpolation one assumes that all the knowledge necessary to describe the system is known. The system can for example be described by a physical model where phenomena results from a process that minimises the energy. In stochastic interpolation one instead incorporates a stochastic term in the interpolated values representing for example stochastic fluctuations in the environment.

A list of methods for spatial interpolation along with their classifications can be seen in Table 2.1. Three of these are suitable for elevation studies [8] and they are described in the following sections. These three methods are

- Inverse Distance Weighted (local, deterministic, exact)
- Ordinary Kriging (local, stochastic, exact/inexact)
- Regularized Spline with Tension(local, deterministic, exact)

Table 2.1: Examples of spatial interpolation methods and their classification.

Global		Local	
Deterministic	Stochastic	Deterministic	Stochastic
Trend surface (exact)	Regression (inexact)	Thiessen (exact) Density estimation (inexact) Inverse distance weighted (exact) Splines (exact)	Kriging (exact)

2.2.1 Inverse Distance Weighting

Inverse Distance Weighted, or *IDW*, is the most common methods in the family *Weighted Moving Average* methods (WMA). The general formula for a WMA is

$$\hat{z}(\mathbf{x_0}) = \sum_{i=1}^{N} \lambda_i z(\mathbf{x_i}) \tag{2.8}$$

where $z(\mathbf{x_i})$ are the data values for the N points $(\mathbf{x_1}, \ldots, \mathbf{x_N})$ and $\hat{z}(\mathbf{x_0})$ is the estimate at $\mathbf{x_0}$. The equation states that the value of an arbitrary point is a linear combination of the known points where each point is weighted with λ_i. A condition to these weights are that they must sum up to one

$$\sum_{i=1}^{N} \lambda_i = 1 \tag{2.9}$$

To make the algorithm more efficient only a subset of the known points are used. The most common approach is to select the subset is to either only look at points within a certain radius or select the n closest points.

IDW specifies how to select the weights in (2.8) according to

$$\lambda_i = \frac{d_{i0}^{-\alpha}}{\sum_{i=1}^{N} d_{i0}^{-\alpha}} \tag{2.10}$$

where d_{i0} denotes the euclidean distance between the point that is being interpolated and the known point i. The α-parameter is chosen a priori and will determine the so called distance-decay effect. Smaller α tends to yield estimates as an average of the known points $z(\mathbf{x_i})$. For example if we let $\alpha \to 0$ we get

$$\lambda_i = \lim_{\alpha \to 0} \frac{d_{i0}^{-\alpha}}{\sum_{i=1}^{N} d_{i0}^{-\alpha}} = \frac{1}{\sum_{i=1}^{N} 1} = \frac{1}{N} \tag{2.11}$$

and

$$\hat{z}(\mathbf{x_0}) = \sum_{i=1}^{N} \frac{1}{N} z(\mathbf{x_i}) \tag{2.12}$$

which is the definition of the average.

Larger values of α will instead give larger weights to the nearest point and decreasing weights to points further away. If we let $\mathbf{x_j}$ denote the nearest point to $\mathbf{x_0}$ and L_i denote the distance between $\mathbf{x_i}$ and $\mathbf{x_0}$ we get $\min(L_i) = L_j$. Considering this and applying $\alpha \to \infty$ we get

$$\lambda_i = \begin{cases} 1 & i = j \quad (L_j = \min(L_i)) \\ 0 & i \neq j \end{cases} \tag{2.13}$$

and

$$\hat{z}(\mathbf{x_0}) = z(\mathbf{x_j}) \tag{2.14}$$

so the estimated value equals the value of the closest point. Usually one sets α to a value between 1 and 3 [9]. When $\mathbf{x_0} = \mathbf{x_i}$ one sets $\lambda_i = 1$ and $\lambda_{j \neq i} = 0$.

The relative weight (before being normalized) can be seen in Figure 2.2 where we can see that larger values of α increases the weights for small distances.

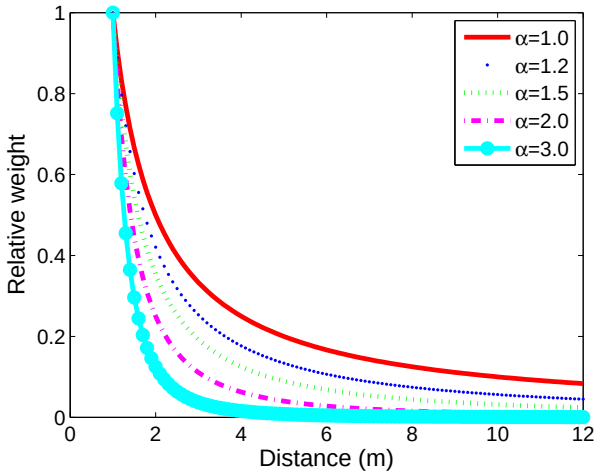


Figure 2.2: The relative weight, λ, as a function of the distance for different values of α in (2.10).

2.2.2 Ordinary Kriging

Kriging is a much more complex method than IDW but along with this comes a number of advantages. Kriging is less susceptible to arbitrary decisions when implemented (such as search radius, number of sample points e.t.c.) and it also provides an indication on the reliability of the estimate. [10]

There exist various types of Kriging, the two most simple methods are *Ordinary Kriging*, referred to as *OK*, and *Simple Kriging*. Ordinary Kriging is the original formulation of Kriging which assumes that the mean of the measured property is unknown while Simple Kriging assumes that the mean is known. The ordinary Kriging is by far the most widely used type [10] and it is also the one that will be used in this thesis.

Before describing Ordinary Kriging algorithm, the variogram which is a central concept in this method will be explained.

The Variogram

A variogram is a function describing the spatial dependence of a spatial stochastic process $Z(\mathbf{x})$. Assume that a property $Z(\mathbf{x})$ at position $\mathbf{x}$ is a random variable with mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$. Also make the assumption that two values $Z(\mathbf{x_i})$ and $Z(\mathbf{x_j})$ where $\mathbf{x_i}$ and $\mathbf{x_j}$ are near one another also are related. The covariance function for the random variable is described by

$$C(\mathbf{x_i}, \mathbf{x_j}) = E[\{Z(\mathbf{x_i}) - \mu(\mathbf{x_i})\}\{Z(\mathbf{x_j}) - \mu(\mathbf{x_j})\}] \tag{2.15}$$

where E denotes the expected value. Since only the realisation of $Z(\mathbf{x})$ is given by the measurement $z(\mathbf{x})$ the mean will be unavailable and the equation lacks a solution.

Assume that the mean will be constant and that Z is a stationary process. The mean can be estimated from repetitive sampling and $\mu(\mathbf{x_1})$ and $\mu(\mathbf{x_2})$ in (2.15) can be replaced with μ. Their covariance will now only depend on the separation $\mathbf{h} = \mathbf{x_i} - \mathbf{x_j}$ and not their absolute position. Using this (2.15) can be simplified to

$$C(\mathbf{x_i}, \mathbf{x_j}) = E[\{Z(\mathbf{x_i}) - \mu\}\{Z(\mathbf{x_j}) - \mu\}] = E[\{Z(\mathbf{x})\}\{Z(\mathbf{x}+\mathbf{h})\} - \mu^2] = C(\mathbf{h}) \tag{2.16}$$

where $\mathbf{h}$, which is a vector in both distance and direction, is called the *lag*.

However, the assumption that the mean is constant usually does not hold and the covariance can not be easily estimated. A solution to this is to only look at small lag distance where the expected value of two values would be the same. When the expected difference is zero their variance is defined as follows

$$E[\{Z(\mathbf{x}) - Z(\mathbf{x}+\mathbf{h})\}^2] = \text{var}[Z(\mathbf{x}) - Z(\mathbf{x}+\mathbf{h})] = 2C(0) - 2C(\mathbf{h}) = 2\gamma(\mathbf{h}) \tag{2.17}$$

where $\gamma(\mathbf{h})$ is known as the *semivariance* at lag $\mathbf{h}$ and the function is called the *variogram*.

There exists different methods for estimating the variogram from sampled data and the most common is Matherhorn's [10] method of moments. When one have irregular sampled data the plot $\gamma(\mathbf{h})$ against $\mathbf{h}$ would be very scattered and hard to interpret if it would be calculated for every lag existing in the data. Instead the separation between pairs of points are placed into bins with limits in distance and direction. The semivariance is then estimated according to

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2m(\mathbf{h})} \sum_{i=1}^{m(\mathbf{h})} \{z(\mathbf{x_i}) - z(\mathbf{x_i}+\mathbf{h})\}^2 \tag{2.18}$$

where $z(\mathbf{x_i})$ and $z(\mathbf{x_i}+\mathbf{h})$ are measurements and $m(\mathbf{h})$ is the number of paired comparison in the same bin, defined by the lag $\mathbf{h}$.

To summarize, instead of calculating the variance for each pair, pairs with similar lag are placed into bins and the mean is calculated. The bigger the bins are, the more smooth will the variogram be, but it will also result in a loss of detail.

To make use of the variogram a function has to be fitted to the sampled data to create a continuous variogram. Instead of calculating a variogram-surface depending on both the distance and the direction, the lags $\mathbf{h}$ are grouped using their directions where each group contains lags with similar directions. By doing this it only remains to create one variogram in each group, or direction, which only depends on the size of $\mathbf{h}$, denoted h. This way N variograms are created where each $\gamma_i(h)$ with $i = 1, \ldots, N$ gives the variogram in a specific direction. Here N denotes the number of directions used to discretize the directions between points.

The functions that are fitted must be reasonable in the sense that it can represent spatial characteristics. Three of the most common variogram models for spatial interpolation [10] are described below

- The stable exponential model

$$\gamma(h) = c_0 + c\left(1 - e^{-\frac{h^\alpha}{r^\alpha}}\right) \tag{2.19}$$

- The spherical model

$$\gamma(h) = \begin{cases} 0 & h = 0 \\ c_0 + c\left(\frac{3h}{2a} - \frac{h^3}{2a^3}\right) & 0 \leq h \leq a \\ c_0 + c & h > a \end{cases} \tag{2.20}$$

- The power model

$$\gamma(h) = c_0 + c \cdot h^a \tag{2.21}$$

where the parameters are optimized to fit the sampled data. There are three values of the variogram describing the properties. The first one is the nugget that describes the variance at zero lag. The second one is the range which gives the distance at which there no longer is any correlation. The third one is the sill which is the maximum variance. [10]

The Algorithm

The estimate of z at $\mathbf{x_0}$ is denoted $\hat{z}(\mathbf{x_0})$ and is a weighted mean according to

$$\hat{z}(\mathbf{x_0}) = \sum_{i=1}^{n} \lambda_i z(\mathbf{x_i}) \tag{2.22}$$

were, as in IDW, n is a subset, usually the local neighbourhood, of N. To ensure that the estimate is unbiased the weights are made to sum to one and the expected difference between two close points are assumed to be zero. The expected error is $E[\hat{Z}(\mathbf{x_0}) - Z(\mathbf{x_0})] = 0$ and the predicted variance is

$$\text{var}[\hat{Z}(\mathbf{x_0})] = E[\{\hat{Z}(\mathbf{x_0}) - Z(\mathbf{x_0})\}^2] = 2\sum_{i=1}^{n} \lambda_i \gamma(\mathbf{x_i}, \mathbf{x_0}) - \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j \gamma(\mathbf{x_i}, \mathbf{x_j}) \tag{2.23}$$

where $\gamma(\mathbf{x_i}, \mathbf{x_j}) = \gamma_i(h)$. Each variogram, γ_i, represents a specific direction and which one to use is determined by the direction of $\mathbf{h} = \mathbf{x_j} - \mathbf{x_i}$. The functions γ_i is the function of the variogram model that fits the experimental semivariances best. The model with the best fit is defined as the model with the lowest root mean square error when looking at distances smaller than the range since points that are further away are assumed to have no correlation. See Appendix A for the derivation of (2.23).

The goal is to find the right weights, that minimizes the variance between the true value and the estimation. To achieve this the derivative with respect to λ_i is taken for $i = 1, \ldots, n$ and set to zero

$$\sum_{i=1}^{n} \lambda_i \gamma(\mathbf{x_i}, \mathbf{x_j}) + \Psi(\mathbf{x_0}) = \gamma(\mathbf{x_j}, \mathbf{x_0}) \text{ for all } j \tag{2.24}$$

with the constraint

$$\sum_{i=1}^{n} \lambda_i = 1 \tag{2.25}$$

where $\Psi(\mathbf{x_o})$ is the Lagrange multiplier which is introduced to achieve minimization. When this is solved the weight can be inserted into (2.22) and the algorithm is completed. The system of equations can also be written on matrix form, using the notation $\gamma(\mathbf{x_i}, \mathbf{x_j}) = \gamma_{ij}$

$$\begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} & 1 \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \gamma_{nn} & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ \Psi \end{bmatrix} = \begin{bmatrix} \gamma_{10} \\ \gamma_{20} \\ \vdots \\ \gamma_{n0} \\ 1 \end{bmatrix} \tag{2.26}$$

which, together with (2.22), defines the algorithm for the ordinary Kriging [10, 11].

2.2.3 Regularized Spline with Tension

The spline-method has a variational approach to the interpolation, which means that it tries to maximize or minimize a certain function. When interpolating using spline one sets up a function that describes the smoothness and then try to minimize this function to get a surface that is as smooth as possible. Talmi and Gilat [12] suggested the following function for measuring the smoothness in two dimensions

$$I^2(z) = \sum_{\alpha} B_\alpha \iint_\Omega \left[\frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2}} z(\mathbf{x}) \right]^2 dx_1 dx_2 \tag{2.27}$$

where $\alpha = (\alpha_1, \alpha_2)$ is a multiindex with positive integers ($\alpha_1 = 0, 1, 2, \dots$ and $\alpha_2 = 0, 1, 2, \dots$) with $|\alpha| = \alpha_1 + \alpha_2$.

The function (2.27) is a seminorm that includes derivatives of all orders with the nonnegative weights B_α.

Given N numbers of a studied phenomena, the goal is to find a function $z(\mathbf{x})$ that fulfils

$$z(\mathbf{x_j}) = z_j \quad j = 1, \dots, N \tag{2.28}$$

and minimizes $I^2(z)$. One special property of (2.27) is that it has an analytical, unique, solution given by [12]

$$z(\mathbf{x}) = T(\mathbf{x}) + \sum_{j=1}^{N} \lambda_j R(\mathbf{x}, \mathbf{x_j}) \tag{2.29}$$

where $T(\mathbf{x})$ is a trend function given by

$$T(\mathbf{x}) = \sum_{l=1}^{N} a_l f_l(\mathbf{x}) \tag{2.30}$$

where $f_k(\mathbf{x})$ is a set of linearly independent functions with zero smooth seminorm. $R(\mathbf{x}, \mathbf{x_j})$ is a radial basis function which depends on the choice of B_α. When the radial basis function is

known the parameters a_l and λ_i is determined by solving a system of linear equations

$$z(\mathbf{x_j}) = z_j \tag{2.31}$$

$$\sum_{j=1}^{N} \lambda_j f_j(\mathbf{x_j}) = 0 \tag{2.32}$$

for all $j = 1, \ldots, N$.

One extension of Spline is the so called *Thin Plate Spline*, or *TPS*, which refers to the physical analogy involving bending of a sheet of metal and forcing it through some points. The TPS only includes the second order derivatives in (2.27) and gives good result but it has some drawbacks. The major disadvantage is when there are rapid changes in the gradient which usually leads to overshooting due to the plate stiffness.

To overcome this drawback Mitasova [13] suggested a solution which suppress this stiffness by including the first derivative to the smooth seminorm. This method is called TSP with tension where the tension parameter, φ, controls the stiffness and lower values will simulate the behaviour of a membrane while higher values will simulate a thin metal plate. When using tension the weights B_α is determined according to

$$B_\alpha = \begin{cases} 0, & |\alpha| = 0 \\ \frac{|\alpha|!}{\alpha_1!\alpha_2!} \cdot \frac{1}{\varphi^{2|\alpha|}(|\alpha|-1)!}, & |\alpha| > 0 \end{cases} \tag{2.33}$$

The weights B_α decreases with increasing derivative order in (2.27) and φ controls how fast they should decrease. The tension parameter is usually determined empirically.

There still remain some drawbacks in the TSP with tension and the major one is that the functions are not sufficiently general and analysis of the surface can be hard. To overcome this, the third derivative is included near the data-points which leads to the method called *Regularized Spline with Tension* or RST. This method was tested by Hofierka [14] and the result indicated that it performed better than the standard methods when estimating elevation data. However, the equations for the basis function and the derivatives are included which requires more computational power to solve.

Regularized spline with tension includes derivatives of all orders and the corresponding interpolation function is given by a constant trend function

$$T(\mathbf{x}) = a_1 \tag{2.34}$$

and the basis functions

$$R(\mathbf{x}, \mathbf{x_j}) = R(r_j) = -\left\{ \ln\left[\left(\frac{\varphi r_j}{2}\right)^2\right] + \mathrm{E}_1\left[\left(\frac{\varphi r_j}{2}\right)^2\right] + C_E \right\} \tag{2.35}$$

which only depends on the distance r_j between $\mathbf{x}$ and $\mathbf{x_j}$, i.e. a radial basis functions. Here, E_1 is the exponential integral function and $C_E = 0.5772\ldots$[4] is the Euler constant.

To summarize, RST is given by

$$\hat{z}(\mathbf{x}) = a_1 + \sum_{j=1}^{N} \lambda_j R(r_j) \tag{2.36}$$

where the coefficients a_1 and λ_j are obtained by solving the following system of linear equations

$$a_1 + \sum_{j=1}^{N} \lambda_j [R(\mathbf{x_i}, \mathbf{x_j}) + \delta_{ij} w] = \quad z_i, \quad i = 1, \ldots, N \tag{2.37}$$

$$\sum_{j=1}^{N} \lambda_j = \quad 0 \tag{2.38}$$

where w is a smoothing parameter and R is given according to equation (2.35).

Unfortunately, the system (2.37) will be very large if there are many data-points so a suggestion by Mitashova [14] is to implement a segmentation of the algorithm.

To segment the algorithm a mesh is created where a parameter $k_{max} < N$ determines how fine it should be. The cell size is selected such that the 3x3 neighbourhood for all cells have a maximum of k_{max} measurements while being as large as possible. During the interpolation, the interpolation point is put in the corresponding cell and the system of linear equations is only solved using the data points in the 3x3 neighbourhood. This way, the number of linear equations never exceeds k_{max} equations.

Another parameter, $k_{min} < k_{max}$, tells that if the 3x3 neighbourhood of an interpolation point contains less than k_{min} data-points, the neighbourhood is increased until the number of data-points no longer is below k_{min}. Since the system of linear equations only depends on the cell, and not the position within the cell, it only has to be solved one time for each cell. This means that the computational time will be proportional to N. The method is also very suitable for strongly inhomogeneous data, for example clustered data.

Other features exist such as horizontal and vertical scaling as well as rotating of the coordinate axis. By scaling the two horizontal axis differently different parameters for the two different directions can be achieved. Furthermore, by rotating the two horizontal coordinates the directions of the different parameters can be changed. By scaling the vertical axis the stability and range of suitable parameters can be changed. [13] [14]

2.3 Genetic Algorithm

A genetic algorithm (GA) is a stochastic optimization algorithm that has been inspired by natural evolution by mimicking features such as inheritance, mutation, selection and crossover. Genetic algorithms belongs to the larger class of evolutionary algorithms (EA) which tries to optimize problems using techniques inspired by biological evolution. In this thesis the genetic algorithm will be used to fit a function to measured data. The problem to optimize will be the minimization of the error between the fitted function and the sampled values.

The base in a genetic algorithm is the individuals. Each individual has a set of chromosomes, which in the most simple form consist of a string of bits. Each chromosome represents a variable in the problem that is to be optimized. Together all the individuals form a population.

Initially each individual is initialised with random chromosomes that falls within the search space of the problem (all the possible values for the variables). To optimize the problem the evolution is started which consist of a loop in which a new generation is created in each iteration.

The first step in creating a new generation is to rank all the individuals based on how good they are, which is decided by a fitness function that for example is the total error when trying to fit a function to measured values. In proportion to their rank, pairs of individuals are selected and their chromosomes is mixed. This step is called crossover and usually consist of randomly selecting a point in the bit string representing the chromosomes dividing it into two parts. One of the parts is then swapped with one of the parts from the other individual creating two new individual. The chromosomes of the two new individuals is then mutated in which each bit with a certain probability switches from 0 to 1 or from 1 to 0.

This is repeated until a completely new generation has been created which indicates one iteration in the evolution loop. A common feature is called elitism in which the best individual in each generation always survives without crossover or mutation and replaces the worst individual in the new generation. [15]

2.4 Cross-Validation

Cross-validation is a technique that is used to validate a model and more specifically how the result of a statistical analysis will generalize to an independent data set. The most common application is when the goal is prediction. In many prediction problems a model is usually given a data set from which parameters are tuned so that the model can predict the known data set.

However problem arise when the model fits the data set too good and are unable to be generalised to an unknown previously not seen data set, this is called overfitting. Too overcome this cross-validation is used. The form of cross-validation that will be used throughout this project consists of dividing the known dataset into two different sets, one called training set and one called validation set. The training set is used for training and optimizing the parameters of the model while the validation set is used as an unknown set on which the model is tested.

Hopefully, when starting to tune the parameters the performance of the model will increase on both the training set and the validation set. After a while a certain point is usually hit where more tuning of the parameters will increase the performance on the training set, but decrease the performance of the validation set. When this point is hit the best parameters are found since more tuning will make the model less general and perform worse when predicting values from previously not seen data. If the tuning would be carried on overfitting would occur.

3 Method

This chapter describes how the thesis was carried out. First of all the implementation of the three interpolation methods will be described along with the algorithm that handles the changing sea water level. Following this is a description of the data that was used as well as how the parameter optimization and evaluation was performed.

3.1 Implementation

To get the distance between two points the Haversine formula was used according to

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{3.1}$$

The distance was calculated between all points so the number of evaluations of the Haversine scales as $\binom{n}{2}$ where n is the total number of data points. Increasing the performance of the Haversine therefore had a big impact of the total runtime. To increase the performance $\sin(\theta_2 - \theta_1)$ was approximated with $(\theta_2 - \theta_1)$ since the difference in the latitude and longitude of two points was small. Furthermore $\cos(\theta_2 - \theta_1)$ was approximated with $\cos(\theta_2 - \theta_1)) \approx 1$ which gave

$$d = 2R\sqrt{\left(\frac{\phi_2 - \phi_1}{2}\right)\left(\frac{\phi_2 - \phi_1}{2}\right) + \frac{1}{2}\left(\cos(\phi_2 + \phi_1) + 1\right)\left(\frac{\lambda_2 - \lambda_1}{2}\right)\left(\frac{\lambda_2 - \lambda_1}{2}\right)} \tag{3.2}$$

3.1.1 Inverse Distance Weighting

The IDW was implemented according to the theory and the two parameters, α and the number of neighbours, was used as inputs to be optimized.

3.1.2 Ordinary Kriging

When implementing the OK the number of neighbours and the number of variograms (i.e. the number of directions) was used as inputs to be optimized.

One important step when implementing Kriging is the fitting of the variogram. This can be done in a number of different ways and the chosen method was to implement a genetic algorithm for the fitting as suggested by Shaohua and Wentao [16]. Each of the three models, stable exponential, spherical and power exponential, was fitted and the one with the lowest root mean square error was chosen as the best variogram. The root mean square was only calculated from the points shorter than the range of the variogram [16].

An important step is to choose a good bin length. Since irregular scattered data will be the source the bin size was set to be the average distance between sampling points as Oliver concluded and suggested [10].

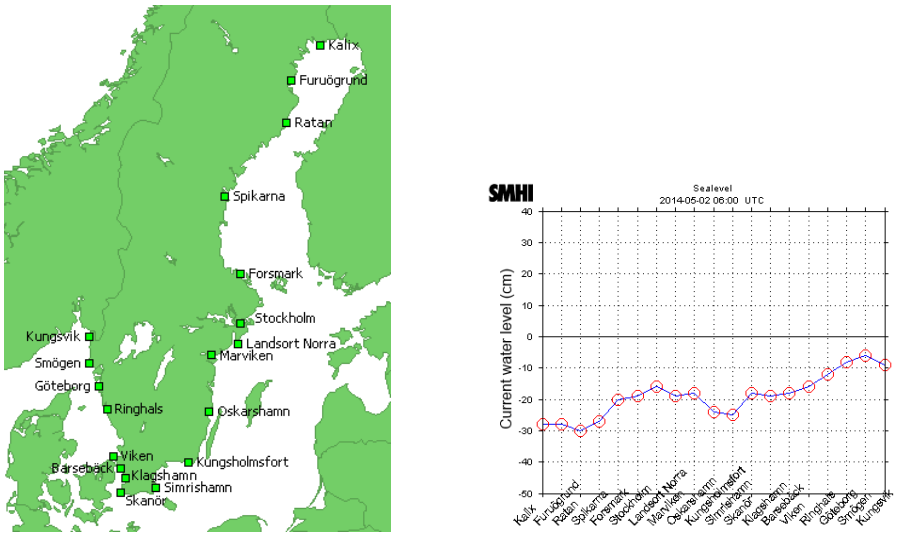
3.1.3 Regularized Spline with Tension

When implementing the RST, k_{max} and k_{min} was fixed at $k_{max} = 300$, $k_{min} = 100$ which is suggested by Mitasova [13]. The smoothing parameters and the tension was set as inputs to be optimized. Horizontal scaling and rotating of the axis was not implemented, however vertical scaling was implemented and evaluated.

3.1.4 Water Level Retrieval

To be able to account for changing water level over time a reference level as well as a source providing historical data of the water level was needed. There is a determined reference for the water level in Sweden from which the depth in the charts are given. Since 2005 the reference level in Sweden is given by *RH2000, (Rikets Höjdsystem 2000)* [17] which was used.

Swedish Meteorological and Hydrological Institute, SMHI, provides historical data of the sea level from 21 measurement stations around the coast of Sweden. The data consist of longitude, latitude and measurements from when the station was installed to present time [18, 19]. The data is given with RH2000 as reference which is the same as modern nautical charts. Since the measurements are rather close, the water level can be approximated as a linear function between two measurements. Given the position of a measurement the closest station on each side (along the coast) was found. In Sweden the water level is approximately linear between two stations and by interpolating between these two stations and in time the water level at an arbitrary point could be found [18]. The positions of the stations can be seen in figure 3.1(a) and an example of output data can be seen in figure 3.1(b).



(a) (b)

Figure 3.1: (a): The position of the stations of SMHI. (b): An example of the water level at different stations.

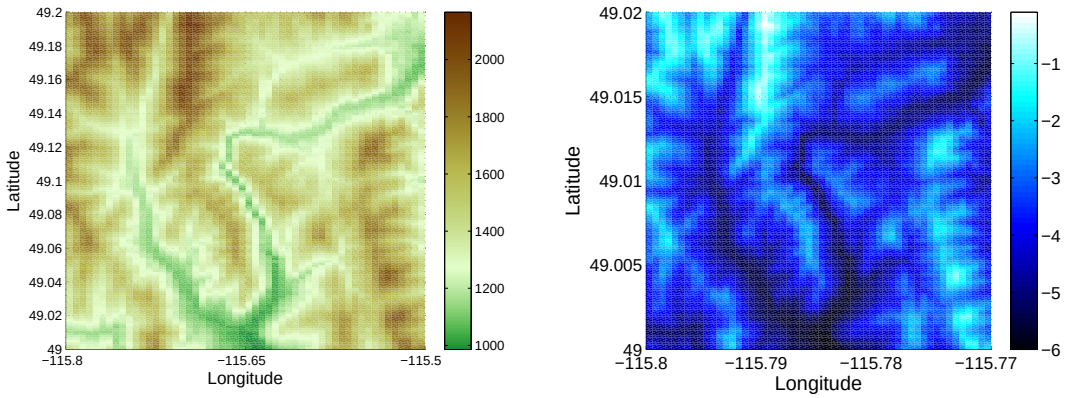
3.2 The Data

Throughout this project no real data from sonar was available due to the fact that the prototype was not yet ready and no data had been collected. There are also some legal aspect on collecting sea level data. The Swedish Defence Force (*Försvarsmakten*) have to give permission to collect data and another government authority has to give permission to store the data.

However, the algorithms developed can also be applied to elevation data. Elevation data from NASA [1] over an area in Canada was used instead of data from sonars. The error in this data, both in position and elevation, was considerably larger than those in data from a sonar and GPS-unit. To handle this the data was scaled until a satisfactory error was achieved. The elevation data was also translated so that the lowest value was at 0m and the elevation above sea level was interpreted as depth. The surface of the original data can be seen in Figure 3.2(a) while the transformed data can be seen in Figure 3.2(b). Specifications for the original data and the transformed data can be seen in Table 3.1. In total, the data contains 7975 measurement points in a grid with size 145×55 (latitude×longitude).

From the data five different sets with various sizes was created, see Table 3.2. The four largest sets was used for the interpolation (training) while the smallest set was used for validation.

The three smallest training sets, S_5, S_{10} and S_{20}, can be seen in Figure 3.3(b)-3.3(d) and the validation set in Figure 3.3(a).



(a) Original (b) Transformed

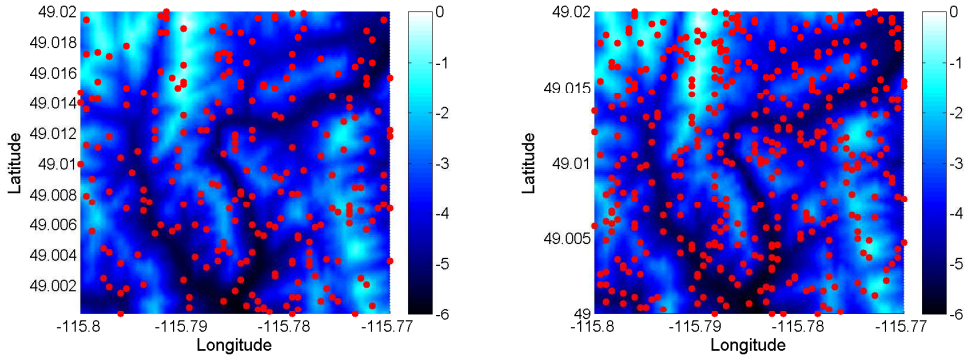
Figure 3.2: (a): The original elevation data from an area outside Fernie, Canada. The plot shows the elevation above the sea level. (b): The translated data where the position is scaled and translated according to Table 3.1. The plot represents the depth.

Table 3.1: Specifications for the original data set as well as the transformed. Plots of the data-set can be seen in Figure 3.2(a) and 3.2(b).

	Latitude width (km)	Longitude width (km)	Min. elev. (m)	Max. elev. (m)	Pos. error (m)	Elev. error (m)
Original	22.22	21.87	986	2166	50	30
Transformed	2.22	2.19	0	5.9	5	0.15

Table 3.2: The five different data sets that were used. The sets S_5, S_{10}, S_{20} and S_{50} were used for interpolation while S_3 was used for validation. None of the points in S_3 is in the other four sets while $S_5 \in S_{10} \in S_{20} \in S_{50}$. Mean distance denotes the mean distance to the closest neighbour in the same set and the size is the relative size to the original set. Plots of the sets can be seen in Figure 3.3(a)-3.3(d).

Set	Size	Data points	Mean distance (m)
S_3	3%	239	76
S_5	5%	399	59
S_{10}	10%	798	43
S_{20}	20%	1595	31
S_{50}	50%	3988	20



(a) Validation set, S_3: 3%

(b) Training set, S_5: 5%

(c) Training set, S_{10}: 10%

(d) Training set, S_{20}: 20%

Figure 3.3: (a): The validation set containing 3% of the original data. (b)-(d): The training sets used for the interpolation. The depth is in meters and specifications for the sets can be seen in Table 3.2.

3.3 Parameter Optimization

To be able to make a fair comparison between the different algorithms they all had to be optimized. For IDW only two parameters had to be optimized, the number of neighbours and the distance decay parameter α. For OK the number of neighbours as well as the number of variograms, which determines the angle of each lag, was optimised. The RST method is much more complex and contains a lot of parameters, for example the tension (φ), the smoothing (w), k_{max}, k_{min}, vertical scaling e.t.c. But since the optimization was rather demanding only the tree parameters with biggest impact on the error was optimized, namely the vertical scaling, the tension and the smoothing [14].

To optimize the parameters a wide range of parameters was tested and evaluated. By narrowing down at the best parameters a good range could be found. The parameter setting which gave the smallest error using cross validation was considered to be the best set up. To minimize the risk of overfitting, that is the risk that the optimal parameters only is the optimal for the given evaluation set, a second validation set was used, S_{3b}. This third set also contained 3% of the original data, but different from S_3.

3.4 Evaluation

The quality of the output data will depend on a number of factors. These factors can be divided into three groups:

- Accuracy, density and distribution of the source data
- The interpolation process
- Characteristics of the seabed

The first two of these can be considered to give errors while the third one can be seen as an uncertainty [8], for example a smooth seabed tends to decrease the uncertainty. The only one of these that can be affected was the interpolation process which is why the different methods that was implemented needed to be evaluated and compared. And even though two other groups cant be affected they could still be analysed to see how they affect the error and uncertainty of the output.

The error from the interpolation process can arise from two different sources

- Error from the algorithm
- Error in the measured data propagating through the algorithm

The analysis is divided into two parts. In the first part the error from the interpolation process is examined for different data densities and distributions of the source data. In the second part the error propagation for the three different interpolation methods is examined.

3.4.1 Error from the Interpolation

The error from interpolation was calculated for each of the four data-sets containing 5%, 10%, 20% and 50% of the original data. S_3 was used for validation and considered to contain correct data. The error was defined as the difference between the known value in S_3 and an interpolated value in at the same position. The maximum, minimum, mean and root mean square (RMS) error was calculated for each data set for each of the three algorithm.

Before the error was calculated the parameters of the algorithms was optimized using the method described in Section 3.3.

Finally the errors in all points was plotted against the distance to the closest neighbour. The points was then divided into bins with a width of 10m. By placing points in each bin with 95% of the datapoints below and 5% above and fitting a curve to these point an empirical curve telling the error as a function of distance with 95% accuracy could be obtained. This was done for all methods and set sizes.

3.4.2 Error Propagation

The error of the input data will propagate through the interpolation to the output. Therefore it is important to keep a record of the accuracy and analyse the model to ensure satisfactory accuracy in the output [20].

Each data point used for training had three variables: the latitude, the longitude and the depth. And for a given latitude and longitude at an interpolation point there was one output parameter, namely the estimated depth. A first order analysis of the uncertainty was done by linearising the interpolation function and analyse how an error propagated.

Analytical Analysis

The analytical analysis was performed by linearising the algorithm and looking at how errors would propagate. To evaluate the linearisation a simple set of synthetic data was used and the result was compared with the result from a Monte Carlo simulation.

Numerical Analysis

To analyse how errors propagate through the algorithms numerically Monte Carlo simulations was performed. A number of interpolations was made where the latitude, longitude and depth for each point and interpolation was randomly changed within the interval of the uncertainty.

By comparing the maximum difference in each point between the interpolation from the original data and the interpolation from the Monte Carlo simulations the maximum error of the output given the uncertainty in the input data could be calculated.

A fitted curve describing the error as a function of distance to the closest points was done the same way it was done for the error from the interpolation.

3.4.3 Runtime

An important aspect when it comes to implementations of the algorithms is the runtime and how they scale with increasing number of data points. Therefore a runtime comparison between the algorithms was performed.

4 Results

In this chapter the results from the parameter optimization, the error from interpolation, the error propagation and the runtime are presented.

4.1 Variogram Fit

Plots are produced for visual evaluation of the genetic algorithm and the fitting of the variogram. An example can be seen in Figure 4.1(a)-4.1(c) along with the convergence of the genetic algorithm in Figure 4.1(d)-4.1(c). The figures illustrates the three types of variograms used and each plot represents a variogram in a specific direction. For this example S_{50} is used for the interpolation and in each case the given model is the one with the lowest root mean square error for the given data. The genetic algorithm consisted of 100 individuals over 200 generations.

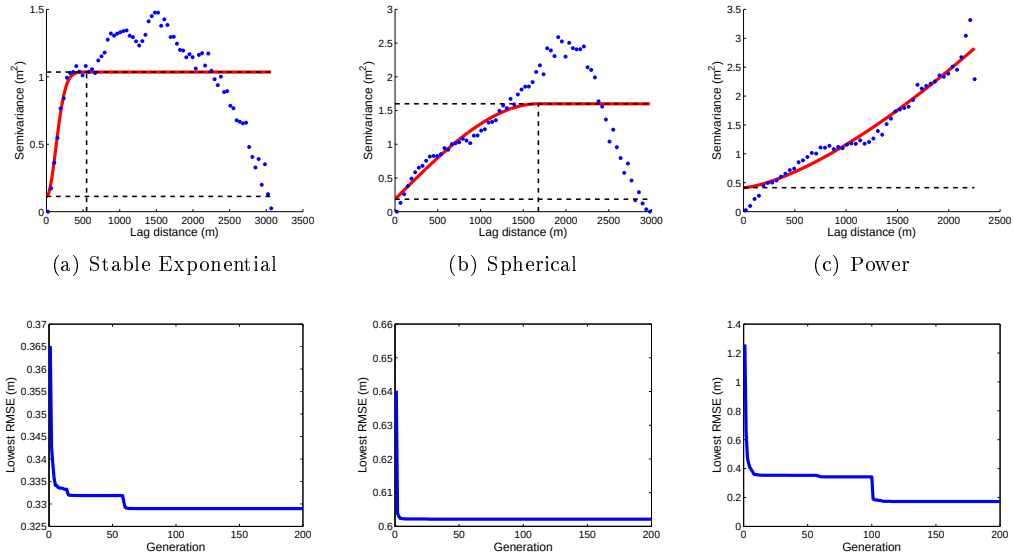
4.2 Parameter Optimization

An example of the parameter optimization using S_{20} for the interpolation and S_3 for training can be seen in Figure 4.2(a), 4.2(c) and 4.2(e). In each point S_{20} is used to interpolate the values and the RMS error is calculated as the difference between this and the known value in S_3. A third set, S_{3b} also containing 3% of the data points, different from S_3, is used to validate the optimization which can be seen in Figure 4.2(b), 4.2(d) and 4.2(f).

For IDW the number of neighbours is varied from 1 to 15 and α from 1 to 5. The optimal parameters found for the four different training sets can be seen in Table 4.1.

The optimal OK parameters can be seen in Table 4.2 where the number of neighbours is ranging from 1 to 15 and the number of variograms from 1 to 8.

For RST the smoothing is varied from 0 to 0.05 and the tension from 0.01 to 0.025. The result can be seen in Table 4.3.



(a) Stable Exponential (b) Spherical (c) Power

(d) Stable Exponential (e) Spherical (f) Power

Figure 4.1: (a)-(c): Examples of each of the three variogram models implemented. The variograms is created from S_{50} and fitted using a genetic algorithm. The lower horizontal dashed line represents the nugget while the upper represents the sill. The vertical dashed line represents the range. Note that the power model does not have any sill or range. (d)-(f): The corresponding convergence for the genetic algorithm.

4.3 Error from the Interpolation

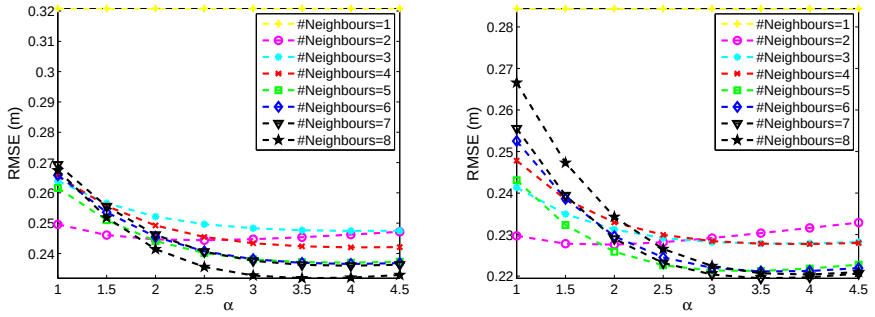
The parameters used for the evaluation is the optimised parameters which can bee seen in Table 4.1, 4.2 and 4.3.

The maximum (+), minimum (-), mean and root mean square error for the four different data sets for all three interpolation methods can be seen in Table 4.4.

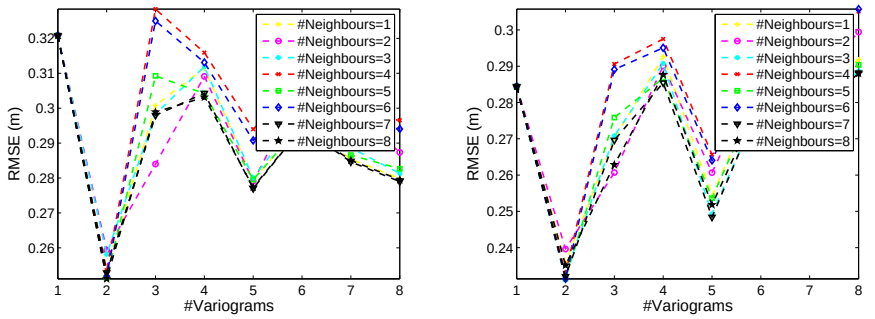
In Figure 4.3 the RMS-error as a function of the density of data-points can be seen for the three different interpolation methods.

A visualisation, using a surf, of the result using S_{20} for the interpolation can be seen in Figure 4.4(a)-4.4(c) along with the correct surface from 100% of the original data in Figure 4.4(d). Contour plots of the results using S_{20} can be seen in Figure 4.5(a)-4.5(c) along with the correct contour in Figure 4.5(d). The absolute error, comparing the interpolated value and all the known values, can be seen in Figure 4.6(a)-4.6(l).

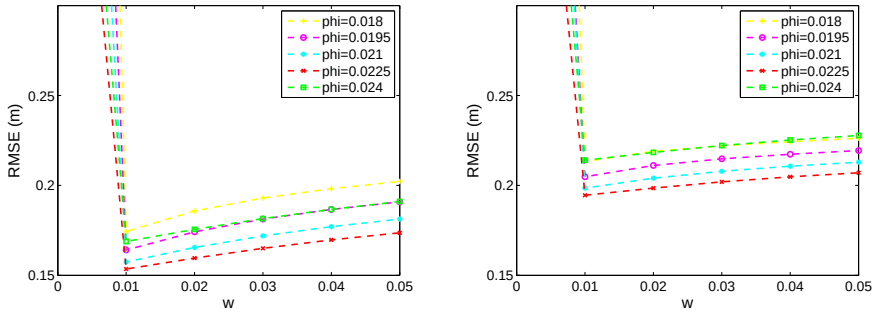
An example of all the errors plotted against the distance to the closest point can be seen in Figure 4.7 where the S_5 was used for interpolation and the interpolation method used was IDW.



(a) IDW S_3 (b) IDW S_{3b}



(c) OK S_3 (d) OK S_{3b}



(e) RST S_3 (f) RST S_{3b}

Figure 4.2: An example of the optimization for the parameters where S_{20} was used for the interpolation while S_3 was used for training the optimization. A third set was used for validating the result, S_{3b}. The left hand side shows the optimization when S_3 is used and the right hand side shows the validation using S_{3b}.

The fitted curve for all thee interpolation methods can be seen in Figure 4.8(a)-4.8(d) for the four different sets.

Table 4.1: The optimal parameters for the IDW method that was found when using cross validation with S_3 as validation set. The number of neighbours was tested for 1 to 15 with step 1 while α was tested for 1 through 5 with step 0.5.

Training set	α	Number of neighbours
S_5	2.50	6
S_{10}	3.00	10
S_{20}	3.50	8
S_{50}	3.00	8

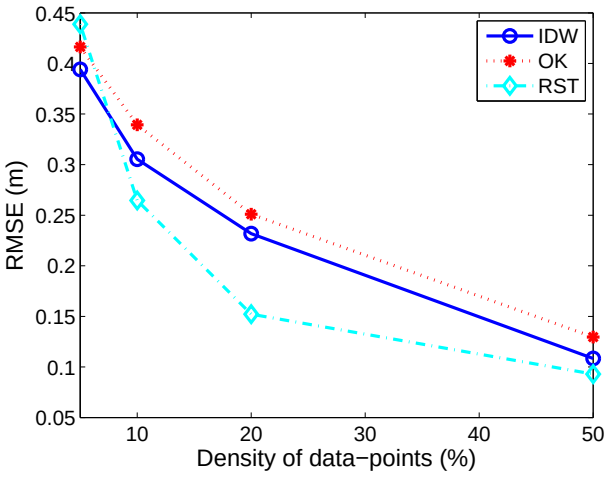


Figure 4.3: The RMS-error for the three different data sets using the 3%-set as validation.

Table 4.2: The optimal values and the variogram models and their corresponding values found for the OK method using S_3 as validation. The number of neighbours was tested for 1 to 15 with step 1 and the number of variograms was tested for 1 through 8 with step 1. A genetic algorithm was used to fit an exponential, power or spherical varigoram model to the data.

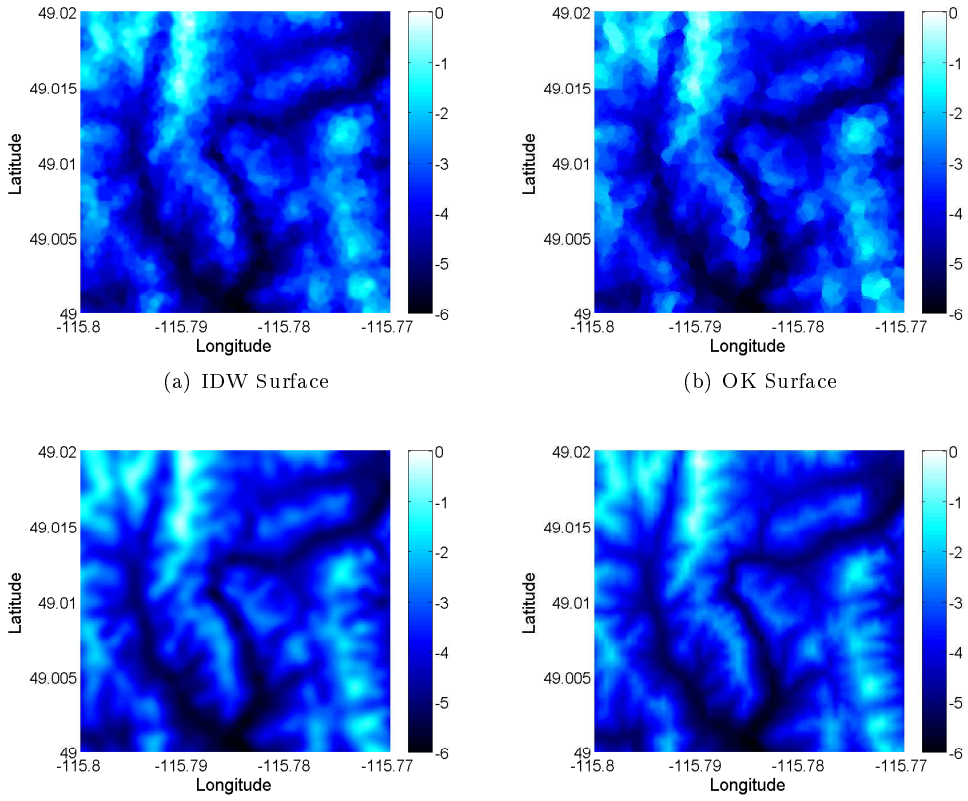
Training set	Number of neighbours	Number of variograms	Model	Range (m)	Nugget (m^2)	Sill (m^2)
S_5	5	1				
			Exponential	712	0.00	1.08
S_{10}	6	8				
			Exponential	958	0.16	1.14
			Exponential	425	0.00	0.76
			Exponential	886	0.13	1.06
			Exponential	6025	0.37	3.00
			Exponential	6025	0.37	4.19
			Exponential	3844	0.35	2.56
			Exponential	598	0.00	1.03
			Spherical	560	0.05	1.27
S_{20}	7	2				
			Exponential	548	0.11	1.04
			Spherical	1677	0.18	1.18
S_{50}	3	8				
			Exponential	1234	0.10	1.18
			Exponential	457	0.00	0.70
			Spherical	664	0.04	0.97
			Exponential	5224	0.30	2.66
			Power	-	0.03	-
			Spherical	2367	0.01	2.36
			Exponential	695	0.00	1.01
			Exponential	1087	0.08	1.32

Table 4.3: The optimal values for the RST-method found when the tension (φ) ranged from 0.010 to 0.025 with step 0.0005 and the smoothing (w) from 0 to 0.05 with step 0.01. S_3 was used for validation.

Training set	ω	φ	Vertical scaling
S_5	0.01	0.0100	1
S_{10}	0.03	0.0195	1
S_{20}	0.01	0.0225	1
S_{50}	0.01	0.0240	1

Table 4.4: The result for the error from the interpolation when using the 3%-set for validation.

Method	Training set	Maximum (+) error (m)	Minimum (-) error (m)	Mean error(m)	RMS error (m)
IDW	S_5	1.3428	-0.9786	0.0429	0.3941
OK	S_5	1.2009	-1.0912	0.0273	0.4163
RST	S_5	2.8562	-1.1911	0.0196	0.4388
IDW	S_{10}	0.7540	-1.1082	0.0014	0.3053
OK	S_{10}	0.8630	-1.1235	-0.0035	0.3394
RST	S_{10}	0.9726	-0.9513	0.0003	0.2646
IDW	S_{20}	0.7807	-0.7175	0.0153	0.2318
OK	S_{20}	1.1837	-0.9550	0.0099	0.2510
RST	S_{20}	0.5584	-0.6766	-0.0034	0.1521
IDW	S_{50}	0.4052	-0.6429	0.0007	0.1084
OK	S_{50}	0.5344	-0.7053	0.0068	0.1295
RST	S_{50}	0.3733	-0.3230	0.0026	0.0929



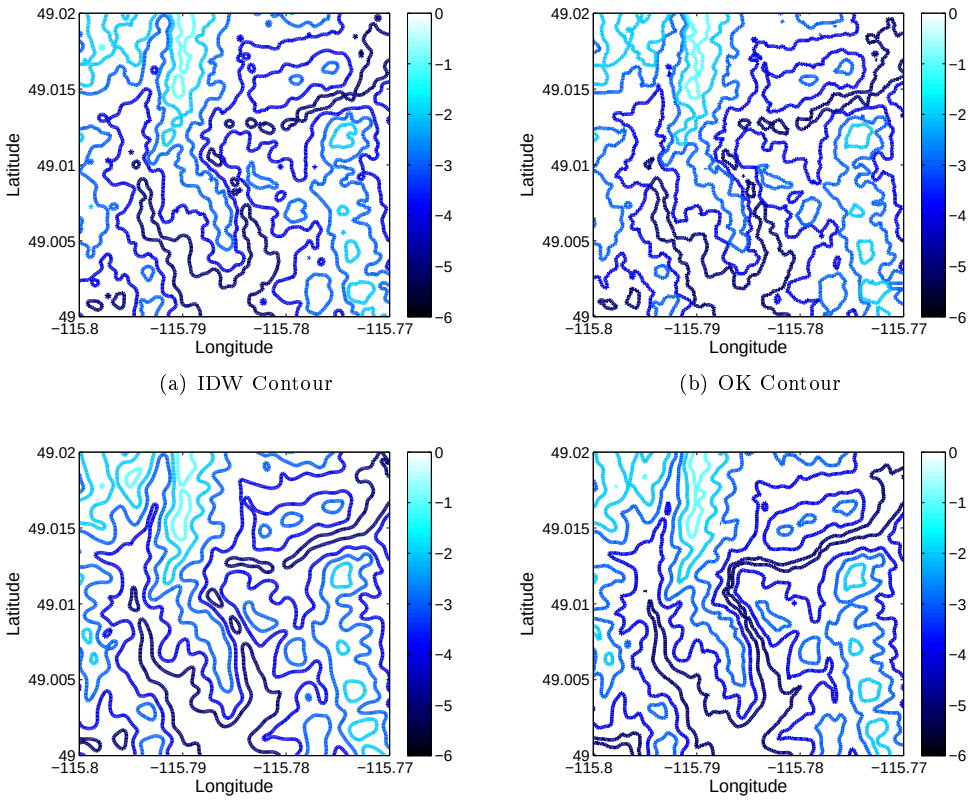
(a) IDW Surface

(b) OK Surface

(c) RST Surface

(d) A surf of the original data

Figure 4.4: (a)-(c): The resulting surface when using S_{20} for the interpolation. (d): The correct surface created from 100% of the data. The depth is meters.



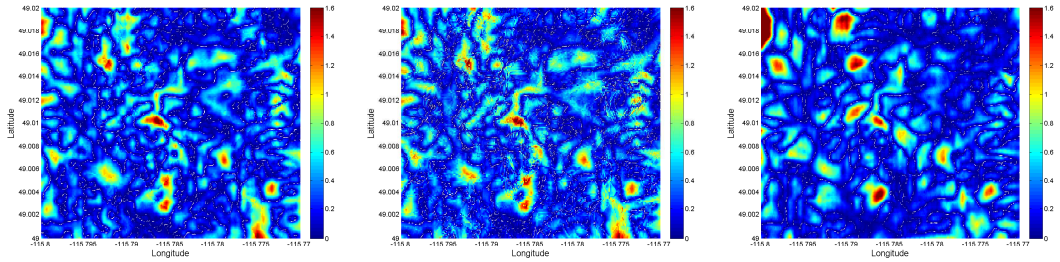
(a) IDW Contour

(b) OK Contour

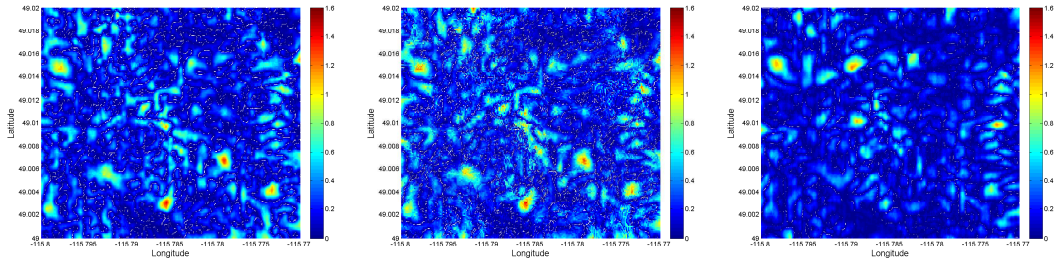
(c) RST Contour

(d) A contour of the original data

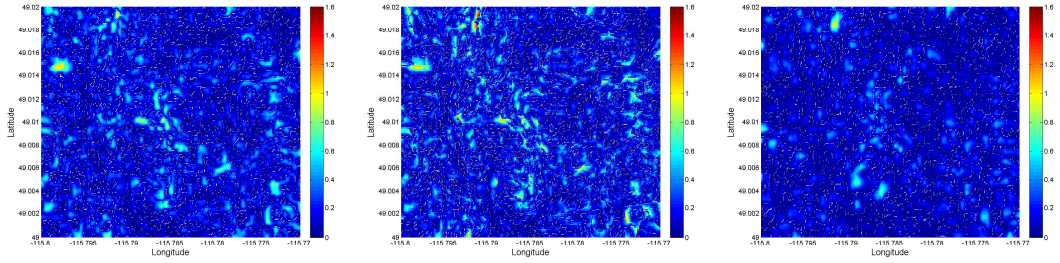
Figure 4.5: (a)-(c): The resulting contours when using S_{20} for the interpolation. (d): The correct contours created from 100% of the data. The depth is in meters.



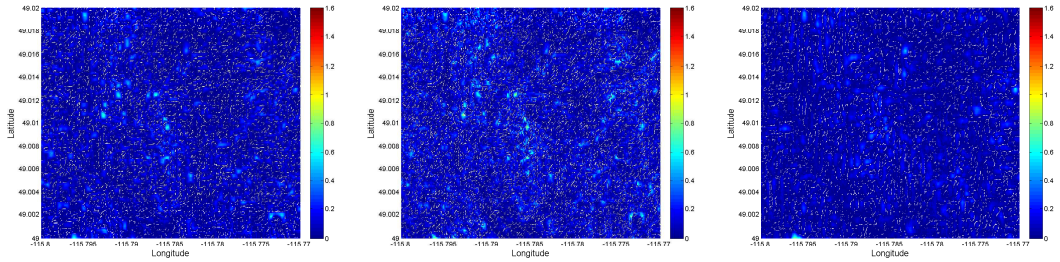
(a) IDW S_5 (b) OK S_5 (c) RST S_5



(d) IDW S_{10} (e) OK S_{10} (f) RST S_{10}



(g) IDW S_{20} (h) OK S_{20} (i) RST S_{20}



(j) IDW S_{50} (k) OK S_{50} (l) RST S_{50}

Figure 4.6: The absolute error from the interpolation, in meters, calculated as the difference between the known value and the interpolated value.

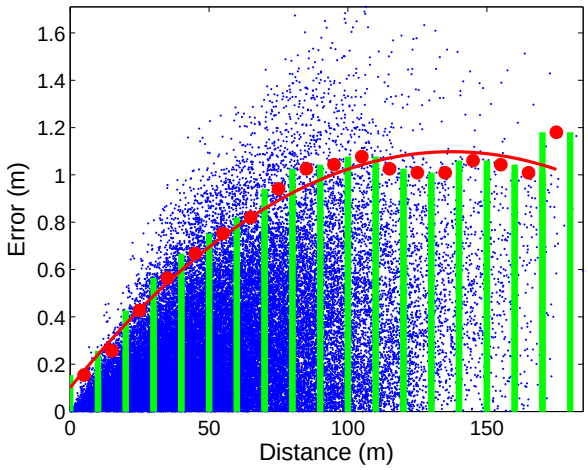
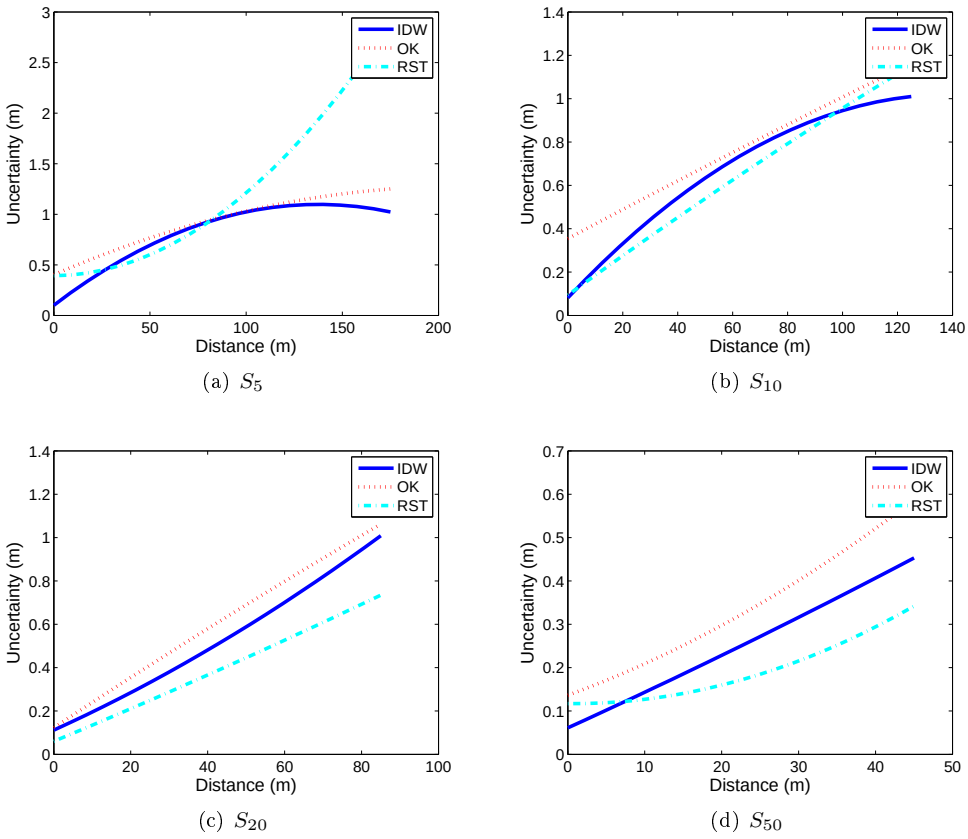


Figure 4.7: The errors plotted against the distance to the closest neighbour using S_5 and IDW. All the points was placed in bins with a width of 10m. In each bin a point is placed which is lovated above 95% of the points and a second degree polynomial is fitted to these points.



(a) S_5 (b) S_{10}

(c) S_{20} (d) S_{50}

Figure 4.8: The error from the interpolation as a function of distance to the closest known point for the different training sets and interpolation methods.

4.4 Error Propagation

This section is divided into two parts. In the first part the results from the analytical analysis is presented for IDW and in the second part the numerical result for all methods is presented.

4.4.1 Analytical Analysis

The input-parameters to the IDW consists of $x = x_{lat}$, $y = x_{long}$ and $z = depth$ for each of the n closest neighbours. To simplify the notation the function is written as

$$\hat{z}(\mathbf{x}_0) = F(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}_i) \tag{4.1}$$

where $\hat{z}(\mathbf{x}_0)$ is the estimate depth at $\mathbf{x}_0$ and

$$f_i(\mathbf{x}_i) = f_i(x_i, y_i, z_i) = \frac{\left[(x_i - x_0)^2 + (y_i - y_0)^2\right]^{-\frac{\alpha}{2}} z_i}{\sum_{j=1}^{n}\left[(x_j - x_0)^2 + (y_j - y_0)^2\right]^{-\frac{\alpha}{2}}} \tag{4.2}$$

Linearisation of $f(\mathbf{x})$ is equal to linearise the functions $\{f_i\}$ according to

$$f_i(x_i + \delta x_i, y_i + \delta y_i, z_i + \delta z_i) \approx f_i(x_i, y_i, z_i) + \frac{\partial f_i}{\partial x}\delta x + \frac{\partial f_i}{\partial y}\delta y + \frac{\partial f_i}{\partial z}\delta z \tag{4.3}$$

where the derivatives are evaluated in the point (x_i, y_i, z_i). Assume that δx_i, δy_i and δz_i are independent random variables with zero mean and standard deviations σ_x, σ_y and σ_z. The standard deviation in the estimated value can, for small deviations, be approximated as

$$\sigma_z = \sqrt{\sum_{i=1}^{n}\left[\left(\frac{\partial f_i}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial f_i}{\partial y}\sigma_y\right)^2 + \left(\frac{\partial f_i}{\partial z}\sigma_z\right)^2\right]} \tag{4.4}$$

which will depend on the position of all the known points relative to the interpolation point, their corresponding depth and the choice of the parameters n and α.

Empirical studies on depth data has shown that typical parameters is $\alpha = 2$ and $n = 8$ but the error will still depend on the position of all the known points relative to the position of the interpolation point.

To test how well the linearisation and analytical analysis performs a simple data set is created which can be seen in Table 4.5.

To evaluate the analytical analysis it is compared to a numerical Monte Carlo analysis of the propagation. Two different sets of uncertainties was used, in the first simulation $\sigma_x = \sigma_y = 0.2, \sigma_z = 0.1$ while in the second $\sigma_x = \sigma_y = 2, \sigma_z = 0.1$.

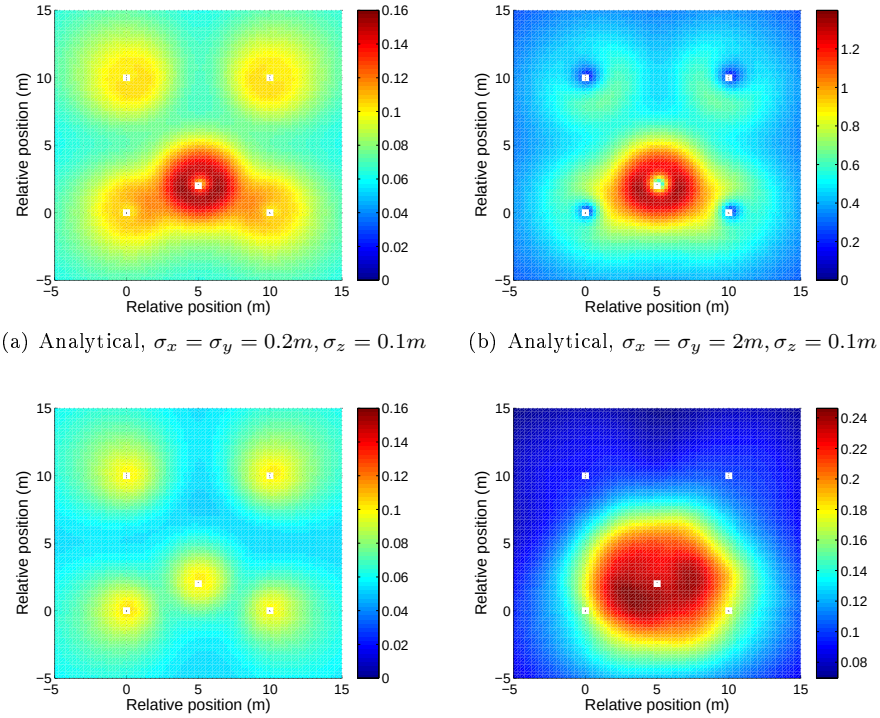
The analytical uncertainty in the output using the linearisation as well as an Monte Carlo simulation can be seen in Figure 4.9(a)-4.9(d). The error calculation using Monte Carlo simulation was taken as the standard deviation in each point using 1000 independent simulations. In each

Table 4.5: Synthetic data used for the analysis of the error propagation.

Y-position (m)	X-position (m)	Depth (m)
0	0	2
0	10	2
10	10	2
10	0	2
5	2	3

simulation, each of the five points was randomly disturbed using a normal distribution with the given standard deviation.

Analytical calculations was not performed on real data since it is shown that the linearisation was not good enough to be used. Analytical calculations was not performed on the two other methods, OK and RST, since they are too complex to linearise.



(a) Analytical, $\sigma_x = \sigma_y = 0.2m, \sigma_z = 0.1m$ (b) Analytical, $\sigma_x = \sigma_y = 2m, \sigma_z = 0.1m$

(c) Monte Carlo, $\sigma_x = \sigma_y = 0.2m, \sigma_z = 0.1m$ (d) Monte Carlo, $\sigma_x = \sigma_y = 2m, \sigma_z = 0.1m$

Figure 4.9: (a) and (b) shows the maximum error in the output using an analytical calculation. (c) and (d) shows the corresponding errors using Monte Carlo simulations. Note the different scale in (b) and (d).

Table 4.6: The result for the error from the interpolation when using the 3%-set for validation. The maximum and mean over-/undershoot is given. Overshoot means that the algorithms estimates a depth deeper then the correct value and undershoots a more shallow estimation.

Method	Training set	Maximum over. (m)	Maximum under. (m)	Mean over. (m)	Mean under. (m)	RMS error (m)
IDW	S_5	0.5147	0.5779	0.0906	0.0899	0.0982
OK	S_5	0.8438	0.9077	0.1129	0.1104	0.1399
RST	S_5	0.4990	0.4383	0.1163	0.1216	0.1299
IDW	S_{10}	0.3487	0.3516	0.0920	0.0927	0.0995
OK	S_{10}	0.7478	0.6620	0.0894	0.0898	0.1124
RST	S_{10}	5.2284	6.0708	0.1764	0.1604	0.3567
IDW	S_{20}	0.3984	0.4005	0.1016	0.1006	0.1092
OK	S_{20}	1.5905	1.4176	0.1223	0.1231	0.1535
RST	S_{20}	7.0000	2.3488	0.1385	0.1361	0.1627
IDW	S_{50}	0.3489	0.3279	0.0982	0.0982	0.1056
OK	S_{50}	1.2202	1.1133	0.1444	0.1435	0.1813
RST	S_{50}	0.4966	0.3670	0.0840	0.0842	0.0909

4.4.2 Numerical Analysis

A table containing the maximum-, minimum-, mean- and root mean square-error for all three interpolation methods and data sets can be seen in Table 4.6. The result is based on the output from 100 Monte Carlo simulations for each algorithm and data set. In each Monte Carlo simulation the position of each data point was changed independently using a normal distrubution with zero mean and $\sigma = 5m$. The depth of each data point was also changed using $\sigma = 0.2m$. In each point the algorithm can overshoot or undershoot the correct value. Overshooting means that the algorithm estimates a depth deeper than the correct value. Undershoot means that the estimation is more shallow than the correct value.

In Figure 4.10 the RMS-error as a function of the density of data-points can be seen for the three different interpolation methods.

The uncertainty for the different interpolation method and data sets was plotted against the corresponding position which can be seen in Figure 4.11(a)-4.11(c).

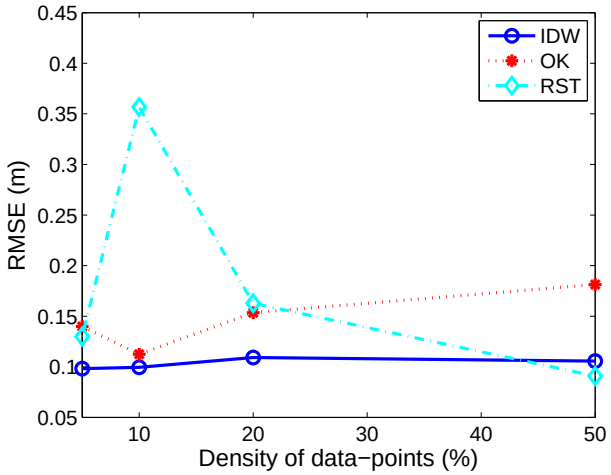
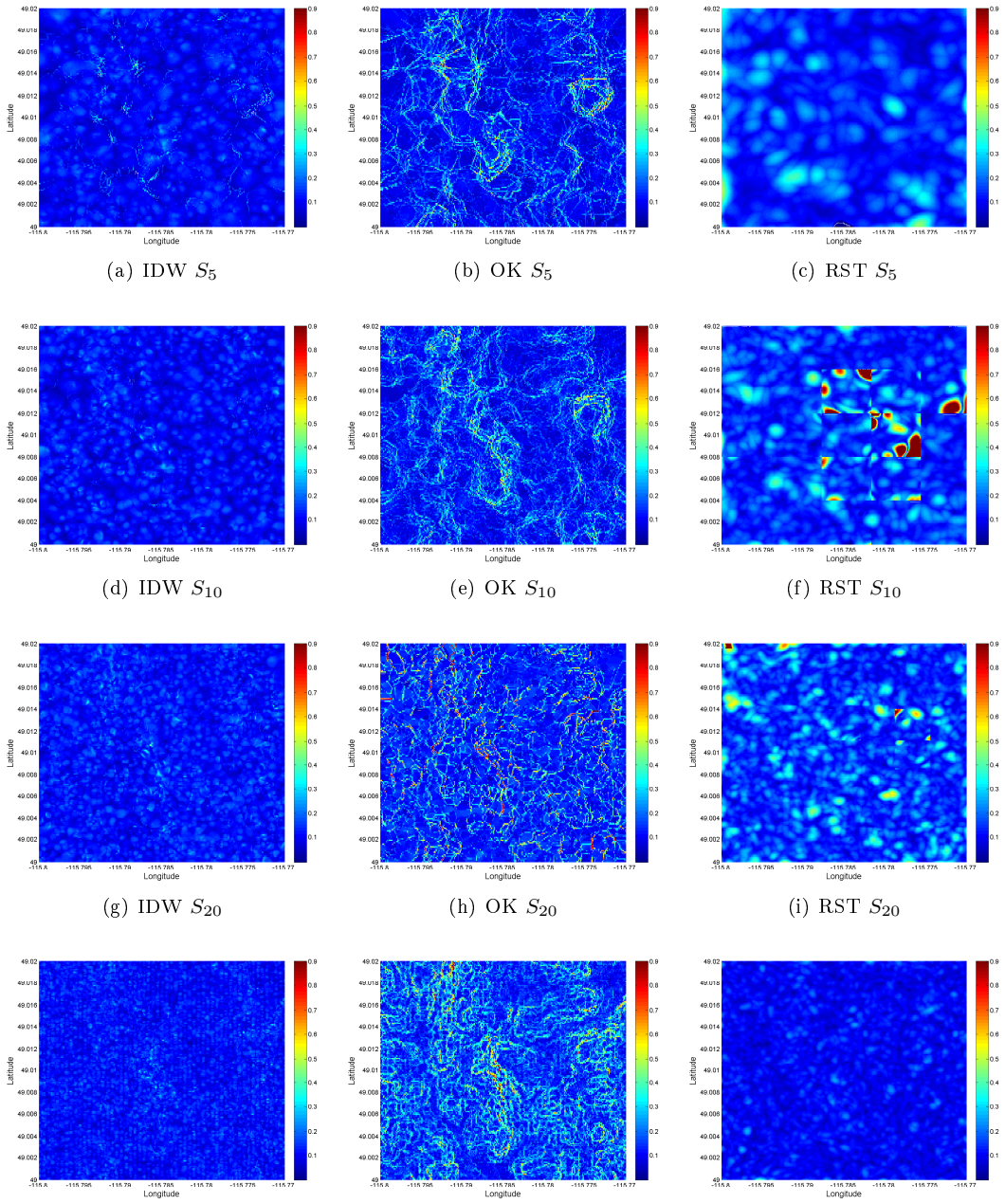


Figure 4.10: The RMS-error for the three different data sets using the 3%-set as validation.



(a) IDW S_5 (b) OK S_5 (c) RST S_5

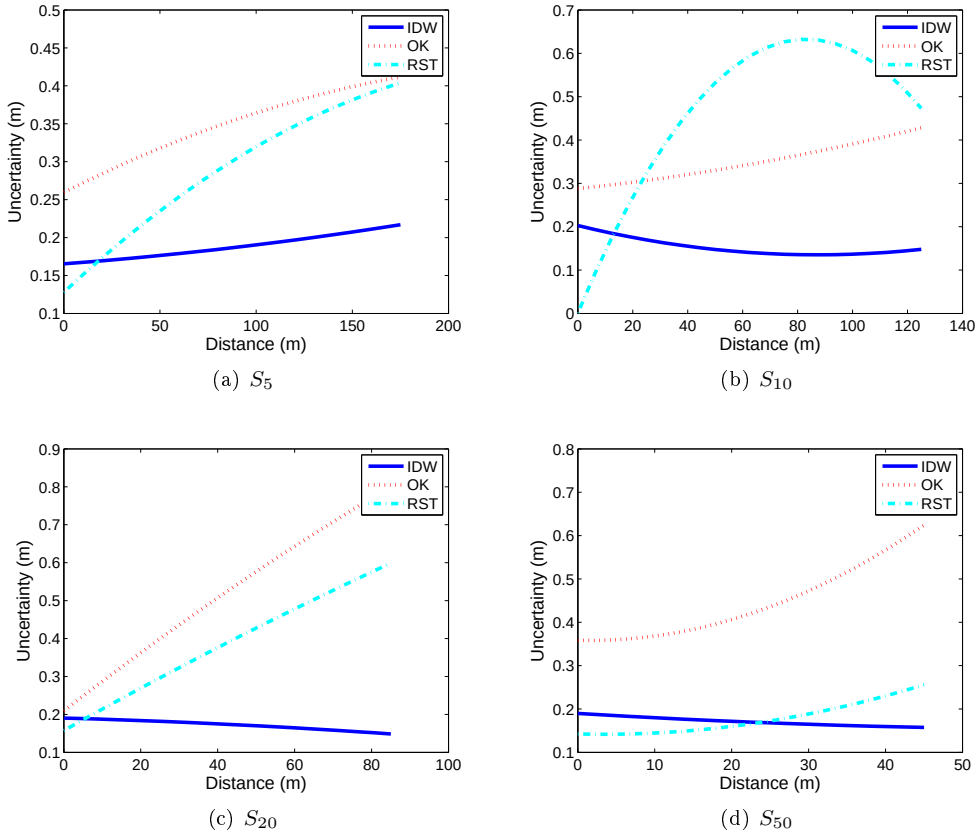
(d) IDW S_{10} (e) OK S_{10} (f) RST S_{10}

(g) IDW S_{20} (h) OK S_{20} (i) RST S_{20}

(j) IDW S_{50} (k) OK S_{50} (l) RST S_{50}

Figure 4.11: The maximum error propagation, in meters, calculated as the difference between the known value and the value from the Monte Carlo simulations. The square pattern in 4.11(f) arises from bad parameters and from the fact that the input data was divided into a grid.

The fitted curve for all three interpolation methods can be seen in Figure 4.12(a)-4.12(d) for the four different sets.



(a) S_5 (b) S_{10}

(c) S_{20} (d) S_{50}

Figure 4.12: The error as a function of distance to the closest known point for the different training sets and interpolation methods.

4.5 Runtime

The runtime of the different algorithms as a function of the number of data points can be seen in Figure 4.13. The runtime is taken as the mean over five simulations. Only the interpolation process is considered, the time for optimisation is not represented. The program was tested on a computer with a dual core 1.9 GHZ processor and 10 GB of ram.

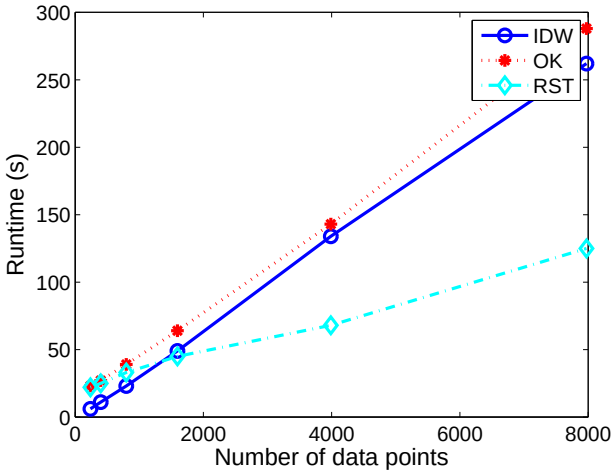


Figure 4.13: The runtime for the three diferent interpolation methods as a function of distance. The time is taken as the mean over five different runs.

5 Discussion

The following chapter contains a discussion of the results and suggestions for improvements.

5.1 Variogram Fit

A part from the root mean square error the genetic algorithm and the variogram fit was also evaluated by visualisation. Samples were taken and in Figures 4.1(a)-4.1(f) it can be seen that the algorithm converged and that the variogram fits the sampled data below the range. There is no need to fit the data above the range since the range determines at what distance data points no longer are correlated. Improvements here can be to implement more variogram models.

5.2 Parameter Optimization

First of all, it can be seen in Figures 4.2(a)-4.2(f) that for all three methods the optimal parameters were the same for the two different validation sets. This indicates that the parameters should be able to be generalized to be the optimal parameters for the given area. It should however be noted that data from another area can, and probably will, have different optimal parameters. The reason for this is for example that a mountain bottom is much less smooth than a sand bottom. In future research more areas could be examined and the different parameters can be compared.

It should also be noted that the parameters was optimized for the smallest error in the interpolation method, not the smallest error propagation. Further work here could include finding a balance between parameter optimized for small interpolation error and small error propagation.

Overall the parameter optimization algorithm implemented is not very intelligent since it only iterates over the given parameter space. A solution could be to look at the variogram and find pattern in this to make a good guess on the choice of the parameters. In the end, the parameters should only depend on the properties of the sea floor, where some properties can be represented by a variogram. The type of ocean floor (sand, mud, bedrock, vegetation e.c.t.) could also be used as input and indicate which parameters to use to speed up the process.

When looking at the variogram, see figure 4.1(a)-4.1(b), one can also notice a rather significant difference in different directions. This indicates that the correlation between points is different depending on the direction. This motivates a change in scale for the two horizontal coordinates in RST which gives different parameters in different directions.

For IDW it can be seen that when the number of neighbours is equal to 1, the RMSE is independent of α which was expected since α tells how the neighbours should be weighted, and if we only have one, α has no impact. The error curves are also rather smooth.

In OK on the other hand, the curves are not that smooth and one can find many local minima. The most realistic number of variograms should be 2 since the variogram indicates in how many different directions different properties exists. These two directions can for example be parallel to the coast and perpendicular to the coast. The result however indicated that more variograms should be used, up to 8 in two cases.

For RST the optimal parameters were very hard to find since there do not exist any general range or limits for the parameters. It was however noted that the vertical scaling did not have any impact on the error. When optimizing the RST parameters a lot of manual work was done and trial and error was used to find good ranges for the parameters. Wrong parameters gave a very unstable algorithm where the depth diverged for all points expect for the known points. The vertical scaling is there to handle a large difference in scale in position and depth, but since this is not the case the vertical scaling had no noticeable effect.

It can be noted that the smoothing parameter for RST is roughly the same for all densities. The smoothing parameter tells how smooth the original surface should be, so this should be density independent, which was shown. The tension parameter on the other hand increased with increasing density. The tension controls the stiffness and lower values will simulate the behaviour of a membrane while higher values will simulate a thin metal plate. An explanation to increasing tension with increasing density is that too sparse data can cause the algorithm to overshoot if the stiffness is too high. Therefore the tension is lowered to simulate a membrane and reduce overshooting. When the density increases the risk of overshooting decreases and a more stiff spline is suitable.

5.3 Error and Accuracy

It should be mentioned that the tables of error for the different data sets and algorithm uses one training set and one validation set. This can also be achieved with real measured data. The plots with interpolation error against position however uses all known values in the whole grid, which would not be possible with irregular scattered data which one would have in a real situation.

For the interpolation error, Kriging performed the worst for all densities. The reason for this is unknown but one explanation could be that the data was too sparse to create a good variogram. It is suggested [10] that at least 100 different bins should be available to create a good variogram. Looking at the variograms created, too few bins was created for S_5 and S_{10} due to too sparse data.

The Regularized Spline with Tension was proven to be best for all densities except for the lowest one where Inverse Distance Weightning performed best. A reason for the RST to performe worse at low densities is that too little information is available to create a good spline of the surface. Instead the more simple IDW performs good since it simply takes an average between the closest points which turns out to be the best guess.

When it comes to the error propagation a comparison was made between the analytical solution and the Monte Carlo simulation and a significant difference could be noticed. This difference increased with increasing uncertainty in the input and in real data the uncertainty would be even higher. The conclusion is that the linearisation is not good enough since Monte Carlo was taken over a large number of simulations and can be considered correct. Due to this the decision was made not to make a numerical analysis on the real data or the other methods.

When performing Monte Carlo simulations on the other methods and with the real data sets only 100 simulations was performed. It was however determined that 100 simulations was more than enough by trying different number of simulations. Due to the larger number of data points, good indications and convergence of the error propagation could be achieved after only 50 simulations. In Figure 4.10 it can be seen that there is a peak in the error for RST at 10%. The reason for this is probably that the parameters was not optimised for low error propagation, for example the smoothing parameter for this set was 0.03 while it was 0.01 for the other sets. When looking at Figure 4.11(f) one can see large areas where that algorithm tends to overshoot or undershoot the correct values. One can also see that the data is divided into a grid which also is an effect of bad parameters. A better choice of parameters could prevent this by creating a smooth transition between the cells and reduce the overall error. Overall, it could be noted that the density had a very small impact on the error propagation.

It could also be noted that the plots of the errors against position looks rather similar for OK and IDW, the error is spread out. The reason is that both of these use a sort of weighted average over the nearest neighbours. In the RST on the other hand, the errors seem to be much more clustered. The reason could be these these error clusters arises where the data points and errors in them causes the spline to overshoot or undershoot the real values.

5.4 Runtime

In Figure 4.13 it can be seen that all algorithms seems to scale linearly with the number of data points. However, RST increase much slower and is therefore the fastest algorithm for higher number of data points. The main reason for this is that in the RST a grid was implemented and the data divided which make the algorithm scale better. There are however still optimizations to be done to all algorithms. For example a grid can be used in both OK and IDW as well by only looking at the local area when searching for the closest neighbours. To improve runtime even further for high densities close data points could be merged to reduce the number of samples. Merging data points also have other advantages, for example reducing the uncertainty in the specified location.

6 Conclusion

The three methods Inverse Distance Weighting, Ordinary Kriging and Regularized Spline With Tension was implemented and compared. It could be shown that for very sparse data the most simple method, IDW, performed best while for more dense data the more complex method RST outperformed both IDW and OK in terms of interpolation error.

For the error propagation IDW performed best in most cases. However, the parameters was not optimized for this type of error which can give a biased result. By optimizing the parameters for low error propagation a better result can be achieved.

It could also be noted that in terms of the runtime, the RST scales much better (it runs faster) than the two other methods. This was expected from the theory since splitting the data into different regions was implemented in the RST.

From the theory and literature studies, RST is the most suitable for inhomogeneous and cluster data [13]. When real data will be collected, this will most likely be the case and clusters will appear near harbours while more sparse data will be available at more hard to reach places.

Furthermore, water level data could be collected from SMHI and a system for compensation for changing water level was developed. This was however only tested on synthetic data to verify the algorithm.

6.1 Future Research

The RST algorithm still has a lot of parameters to be optimized. For example k_{min} and k_{max} remain to be examined. And as mentioned the variogram indicates different properties in different directions. Therefore horizontal scaling should also be implemented and optimized. Along with horizontal scaling, rotation of the axis also needs to be implemented to find the right directions of the different properties. To optimize over more parameters the optimization has to be made more effective and intelligent. As suggested above, properties from the variogram as well as the type of ocean floor could be used to find good initial guesses for the parameters.

There are still much left to do in optimizing the speed of the algorithm. More clever algorithms for finding the nearest neighbours could be implemented by dividing the data into a grid with suitable size. The speed can also be optimized by merging data points close by which also can reduce the uncertainty.

Future research also include testing the algorithms on real data, where real depth is represented and correct values for uncertainties can be used. This also includes comparing the methods on different types of ocean floors and areas. The hypothesis is that RST will perform best since this is most suitable for irregular sampled data.

Bibliography

[1] "Elevation data from nasa." http://dds.cr.usgs.gov/srtm/version2_1/SRTM1/, mar 2014.

[2] "Accurate bathymetry is the foundation for much of ocean science and policy." http://oceanservice.noaa.gov/facts/bathyuses.html, may 2014.

[3] S. Tanaka, "Aktuella sjökort viktigt." *Göteborgs-Posten*, 25 apr 2014.

[4] C. Nordling and J. Österman, *Physics Handbook*. Studentlitteratur AB, eighth ed., 2006.

[5] B. D. Kifana and M. Abdurohman, "Great circle distance methode for improving operational control system based on gps tracking system," *International Journal on Computer Science and Engineering*, vol. 4, pp. 647–662, 04 2012.

[6] D. Hendricks, "Maps in environmental monitoring," in *Environmental Monitoring and Characterization* (J. F. Artiola, I. L. Pepper, and M. L. Brusseau, eds.), pp. 69 – 84, Burlington: Academic Press, 2004.

[7] R. Macarthur, "Geographic information systems and their use for environmental monitoring," in *Environmental Monitoring and Characterization* (J. F. Artiola, I. L. Pepper, and M. L. Brusseau, eds.), pp. 85 – 100, Burlington: Academic Press, 2004.

[8] S. Erdogan, "A comparision of interpolation methods for producing digital elevation models at the field scale," *Earth Surface Processes and Landforms*, vol. 34, no. 3, pp. 366–376, 2009.

[9] G. Y. Lu and D. W. Wong, "An adaptive inverse-distance weighting spatialinterpolation technique," *Computers & Geosciences*, vol. 34, no. 9, pp. 1044 – 1055, 2008.

[10] M. Oliver, "The variogram and kriging," in *Handbook of Applied Spatial Analysis* (M. M. Fischer and A. Getis, eds.), pp. 319–352, Springer Berlin Heidelberg, 2010.

[11] E. Gringarten and C. V. Deutsch, "Teacher's aide variogram interpretation and modeling," *Mathematical Geology*, vol. 33, pp. 507–534, may 2001.

[12] A. Talmi and G. Gilat, "Method for smooth approximation of data," *Journal of Computational Physics*, vol. 23, no. 2, pp. 93 – 123, 1977.

[13] H. Mitášová and L. Mitáš, "Interpolation by regularized spline with tension: I. theory and implementation," *Mathematical Geology*, vol. 25, no. 6, pp. 641–655, 1993.

[14] J. Hofierka, J. Parajka, H. Mitasova, and L. Mitas, "Multivariate interpolation of precipitation using regularized spline with tension," *Transactions in GIS*, vol. 6, no. 2, pp. 135–150, 2002.

[15] M. Wahde, *Biologically inspired optimization methods: an introduction*. Southampton: WIT Press, 2008.

[16] J. Lei-yin and L. Hong-zhuan, "The application of improved genetic algorithm in fitting the spatial variogram," in *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, vol. 2, pp. 1031–1036, Dec 2011.

[17] "Höjdsystem och havsvattenstånd." `http://www.smhi.se/kunskapsbanken/oceanografi/hojdsystem-och-havsvattenstand-1.13582`, jan 2014.

[18] "Sea level." `http://www.smhi.se/klimatdata/Oppna-data/Oceanografiska-data/oceanografiska-observationer-1.30449`, jan 2014.

[19] "Stationslista havsvattenstånd." `http://www.smhi.se/kunskapsbanken/oceanografi/matstationer-for-havsvattenstand-1.13981`, jan 2014.

[20] G. B. M. Heuvelink, "Propagation of error in spatial modelling with gis," in *Geographic Information Systems – Principles, Techniques, Management, and Applications* (L. P, G. MF, M. DJ, and R. DW, eds.), pp. 207 – 217, Wiley, 1999.

A | Kriging Derivation

Derivation of (2.23):

$$\text{var}[\hat{Z}(x_0)] = E[\{\hat{Z}(x_0) - Z(x_0)\}^2] = 2\sum_{i=1}^{n}\lambda_i\gamma(x_i, x_0) - \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\gamma(x_i, x_j) \tag{A.1}$$

The variance can be written as

$$\text{var}[\hat{Z}(x_0)] = E[\{\hat{Z}(x_0) - Z(x_0)\}^2] + \Big(E[\hat{Z}(x_0) - Z(x_0)]\Big)^2 = E[\{\hat{Z}(x_0) - Z(x_0)\}^2] + 0 \tag{A.2}$$

since it is assumed that $E[\hat{Z}(x_0) - Z(x_0)] = 0$. Next is to show that

$$E[\{\hat{Z}(x_0) - Z(x_0)\}^2] = 2\sum_{i=1}^{n}\lambda_i\gamma(x_i, x_0) - \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\gamma(x_i, x_j) \tag{A.3}$$

To make the notation simpler $Z(x_i)$ is denoted as Z_i which gives

$$E[(\hat{Z}_0 - Z_0)^2] = E\left[\left(\sum_{i=1}^{n}\lambda_i Z_i - Z_0\right)^2\right] = \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j E[(Z - Z_i)(Z - Z_j)] \tag{A.4}$$

By rewriting $E[(Z - Z_i)(Z - Z_j)]$ according to

$$\begin{gathered}(E[Z_i - Z_j])^2 = (E[(Z_i - Z) + (Z - Z_j)])^2 = \\ = (E[Z_i - Z])^2 + 2E[(Z_i - Z)(Z - Z_j)] + (E[Z - Z_j])^2\end{gathered}$$

the following can be derived

$$E[(Z - Z_i)(Z - Z_j)] = \frac{1}{2}\left[(-E[Z_i - Z_j])^2 + (E[Z_i - Z])^2 + (E[Z - Z_j])^2\right] \tag{A.5}$$

This is used to rewrite (A.4)

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j E[(Z - Z_i)(Z - Z_j)] = 2\sum_{i=1}^{n}\lambda_i\frac{1}{2}(E[Z - Z_i])^2 - \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\frac{1}{2}(E[Z_i - Z_j])^2 \tag{A.6}$$

by using the fact that $\sum_{j=1}^{n} \lambda_j = 1$. The expected value is replaced by the variogram according to $2\gamma(x_i, x_j) = (E[Z_i, Z_j])^2$

$$E[(\hat{Z}_0 - Z_0)^2] = 2\sum_{i=1}^{n} \lambda_i \gamma(x_0, x_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j \gamma(x_i, x_j) \tag{A.7}$$

and (2.23) is shown.