Investigating "Optimal" Kriging Variance Estimation: Analytic and Bootstrap Estimators

Mzabalazo Z. Ngwenya

A dissertation submitted in fulfilment of the requirements for the degree of Master of Science in Mathematical Statistics

Supervisors: Associate Professor Christien Thiart & Professor Linda M. Haines

Department of Statistical Sciences
University of Cape Town

February 2011

Declaration

I hereby grant the University of Cape Town permission to copy and disseminate this work, or any part thereof, for the purposes of study and research.

I hereby declare

1. This dissertation is my own work. It has not been submitted before for any degree or examination to any other University.
2. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.
3. Each significant contribution to, and quotation in, this dissertation from the work of other people has been cited and referenced.

Signature :_____

Date :_____

Acknowledgements

I would firstly like to thank the National Research Foundation (NRF) for the financial support they afforded me over two years through the Prestigious/Equity Masters scholarship. The Statistical Sciences department and in particular my supervisors Associate Professor Christien Thiart and Professor Linda Haines for the time, effort and other resources they put towards the undertaking and completion of this research. My family, namely my mother and sister, for the emotional and financial support plus encouragement they have provided me through my years of study. Londiwe for the emotional support and encouragement she has shown me. And Yahweh for blessing me with wonderful, understanding and supportive people around me and for girding me with the strength so I can complete this work.

Contents

Abstract

Kriging is a widely used group of techniques for predicting unobserved responses at specified locations using a set of observations obtained from known locations. Kriging predictors are best linear unbiased predictors (BLUPs) and the precision of predictions obtained from them are assessed by the mean squared prediction error (MSPE), commonly termed the kriging variance. Expressions for the BLUPs are readily derived assuming that the covariance function and its parameters are known. However, in practice the covariance function and its parameters are usually not known and have to be inferred and estimated from the data respectively. This thesis is concerned with examining suitable estimators for the MSPE of the empirical best linear unbiased predictor (EBLUP), which is the BLUP with estimates of the covariance parameters, given here by maximum likelihood (ML) and restricted maximum likelihood (REML) estimators, "plugged-in" in place of the unknown parameters. Five such estimators which are the empirical mean squared prediction error (EMSPE), the Kacker-Harville and Prasad-Rao estimators and two bootstrap estimators, the unconditional and conditional bootstrap estimators, are examined in detail. A simulation study with the aim of identifying the best performing estimator in terms of bias within the context of the study is described. In addition the feasibility of calculating and using the estimators in the spatial analysis of two real data sets, one a widely used example in the geostatistical literature and the other a South African data set comprising measurements on micronutrients at locations in the Gauteng area, is also presented. A suite of R programs for calculating the estimators of the MSPE of the EBLUP is provided.

List of Figures

List of Tables

Acronyms

BLUP	Best Linear Unbiased Predictor
DACE	Design and Analysis of Computer Experiments
EGLS	Estimated Generalized Least Squares
EBLUP	Empirical Best Linear Unbiased Predictor
EMSPE	Empirical Mean Squared Prediction Error
GLS	Generalized Least Squares
LS	Least Squares
MLE	Maximum Likelihood Estimator
MSE	Mean Squared Error
MSPE	Mean Squared Prediction Error
OLS	Ordinary Least Squares
REML	Restricted Maximum Likelihood Estimation
RMSPE	Root Mean Squared Prediction Error
WLS	Weighted Least Squares

Basic Notation

Symbols

$\delta(\cdot)$	error process
$\forall$	for all
$\in$	in
h_{ij}	distance between point $\mathbf{x}_i$ and $\mathbf{x}_j$
τ^2	nugget variance
$n(\mathbf{x}_0)$	number of observations in $\mathbf{x}_0$'s neighbourhood
D	observation domain
$\mathbf{x}_i$	observation location
σ_0^2	partial sill
$\mathbf{x}_0$	prediction location
$\hat{Z}(\mathbf{x}_0)$	predictor
$\mathbb{R}^d$	process domain
ϕ	range parameter
$C(0)$	stationary variance of $Z(\cdot)$
$Z(\cdot)$	stochastic process
$\subset$	subset
σ^2	variance

Functions

$\text{Corr}\{\cdot,\cdot\}$	correlation
$\text{Cov}\{\cdot,\cdot\}$	covariance
$C(\cdot)$	covariance function
$\text{E}[\cdot]$	expected value
$\mu(\cdot)$	mean
$\gamma(\cdot)$	semivariogram
$\text{Var}\{\cdot\}$	variance

Matrices and Sets

$\mathbf{V}(\boldsymbol{\theta})$	data to data covariance matrix; $n \times n$
$\mathbf{v}(\boldsymbol{\theta})$	data to prediction covariance vector; $n \times 1$
$\mathbf{R}(\phi)$	data to data correlation matrix; $n \times n$
$\boldsymbol{\delta}=[\delta_1,\ldots,\delta_n]^\top$	error process vector; $n \times 1$
$\boldsymbol{\lambda}=[\lambda_1,\ldots,\lambda_n]^\top$	kriging weights vector; $n \times 1$
$\boldsymbol{\varphi}=[\varphi_0,\ldots,\varphi_K]^\top$	Lagrange parameter vector; $(K+1) \times 1$
$\boldsymbol{\mu}=[\mu(\mathbf{x}_1),\ldots,\mu(\mathbf{x}_n)]^\top$	mean vector; $n \times 1$
$\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$	observation locations set
$\boldsymbol{\theta}=[\theta_1,\ldots,\theta_q]^\top$	parameter vector; $q \times 1$
$\{\mathbf{x}_{01},\ldots,\mathbf{x}_{0m}\}$	prediction (test) locations set
$\hat{\boldsymbol{\gamma}} = [\hat{\gamma}(h_y),\ldots,\hat{\gamma}(h_Y)]^\top$	semivariogram estimates at Y lags vector; $Y \times 1$
$\boldsymbol{\beta}=[\beta_0,\ldots,\beta_K]^\top$	unknown constants vector; $(K+1) \times 1$
$\mathbf{F}$	matrix of polynomial functions on $\mathbf{x}$; $n \times (K+1)$
$\mathbf{f}_0 = [f_0(\mathbf{x}),\ldots,f_K(\mathbf{x})]^\top$	regression functions vector on $\mathbf{x}_0$; $(K+1) \times 1$

Chapter 1

Introduction

1.1 Background

Kriging is a set of optimal spatial linear prediction methods that result in predictions with minimum mean squared prediction errors (Cressie 1993). These methods of interpolation are akin to the optimal prediction methods first developed by Wold (1938), Kolmogorov (1941) and Wiener (1949). Kriging arises from the work of the South African mining engineer Krige (1951) who collaborated with fellow South African, the statistician Sichel, to develop methods for determining the distribution of ore grades in the Witwatersrand based on a set of samples ore grades. The techniques and concepts developed by Krige were then later formalized and pooled into one body of knowledge by French mathematician Matheron (1963; 1973; 1975; 1989) who coined the term "kriging" in recognition of Krige's pioneering work.

Although the method has its roots firmly in mining, kriging has found wide application in various other earth science disciplines as an interpolation method such as in hydrology (Kitandis and and Vomoris, 1993), soil science (Goovaerts, 1999; Webster and Oliver, 2001), petroleum geology (Hohn, 1999) and agriculture (Cahn *et al*, 1994). It has also found uses outside of the earth sciences such as in public health (Carret and Vellaron, 1992) and increasingly in the arena of the design and analysis of computer experiments (DACE) (Sacks *et al*, 1989; Welch *et al*, 1992; Bernardo *et al*, 1992; Jones *et al*, 1998; Jones, 2001; Santer *et al*, 2003; Martin and Simpson, 2005; Fang *et al*, 2006; Kleijnen, 2009). In DACE, kriging models are used as response surface models and model the input/output behaviour of complex simulation models (den Hertog *et al*, 2006). In this setting they fall within a class of models which are termed metamodels (Kleijnen, 1987).

There are several reasons for the popularity of kriging as an interpolation method some of which are noted below:

- In the absence of measurement error, referred to as a nugget, it interpolates observations exactly.

- Kriging allows the user to interpolate over areas or volumes that are larger than the sample sites (Journel and Huijbregts, 1978; Burrough and McDonnell, 1998).
- It allows the interpolation of binary data.
- It can also make use of additional information such as covariate information which is cheaply obtainable at the sample and prediction locations.
- It assigns a prediction error to each predicted value.

The squared prediction error is termed the kriging variance and is a measure of a predictions accuracy. It can also be used to judge the overall quality of a fitted model. The kriging variance, which is in fact a mean squared prediction error (MSPE), besides just being used for gauging the quality of predictions or the quality of the kriging models has other applications. For instance it is used in the following settings:

- To build maps of uncertainty for interpolated surfaces in the earth sciences (Todini, 2001).
- To build confidence intervals for predicted ore grades in mining (Journel and Huijbregts, 1978).
- To design "optimal" environmental monitoring networks (Cressie *et al*, 1990; Zimmerman, 2006).
- To construct application-driven sequential designs for computer experiments (Sacks *et al*, 1989; Kleijnen *et al*, 2004).
- To select new input design points to find the global optimum of complex computer simulation models (Booker *et al*, 1999; Sasena *et al*, 2002).

1.2 Research Problem

The expressions for obtaining kriging predictions and the corresponding kriging variances contain unknown covariance parameters. In practice the covariance structure is usually modelled using one of various parametric covariance functions and the unknown covariance parameters, $\boldsymbol{\theta}$, are estimated by $\hat{\boldsymbol{\theta}}$ using the existing data. The estimate, $\hat{\boldsymbol{\theta}}$, is then plugged into the kriging predictor, producing the empirical best linear unbiased predictor (EBLUP), and the corresponding kriging variance is obtained similarly by "plugging-in" $\hat{\boldsymbol{\theta}}$ into the expression for MSPE resulting in an estimated kriging variance.

A major practical question is how to incorporate the "uncertainty" that arises from using estimated instead of known covariance parameters in the determination of the

mean squared prediction error (MSPE) of the EBLUP. By ignoring the fact that $\boldsymbol{\theta}$ is estimated, the resulting MSPE, which is termed the empirical MSPE (EMSPE), leads to the underestimation of the MSPE of the EBLUP (Zimmerman and Cressie, 1992; Cressie, 1993; Abt 1998; Abt 1999; Wang and Wall, 2003; den Hertog et al, 2006).

1.3 Aims of Research

The aims of this research are as follows.

- To explore EMSPE, alternative methods of estimating the MSPE of the EBLUP and possibly develop another method of estimating the MSPE of the EBLUP all with the final aim of finding the most "optimal" estimator.
- It is also the intention of the research to avail means by which practitioners can calculate the "optimal" estimator and the various other estimators that will be investigated by developing flexible and portable programs.
- Lastly, we intend to investigate, though somewhat briefly, the consequences of using the various estimators as the basis of design criteria in optimal sampling design.

Section 1.4 provides a summary of the specific research objectives of this study.

1.4 Research Objectives

1. To investigate the performance of the empirical mean squared prediction error (EMSPE) in estimating the mean squared prediction error (MSPE) of the empirical best linear unbiased predictor (EBLUP).
2. To investigate and develop alternative methods to the EMSPE for estimating the MSPE of the EBLUP and the evaluation of the performance of these alternative estimators.
3. To develop portable programs in R (R Development Core Team, 2010) that provide the user with the various estimators of the MSPE of the EBLUP.
4. To illustrate and test the flexibility of the alternative methods of estimating the MSPE of the EBLUP mentioned in 3 on real data sets.
5. To examine how sampling designs based on the different kriging variance estimators as the basis of design criteria in optimal sampling design differ from each other.

1.5 Outline of thesis

In Chapter 2, the theory underlying kriging models is given, basic kriging equations, predictors and variances, are carefully derived and a discussion of the kriging assumptions undertaken. The semivariogram and covariance, their role in kriging and the estimation of the parameters of these functions, namely via least squares and likelihood estimation methods, are then discussed. Lastly the research problem is introduced and reasons as to why the empirical mean squared prediction error (EMSPE) is a biased estimator of the MSPE of the empirical best linear unbiased predictor (EBLUP) are discussed. Existing alternative methods to the EMSPE for estimating the kriging variance when the covariance parameters are estimated are explored and a novel approach involving bootstrapping is briefly outlined.

Chapter 3 details the setup, execution and the results of a simulation study intended not only to investigate the bias of the EMSPE but also to identify the estimator with the smallest bias from two analytic estimators proposed by Zimmerman and Cressie (1992), which are the Kacker-Harville and Prasad-Rao estimators and two bootstrap estimators, namely the unconditional and conditional bootstrap estimators.

In Chapter 4 the estimators investigated in Chapter 3 are applied to two real world data sets. The first data set is the well known Meuse data set which consists of measurements of heavy metals from the soils in the flood plain of the River Meuse (see Burrough and McDonnell, (1998)). The second data set is a local, South African, data set on which kriging methods have not been applied before. This involves measurements of micronutrients in the Witwatersrand area. and is a subset of data used in de Villers *et al* (2010).

Chapter 5 summarizes the important findings and contributions of this research. The limitations of the research are also documented and directions for further research are briefly discussed.

Appendix A gives the R functions that were developed during the course of this research, namely functions for calculating the various estimators investigated here and functions for executing the simulation experiment reported in Chapter 3.

Appendix B gives R documentation for the functions that compute the estimators of Chapter 3. The documentation is important as, as mentioned earlier on, it is intended that the functions be used by other geostatistical practitioners in their own work.

Appendix C gives R syntax for conducting the analysis reported in Chapter 4. Also given are some of the Routput, namely the fitted models, reported in the chapter.

Chapter 2

Spatial Prediction and Kriging

A common problem in geosciences is to predict the value of a physical process at an unobserved location $\mathbf{x}_0$ using n observations $\mathbf{z} = [z(\mathbf{x}_1), \ldots, z(\mathbf{x}_n)]^\top$ obtained at n locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ within a study area D, see Figure 2.1. Conventionally the physical process is modelled as a spatial process

$$Z(\cdot) = \{Z(\mathbf{x}) : \mathbf{x} \in D \subset \mathbb{R}^d\}, \tag{2.1}$$

operating in d dimensions over the entire domain D (Schabenberger and Gotway, 2005). The practical problem is to predict $Z(\cdot)$ at $\mathbf{x}_0 \in D$ using $\mathbf{z} = [z(\mathbf{x}_1), \ldots, z(\mathbf{x}_n)]^\top$ under the assumption that $\mathbf{z}$ is a realization of the n random variables $\mathbf{Z} = [Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_n)]^\top$ which form one partial realization of $Z(\cdot)$. The dimensions over which the process operates is commonly $d = 2$ but may also be $d = 1$ or $d = 3$.

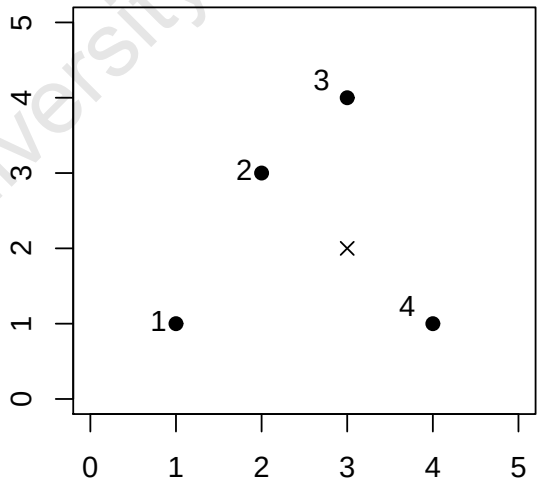


Figure 2.1: Spatial prediction scenario. An attribute is observed at four locations $(1, 1), (2, 3), (3, 4), (4, 1)$ (black dots). Prediction of the attribute at the unsampled location $\mathbf{x}_0 = (3, 2)$ (cross) is required.

A variety of interpolation methods can be used to predict a value at an unsampled location. The prediction of the unknown value $z(\mathbf{x}_0)$ is then a weighted average of

the neighbouring observations, for example:

$$\hat{z}(\mathbf{x}_0) = \sum_{i=1}^{n(\mathbf{x}_0)} \lambda_i z(\mathbf{x}_i), \tag{2.2}$$

where λ_i is the weight associated with the i^{th} observation $z(\mathbf{x}_i)$ and $n(\mathbf{x}_0)$ is the number of points in a search neighbourhood around the prediction location $\mathbf{x}_0$. The weighting is usually a function of the distance $h_{0i} = ||\mathbf{x}_0 - \mathbf{x}_i||$ between the observation location $\mathbf{x}_i$ and the point of interest $\mathbf{x}_0$, $i = 1, \ldots, n(\mathbf{x}_0)$. The distance measure h_{ij} between any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ may be defined in various ways. Let $\mathbf{x}_i = (x_{1i}, x_{2i})$ and $\mathbf{x}_j = (x_{1j}, x_{2j})$ denote the coordinates of two points in Euclidean space. Then h_{ij}, the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, is given by

$$h_{ij} = \sqrt{(x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2}.$$

Similarly if $\mathbf{x}_i = (u_i, \upsilon_i)$ and $\mathbf{x}_j = (u_j, \upsilon_j)$ are points on the earth's surface, where u represents latitude and υ longitude, then the distance h_{ij} can be defined as the great arc distance

$$h_{ij} = 6378 \cdot \arccos[\sin u_i \sin u_j + \cos u_i \cos u_j \cos(\upsilon_i - \upsilon_j)].$$

The constant 6378 is the radius of the spherical earth in kilometres (Diggle and Riberio, 2007). In this research we use Euclidean distance exclusively.

Some common prediction and interpolation methods include:

- Inverse distance weighting (IDW)
  This deterministic method is the least sophisticated of all the interpolation methods. The weight λ_i in expression (2.2) is a function of the distance h_{0i}, usually $1/h_{0i}$, such that $\lambda_i = (1/h_{0i})^p$, $i = 1, \ldots, n$ where p is some integer (Davis, 2002).

- Interpolation splines
  Interpolation splines are a set of deterministic functions that depend on distance only. These "piecewise functions" act like flexible surfaces which pass through observation points when interpolating a surface (see Wahba, 1990). These splines are rarely used in geosciences as they do not pass through observation points.

- Local and global polynomials
  These deterministic methods are based on mathematical functions that model the mean function over D. Global polynomials fit one function over the whole domain. Local polynomial interpolation fits a polynomial in a moving window (specific neighbourhood) over the domain D (Davis, 2002).

- Kriging
  This is a geostatistical method based on the concept that spatial variables

can be considered as partly deterministic and partly stochastic (see Journel and Huijbregts, 1978, §II.A). In kriging the weights in expression (2.2) are determined by minimizing the prediction error $\mathrm{E}\{[z(\mathbf{x}_0) - \hat{z}(\mathbf{x}_0)]^2\}$ (Krige, 1951; Matheron,1963; Cressie, 1990).

Note that for deterministic methods, the only way to judge the accuracy of predictions is to estimate the Root Mean Square Prediction Error (RMSPE) using cross-validation. The RMSPE is defined as

$$\mathrm{RMSPE} = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\{z(\mathbf{x}_i) - \hat{z}_{(-i)}(\mathbf{x}_i)\}^2},$$

where $z(\mathbf{x}_i)$ is the observed value at $\mathbf{x}_i$ and $\hat{z}_{(-i)}(\mathbf{x}_i)$ is the predicted value obtained by fitting the data over all locations other than $\mathbf{x}_i$. The RMSPE measures the overall accuracy of the predictions. For kriging, the accuracy of predictions can not only be judged by use of the RMSPE but may also be gauged using the Mean Squared Prediction Error (MSPE)

$$\mathrm{E}\{[\hat{z}(\mathbf{x}_0) - z(\mathbf{x}_0)]^2\} = \mathrm{Var}\{z(\mathbf{x}_0)\} + \mathrm{Var}\{\hat{z}(\mathbf{x}_0)\} - 2\ \mathrm{Cov}\{\hat{z}(\mathbf{x}_0), z(\mathbf{x}_0)\}, \qquad (2.3)$$

which is the prediction error associated with each prediction $\hat{z}(\mathbf{x}_0)$.

2.1 Kriging

It is the norm to denote random variables by capital letters $\mathbf{Z} = [Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_n)]^\top$ and observations by small letters $\mathbf{z} = [z(\mathbf{x}_1), \ldots, z(\mathbf{x}_n)]^\top$. However, from hence forth we will use capital letters to denote both random variables and observations. It should however be clear as to which we are referring to from the context of each discussion.

Kriging models are based on the assumption that the random variables which form the spatial process are generated by the model

$$\mathbf{Z} = \boldsymbol{\mu} + \boldsymbol{\eta} + \boldsymbol{\epsilon}, \qquad (2.4)$$

where the components of the realization vector $\mathbf{Z}$ reflect the various scales of variation of the physical process (Cressie, 1993). The decomposition of spatial processes is not unique and depends on the nature of the problem and statistical analysis to be carried out. In decomposition (2.4), $\boldsymbol{\mu} = \mathrm{E}\{\mathbf{Z}\}$ is the large scale deterministic variation and the other two components, which are stochastic processes, reflect spatial variation.

The stochastic process $\boldsymbol{\eta}$ is a zero mean process:

$$\boldsymbol{\eta} \sim (\mathbf{0}, \sigma_0^2\mathbf{R}(\phi)),$$

where $\mathbf{R}(\phi)$ is the spatial correlation matrix with ij^{th} element

$$\mathrm{R}_{ij}(\phi) = \mathrm{Corr}\{Z(\mathbf{x}_i), Z(\mathbf{x}_j)\}$$

depending on some function of the vector $(\mathbf{x}_i - \mathbf{x}_j)$ and on an unknown distance parameter ϕ. The distance parameter ϕ is commonly known as the range and the term σ_0^2 is referred to as the partial sill. The stochastic process $\boldsymbol{\epsilon}$, the nugget, which is random noise is also a zero mean process:

$$\boldsymbol{\epsilon} \sim (\mathbf{0}, \tau^2 \mathbf{I}_n),$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix and τ^2 is the nugget variance. The nugget can be attributed to two sources. These are spatial variation on a scale smaller than the available minimum distance between two observation points and measurement error (Cressie, 1993; Riberio and Diggle, 2007). Commonly the stochastic components of $\mathbf{Z}$ are grouped together to form the component $\boldsymbol{\delta} = \boldsymbol{\eta} + \boldsymbol{\epsilon}$ with zero mean and variance-covariance matrix $\mathbf{V}(\boldsymbol{\theta}) = \sigma_0^2 \mathbf{R}(\phi) + \tau^2 \mathbf{I}_n$ where $\boldsymbol{\theta} = (\tau^2, \sigma_0^2, \phi)$.

The deterministic trend $\boldsymbol{\mu}$ is commonly modelled as a function of the observation locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ by polynomial functions $f_k(\cdot)$ and unknown parameters β_k for $k = 1, \ldots, K$ such that $\boldsymbol{\mu} = \mathbf{F}\boldsymbol{\beta}$ where

$$\mathbf{F} = \begin{bmatrix} 1 & f_1(\mathbf{x}_1) & \ldots & f_K(\mathbf{x}_1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & f_1(\mathbf{x}_n) & \ldots & f_K(\mathbf{x}_n) \end{bmatrix}_{n \times (K+1)}$$

with $f_0(x) \equiv 1$ and $\boldsymbol{\beta} = [\beta_0, \ldots, \beta_K]^\top$ is a $(K+1) \times 1$ vector. The structure $\mathbf{F}\boldsymbol{\beta}$ can be modified to give a variety of models. The general kriging model, which is the universal kriging model, is given by

$$\mathbf{Z} = \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \sim (\mathbf{0}, \mathbf{V}(\boldsymbol{\theta})). \tag{2.5}$$

In this model, the mean is unknown and is assumed to vary throughout the observation domain D. Ordinary kriging is another form of kriging. In this model the mean is assumed to be constant over D, that is stationary, but is unknown, that is $f_0(\mathbf{x}) \equiv 1$, $\mathbf{F} = \mathbf{1}$ is a $n \times 1$ vector of ones and β is a scalar. The ordinary kriging model is then given as

$$\mathbf{Z} = \mathbf{1}\beta + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \sim (\mathbf{0}, \mathbf{V}(\boldsymbol{\theta})), \tag{2.6}$$

where $\mathbf{1}\beta = \boldsymbol{\mu}$. The broader concept of stationarity is discussed in detail in Section 2.2.1. Though it is uncommon in practice, the mean may not only be assumed constant but it may also be taken as known. This produces a kriging model similar to that specified in (2.6) but in which the mean is not estimated. This kriging model is termed the simple kriging model. The mean assumptions and the mean models for the universal, ordinary and simple kriging models are summarized in Table 2.1.

Table 2.1: Universal, ordinary and simple kriging model assumptions and models of the mean.

Type	**Assumption**	**Mean model**
Universal kriging	non-stationary, unknown mean	$\mathbf{F}\boldsymbol{\beta}$
Ordinary kriging	constant, unknown mean	$\mathbf{1}\beta$
Simple kriging	constant, known mean β	$\mathbf{1}\beta$

2.1.1 Optimal prediction

Given the n observations $Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_n)$ at the n locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and the unknown value $Z(\mathbf{x}_0)$ at the location $\mathbf{x}_0$, the aim of kriging is to find the best predictor $\hat{Z}(\mathbf{x}_0)$ of $Z(\mathbf{x}_0)$. This can be done in the absence of any distributional assumptions on $\mathbf{Z}$. The best predictor is defined to be the predictor that is:

1. Linear in $\mathbf{Z} = [Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_n)]^\top$:

$$\hat{Z}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \mathbf{Z} \tag{2.7}$$

   where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^\top$ is an $n \times 1$ vector of weights.

2. Unbiased:

$$\mathrm{E}\{\hat{Z}(\mathbf{x}_0)\} = \mathrm{E}\{Z(\mathbf{x}_0)\} = \mathbf{f}_0^\top \boldsymbol{\beta}, \tag{2.8}$$

   where $\mathbf{f}_0 = [f_0(\mathbf{x}_0), \ldots, f_K(\mathbf{x}_0)]^\top$ is a $(K+1) \times 1$ vector of regressors.

3. Minimizes the expected MSPE:

$$\mathrm{E}\{[\hat{Z}(\mathbf{x}_0) - Z(\mathbf{x}_0)]^2\}.$$

Thus we seek to find a linear unbiased minimum mean-squared predictor. Formally the statistical criterion for obtaining such a predictor involves the minimization of the mean squared prediction error (MSPE)

$$\min_{\lambda} \ \mathrm{E}\{[\boldsymbol{\lambda}^\top \mathbf{Z} - Z(\mathbf{x}_0)]^2\},$$

with respect to the weights $\boldsymbol{\lambda}$=$[\lambda_1, \ldots, \lambda_n]^\top$ under the constraint of unbiasedness (2.8). The resulting predictor $\hat{Z}(\mathbf{x}_0)$ is termed the Best Linear Unbiased Predictor (BLUP) (Cressie, 1993).

2.1.2 Derivation of predictors and kriging variances

Given the universal kriging model (2.5)

$$\mathbf{Z} = \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \sim (\mathbf{0}, \mathbf{V}(\boldsymbol{\theta}))$$

a predictor $\hat{Z}(\mathbf{x}_0)$ of $Z(\mathbf{x}_0)$ at $\mathbf{x}_0$ is required, which is the BLUP. Under model (2.5) the expected value of $\hat{Z}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \mathbf{Z}$ is

$$\begin{aligned} \mathrm{E}\{\hat{Z}(\mathbf{x}_0)\} &= \mathrm{E}\{\boldsymbol{\lambda}^\top \mathbf{Z}\} \\ &= \mathrm{E}\{\boldsymbol{\lambda}^\top \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\lambda}^\top \boldsymbol{\delta}\} \\ &= \boldsymbol{\lambda}^\top \mathbf{F}\boldsymbol{\beta} \equiv \mathbf{f}_0^\top \boldsymbol{\beta}, \end{aligned}$$

and thus $\mathbf{F}^\top \boldsymbol{\lambda} \equiv \mathbf{f}_0$.

As noted in Section 2.1.1, to obtain the BLUP the prediction variance

$$\begin{aligned} \mathrm{E}\{[\boldsymbol{\lambda}^\top \mathbf{Z} - Z(\mathbf{x}_0)]^2\} &= \mathrm{Var}\{Z(\mathbf{x}_0)\} + \mathrm{Var}\{\boldsymbol{\lambda}^\top \mathbf{Z}\} - 2\mathrm{Cov}\{Z(\mathbf{x}_0), \boldsymbol{\lambda}^\top \mathbf{Z}\} \\ &= \sigma^2 + \boldsymbol{\lambda}^\top \mathbf{V}(\boldsymbol{\theta})\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^\top \mathbf{v}(\boldsymbol{\theta}), \end{aligned} \tag{2.9}$$

where $\sigma^2 = \tau^2 + \sigma_0^2 = \mathrm{Var}\{Z(\mathbf{x}_0)\}$ and $\mathbf{v}(\boldsymbol{\theta})$ is the observation location $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ to prediction location $\mathbf{x}_0$ covariance vector, must be minimized. To simplify the notation we omit the symbol $\boldsymbol{\theta}$ from the terms $\mathbf{V}(\boldsymbol{\theta})$ and $\mathbf{v}(\boldsymbol{\theta})$ and use $\mathbf{V}$ and $\mathbf{v}$ respectively.

To obtain a predictor that minimizes the prediction variance, expression (2.9) must be minimized with respect to the weights $\boldsymbol{\lambda}$. However, the constraint of unbiasedness $\mathbf{F}^\top \boldsymbol{\lambda} \equiv \mathbf{f}_0$ must also be imposed. Thus Lagrange parameters are introduced to allow for unconstrained optimization. The resulting Lagrangian function is

$$\ell = \sigma^2 + \boldsymbol{\lambda}^\top \mathbf{V}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^\top \mathbf{v} + 2\boldsymbol{\varphi}^\top (\mathbf{F}^\top \boldsymbol{\lambda} - \mathbf{f}_0) \tag{2.10}$$

where $\boldsymbol{\varphi} = [\varphi_0, \ldots, \varphi_K]^\top$ is a $(K+1) \times 1$ vector of the Lagrange parameters, and the term $(\mathbf{F}^\top \boldsymbol{\lambda} - \mathbf{f}_0)$ incorporates the unbiasedness condition. Differentiating (2.10) with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\varphi}$ and setting the results to zero yields

$$\begin{aligned} \frac{\partial \ell}{\partial \boldsymbol{\lambda}} &= 2\mathbf{V}\boldsymbol{\lambda} - 2\mathbf{v} + 2\mathbf{F}\boldsymbol{\varphi} = \mathbf{0} \\ \frac{\partial \ell}{\partial \boldsymbol{\varphi}} &= 2(\mathbf{F}^\top \boldsymbol{\lambda} - \mathbf{f}_0) = \mathbf{0}. \end{aligned}$$

Rearranging the terms and dividing by 2 yields what are termed the kriging system of equations

$$\begin{bmatrix} \mathbf{V} & \mathbf{F} \\ \mathbf{F}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\varphi} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{f}_0 \end{bmatrix}.$$

Since

$$\begin{bmatrix} \mathbf{V} & \mathbf{F} \\ \mathbf{F}^\top & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{V^{-1}} - \mathbf{V^{-1}F\Delta^{-1}F^\top V^{-1}} & \mathbf{V^{-1}F\Delta^{-1}} \\ \mathbf{\Delta^{-1}F^\top V^{-1}} & \mathbf{-\Delta^{-1}} \end{bmatrix}$$

where $\boldsymbol{\Delta} = \mathbf{F}^\top \mathbf{V}^{-1}\mathbf{F}$, solving the kriging system of equations for $\boldsymbol{\varphi}$ and $\boldsymbol{\lambda}$ simultaneously results in the expressions

$$\boldsymbol{\varphi} = \boldsymbol{\Delta}^{-1}(\mathbf{F}^\top \mathbf{V}^{-1}\mathbf{v} - \mathbf{f}_0)$$

and

$$\begin{aligned} \boldsymbol{\lambda} &= \mathbf{V}^{-1}\mathbf{v} - \mathbf{V}^{-1}\mathbf{F}\boldsymbol{\Delta}^{-1}\mathbf{F}^\top\mathbf{V}^{-1}\mathbf{v} + \mathbf{V}^{-1}\mathbf{F}\boldsymbol{\Delta}^{-1}\mathbf{f}_0 \\ &= \mathbf{V}^{-1}\mathbf{v} - \mathbf{V}^{-1}\mathbf{F}\boldsymbol{\Delta}^{-1}(\mathbf{F}^\top\mathbf{V}^{-1}\mathbf{v} - \mathbf{f}_0) \\ &= \mathbf{V}^{-1}(\mathbf{v} - \mathbf{F}\boldsymbol{\varphi}). \end{aligned} \tag{2.11}$$

Substituting (2.11) into expression (2.7) gives the BLUP

$$\begin{aligned} \hat{Z}(\mathbf{x}_0) &= \boldsymbol{\lambda}^\top\mathbf{Z} \\ &= \mathbf{f}_0^\top\boldsymbol{\Delta}^{-1}\mathbf{F}^\top\mathbf{V}^{-1}\mathbf{Z} + \mathbf{v}^\top\mathbf{V}^{-1}\mathbf{Z} - \mathbf{v}^\top\mathbf{V}^{-1}\mathbf{F}\boldsymbol{\Delta}^{-1}\mathbf{F}^\top\mathbf{V}^{-1}\mathbf{Z} \\ &= \mathbf{f}_0^\top\hat{\boldsymbol{\beta}} + \mathbf{v}^\top\mathbf{V}^{-1}(\mathbf{Z} - \mathbf{F}\hat{\boldsymbol{\beta}}) \end{aligned} \tag{2.12}$$

where

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}^\top\mathbf{V}^{-1}\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{V}^{-1}\mathbf{Z}$$

is the Generalized Least Squares (GLS) estimate of $\boldsymbol{\beta}$. If $\mathbf{V}$ is known $\hat{\boldsymbol{\beta}}$ is the Maximum Likelihood Estimator (MLE). The variance of the predictor, the kriging variance, is obtained by substituting the value of the optimal weights given in (2.11) into the expression for the prediction variance (2.9). The kriging variance of $\hat{Z}(\mathbf{x}_0)$ is then given as

$$\sigma^2(\mathbf{x}_0) = \sigma^2 - \mathbf{v}^\top\mathbf{V}^{-1}\mathbf{v} + (\mathbf{f}_0^\top - \mathbf{v}^\top\mathbf{V}^{-1}\mathbf{F})(\mathbf{F}^\top\mathbf{V}^{-1}\mathbf{F})^{-1}(\mathbf{f}_0 - \mathbf{F}^\top\mathbf{V}^{-1}\mathbf{v}). \tag{2.13}$$

2.1.2.1 Ordinary kriging

For ordinary kriging $\mathbf{F} = \mathbf{1}$ with $\sum_{i=1}^{n}\lambda_i = 1$. Thus the ordinary kriging predictor $\hat{Z}(\mathbf{x}_0)$ is given as

$$\hat{Z}(\mathbf{x}_0) = \hat{\beta} + \mathbf{v}^\top\mathbf{V}^{-1}(\mathbf{Z} - \mathbf{1}\hat{\beta}). \tag{2.14}$$

where

$$\hat{\beta} = (\mathbf{1}^\top\mathbf{V}^{-1}\mathbf{1})^{-1}\mathbf{1}^\top\mathbf{V}^{-1}\mathbf{Z}.$$

The ordinary kriging variance $\sigma^2_{OK}(\mathbf{x}_0)$ is then given as

$$\sigma^2(\mathbf{x}_0) = \sigma^2 - \mathbf{v}^\top\mathbf{V}^{-1}\mathbf{v} + (1 - \mathbf{v}^\top\mathbf{V}^{-1}\mathbf{1})(\mathbf{1}^\top\mathbf{V}^{-1}\mathbf{1})^{-1}(1 - \mathbf{1}^\top\mathbf{V}^{-1}\mathbf{v}). \tag{2.15}$$

2.1.2.2 Simple kriging

As noted before, in simple kriging the mean is assumed to be a known constant. Because as in ordinary kriging the mean is assumed constant, ordinary and simple kriging are similar with the only difference being that there is no unbiasedness constraint for simple kriging so $\boldsymbol{\lambda} = \mathbf{V}^{-1}\mathbf{v}$. The simple kriging predictor is thus

$$\hat{Z}(\mathbf{x}_0) = \beta + \mathbf{v}^\top \mathbf{V}^{-1}(\mathbf{Z} - \mathbf{1}\beta), \tag{2.16}$$

with the variance estimator

$$\sigma^2(\mathbf{x}_0) = \sigma^2 - \mathbf{v}^\top \mathbf{V}^{-1}\mathbf{v}. \tag{2.17}$$

2.2 Kriging Assumptions

2.2.1 Stationarity

One of the assumptions in kriging is that of stationarity of the spatial process $Z(\cdot)$. Various levels of stationarity can be assumed.

2.2.1.1 Strict stationarity:

The process $Z(\cdot)$ is said to be strictly stationary if the multivariate cumulative distribution function (cdf) is invariant under translation of the coordinates (Journel and Huijbregts, 1978; Goovaerts, 1997). That is, the two n-component random vectors $\{Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_n)\}$ and $\{Z(\mathbf{x}_1+\boldsymbol{h}), \ldots, Z(\mathbf{x}_n+\boldsymbol{h})\}$ have the same n variable distribution law whatever the translation vector $\boldsymbol{h}$ and the value of n. This definition can be formally expressed as

$$Pr[Z(\mathbf{x}_1) \leq z_1, \ldots, Z(\mathbf{x}_n) \leq z_n] = Pr[Z(\mathbf{x}_1 + \boldsymbol{h}) \leq z_1, \ldots, Z(\mathbf{x}_n + \boldsymbol{h}) \leq z_n].$$

This implies that a strictly stationary spatial process "repeats itself" throughout the domain in which it operates (Schabenberger and Gotway, 2005, p43). Thus, in particular, the means and variances of $Z(\cdot)$ are constant across the domain.

2.2.1.2 Second order stationarity:

The process $Z(\cdot)$ is second order stationary if the following two conditions hold.

- **Condition 1:** *The expectation* $\mathrm{E}\{Z(\mathbf{x})\}$ *exists and does not depend on the location* $\mathbf{x}$; *thus*,

$$\mathrm{E}\{Z(\mathbf{x})\} = \mathrm{E}\{Z(\mathbf{x} + \boldsymbol{h})\} = \mu \qquad \forall\, \mathbf{x} \in D. \tag{2.18}$$

- **Condition 2:** *For each pair of random variables* $\{Z(\mathbf{x}), Z(\mathbf{x}+\boldsymbol{h})\}$ *the covariance exists and depends only on the separation vector* $\boldsymbol{h}$*, that is*

$$\begin{aligned}\text{Cov}\{Z(\mathbf{x}), Z(\mathbf{x}+\boldsymbol{h})\} &= \text{E}\{[Z(\mathbf{x})-\mu][Z(\mathbf{x}+\mathbf{h})-\mu]\} \\ &= \text{E}\{Z(\mathbf{x}+\boldsymbol{h})\cdot Z(\mathbf{x})\}-\mu^2 \\ &= C(\boldsymbol{h}), \quad \forall\ \mathbf{x},\ \mathbf{x}+\boldsymbol{h}\in D. \end{aligned} \tag{2.19}$$

The function $C(\boldsymbol{h})$ is commonly referred to as the covariance function and is used to build the matrix specified by $\sigma_0^2\mathbf{R}(\phi)$.

2.2.1.3 Intrinsic stationarity:

The process $Z(\cdot)$ is said to be intrinsically stationary if Condition 1 given in (2.18) holds and in addition the following condition holds.

- **Condition 3**: *For all vectors* $\boldsymbol{h}$ *the difference* $Z(\mathbf{x}+\boldsymbol{h})-Z(\mathbf{x})$ *has a finite variance which depends on the separation vector* $\boldsymbol{h}$ *but not on the location* $\mathbf{x}$*, that is*

$$\text{Var}\{Z(\mathbf{x}+\boldsymbol{h})-Z(\mathbf{x})\}=\text{E}\{[Z(\mathbf{x}+\boldsymbol{h})-Z(\mathbf{x})]^2\}=2\gamma(\boldsymbol{h}), \qquad \forall\ \mathbf{x}.$$

The function $2\gamma(\boldsymbol{h})$ is termed the variogram and half of its value is termed the semivariogram. The semivariogram is thus

$$\gamma(\boldsymbol{h})=\frac{1}{2}\text{Var}\{Z(\mathbf{x}+\boldsymbol{h})-Z(\mathbf{x})\}. \tag{2.20}$$

The semivariogram features prominently in the geostatistical literature and applications and is the main tool used to study spatial correlation rather than the covariance function (Chilés and Delfiner, 1999, Section 2.2; Schabenberger and Gotway, 2005, Section 4.2.2; Diggle and Riberio, 2007, Section 5.2). This is because $C(\cdot)$ is only defined for second order stationary processes whilst $\gamma(\cdot)$ is defined for second order and intrinsic stationary processes.

It can be shown that second order stationarity implies intrinsic stationarity using the relationship between the semivariogram and covariance function as follows. Since

$$\text{Var}\{Z(\mathbf{x})\}=\text{E}\{[Z(\mathbf{x})-\mu]^2\}=\tau^2+C(\mathbf{0}), \quad \forall\ \mathbf{x}\in D$$

where τ^2 is the nugget variance and $C(\mathbf{0})$ is the variance of $Z(\mathbf{x})$ under second order stationarity, it follows that

$$\begin{aligned}\frac{1}{2}\text{Var}\{Z(\mathbf{x}+\boldsymbol{h})-Z(\mathbf{x})\} &= \frac{1}{2}\Big[\text{Var}\{Z(\mathbf{x})\}+\text{Var}\{Z(\mathbf{x}+\boldsymbol{h})\}-2\text{Cov}\{Z(\mathbf{x}),Z(\mathbf{x}+\boldsymbol{h})\}\Big] \\ &= \frac{1}{2}\Big[2\cdot\text{Var}\{Z(\mathbf{x})\}-2\cdot C(\boldsymbol{h})\Big] \\ &= \tau^2+[C(\mathbf{0})-C(\boldsymbol{h})]=\gamma(\boldsymbol{h}).\end{aligned}$$

Although second order stationarity automatically implies intrinsic stationarity the converse is not true. Intrinsic stationarity does not automatically imply second order stationarity (Cressie, 1993; Schabenberger and Gotway, 2005).

Philip and Watson (1986) sharply attack the validity of the assumption of stationarity of $Z(\cdot)$. They state that it is impossible for a spatial process $Z(\cdot)$ to be stationary. That is, it is unrealistic to think that spatial variables can have mean and covariance values which are independent of location (Philip and Watson, 1986). They go on to state that, even as a model property the assumption of stationarity is inappropriate. However, Journel (1986) vehemently defends the stationarity assumption. Journel (1986) states that the assumption does not imply that the actual physical process is itself stationary but is only a model assumption needed for inference as replication is not possible. Specifically, the assumption allows the model to accommodate some desired aspects of the data and in addition facilitates modelling of the spatial dependence. For instance, it enables one to pool data over areas that are believed to be homogenous (Journel, 1986; Goovaerts, 1997).

2.2.2 Gaussian model

A spatial process $Z(\cdot)$ is said to be a Gaussian spatial process if the joint distribution of the n random variables $Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_n)$ is multivariate Gaussian for any integer n and any set of locations $\mathbf{x} \in D \subset \mathbb{R}^d$ (Diggle and Riberio, 2007). Gaussian spatial processes are completely specified by their first two moments, that is their mean function $\mathrm{E}\{Z(\mathbf{x})\} = \mu$ and covariance function $C(\boldsymbol{h})$. For Gaussian spatial processes strict and second order stationarity coincide (Cressie, 1993). This feature of the Gaussian model is very convenient when it comes to inference.

Use of the Gaussian spatial model rarely has any physical justification (Diggle and Riberio, 2007). However this "leap of faith" (Chilès and Delfiner, 1999, p17) is often made. The primary reason for the use of the Gaussian model is its tractability. Gaussian models are "convenient empirical models" that allow one to capture different types of spatial behaviour according to the specification of the correlation structure (Diggle and Riberio, 2007, p46).

Due to increased computational power over the last decade, computer intensive methods of inference are now common. This means that the analytical tractability of Gaussian models is becoming less of an incentive to use them as models for spatial processes (Diggle and Riberio, 2007). For instance simulation based methods of inference have made it possible to carry out inference on spatial processes in the absence of the Gaussian model assumption (Goovaerts, 1997, Chapter 7; Chilès and Delfiner, 1999, Chapter 7; Goméz-Hernádez *et al*, 1997; Pachepsky and Acock, 1998; Goovaerts 1999; 2001).

2.2.3 Parametric models for the spatial covariance structure

The correlation structure of $Z(\cdot)$ can be modelled through parametric functions where the q parameters, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, of the functions correspond to some aspect of the correlation structure. Hence for kriging, $\gamma(\cdot)$ and $C(\cdot)$ are assumed to come from some parametric family of models and the parameters of that function are estimated from the data to give estimates of the functions which are then used in kriging.

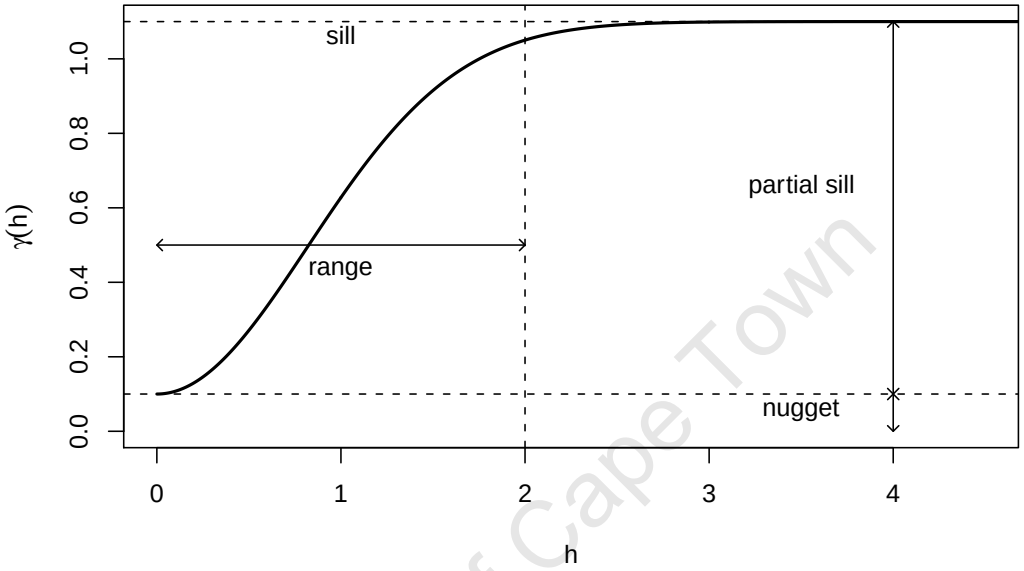


Figure 2.2: A typical semivariogram with the structural parameters indicated.

In order for a function to serve as a parametric model of the spatial correlation it must be continuous. In addition the variance-covariance matrices constructed from it must be non-negative definite. The continuity condition ensures that the models can provide autocorrelation estimates at any desired lag whilst the covariance condition ensures that the functions always lead to non-negative kriging variances. If the functions $C(\cdot)$ and $\gamma(\cdot)$ are functions of only distance they are termed isotropic models. If they are functions of distance and direction they are termed anisotropic models. Some of the most common isotropic functions that are used as parametric models for the spatial correlation in geostatistical applications are given in Table 2.2. Journel and Huijbregts (1978) and Chilès and Delfiner (1999) also give a myriad of other functions. Of the functions given in Table 2.2 the exponential, Gaussian, Matérn and spherical models are the more commonly used models.

Figure 2.2 illustrates how the parameters of the models of the stationary component of spatial processes are captured by semivariogram models. For instance, when a nugget term $\boldsymbol{\epsilon}$ is present, the semivariogram cuts the y-axis at τ^2, the nugget variance. This indicates that at observation locations the variance is greater than zero because of, say, measurement error. The sill σ^2 which is τ^2 plus the partial sill σ_0^2, is the asymptote of $\gamma(\cdot)$ and it corresponds to $\text{Var}\{Z(\mathbf{x})\}$. The range ϕ then indicates how rapidly the function approaches the asymptote or sill.

Table 2.2: Commonly used isotropic $\gamma(\cdot)$ models valid in $\mathbb{R}^2$. K_κ denotes the modified Bessel function of order κ, τ^2 the nugget variance, σ_0^2 the partial sill, ϕ the range parameter, and h the lag between two points.

Model	$\gamma(h)$
Exponential	$\tau^2 + \sigma_0^2\{1 - e^{-3h}/\phi\}$
Gaussian	$\tau^2 + \sigma_0^2\{1 - e^{-3h^2/\phi^2}\}$
Matérn	$\tau^2 + \sigma_0^2\left\{1 - \frac{1}{2^{\kappa-1}\Gamma(\kappa)}\left(\frac{h}{\phi}\right)^{\kappa} K_\kappa\left(\frac{h}{\phi}\right)\right\}$
Cauchy	$\tau^2 + \sigma_0^2\left\{1 - \left[1 + \left(\frac{h}{\phi}\right)^2\right]^{-\kappa}\right\}$
Circular	$\tau^2 + \sigma_0^2\left\{1 - \frac{2}{\pi}\left[\arccos\left(\frac{h}{\phi}\right) - \left(\frac{h}{\phi}\right)\sqrt{1 - \left(\frac{h}{\phi}\right)^2}\,\right]\right\}$, for $h \le \phi$, 0, otherwise
Spherical	$\tau^2 + \sigma_0^2\left\{\frac{3h}{2\phi} - \frac{h^3}{2\phi^3}\right\}$, for $0 < h \le \phi$, $\tau^2 + \sigma_0^2$, for $h \ge \phi$
Cardinal-sine (Wave)	$\tau^2 + \sigma_0^2\left\{1 - \left(\frac{\phi}{h}\right)\sin\left(\frac{h}{\phi}\right)\right\}$

Because some functions never reach the sill but only approach it asymptotically there is a need to define the practical range ϕ_p (Goovaerts, 1997; Schabenberger and Gotway, 2005). The practical range is the value of h at which the function $\gamma(h)$ attains 95% of its sill, that is

$$\gamma(\phi_p) = 0.95 \times (\tau^2 + \sigma_0^2).$$

Also, at the practical range correlations are approximately equal to 0.05, that is $C(\phi_p) \backsimeq 0.05$. The practical range is useful for comparing the correlation for different semivariogram models (Schabenberger and Gotway, 2005). Figure 2.3 shows the shapes of the commonly employed exponential, Gaussian and spherical models with the same sill value and practical range.

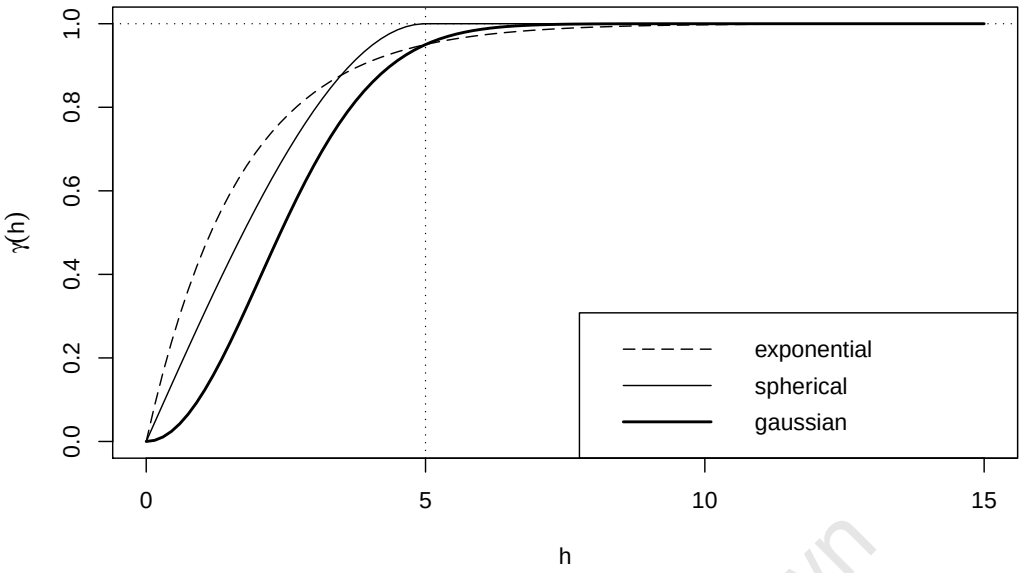


Figure 2.3: Some common semivariogram models. Models have the same range, $\phi = 5$ sill, $\sigma^2 = 1$ and nugget variance $\tau^2 = 0$.

2.3 Estimation of Parameters

2.3.1 Least squares methods

Consider a spatial process $Z(\cdot)$ with an unknown stationary mean, that is following an ordinary kriging model (2.6)

$$\mathbf{Z} = \mathbf{1}\beta + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \sim (\mathbf{0}, \mathbf{V}(\boldsymbol{\theta})).$$

Various least squares (LS) methods have been proposed to estimate the variance components $\boldsymbol{\theta}$ and the mean β of such a process. These include ordinary least squares (OLS), generalized least squares (GLS) and weighted least squares (WLS). These methods are based on fitting curves, that is fitting generic semivariogram models $\gamma(h)$, to M estimates of $\gamma(\cdot)$: $\tilde{\boldsymbol{\gamma}} = [\tilde{\gamma}(h_1), \ldots, \tilde{\gamma}(h_M)]^\top$ the empirical semivariogram, so as to minimize the sum of squares between $\boldsymbol{\gamma} = [\gamma(h_1), \ldots, \gamma(h_M)]^\top$ and $\tilde{\boldsymbol{\gamma}}$. The parameters of the fitted semivariogram models provide estimates $\tilde{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$. The mean β is then obtained by evaluating $\tilde{\beta} = (\mathbf{1}^\top \mathbf{V}(\tilde{\boldsymbol{\theta}})^{-1}\mathbf{1})^{-1}\mathbf{1}^\top \mathbf{V}(\tilde{\boldsymbol{\theta}})^{-1}\mathbf{Z}$.

The M estimates of the empirical semivariogram at each lag are commonly obtained through the Matheron estimator (Matheron, 1963)

$$\tilde{\gamma}(h_{ij}) = \frac{1}{2\ |N(h_{ij})|} \sum_{N(h_{ij})} \{Z(\mathbf{x}_i) - Z(\mathbf{x}_j)\}^2, \qquad h_{ij} \in \mathbb{R}, \tag{2.21}$$

where

$$N(h_{ij}) \equiv \{(\mathbf{x}_i, \mathbf{x}_j) : ||\mathbf{x}_i - \mathbf{x}_j|| = h_{ij}; i, j = 1, \ldots, n\}$$

expresses the set of location pairs $(\mathbf{x}_i, \mathbf{x}_j)$ with coordinate distance $||\mathbf{x}_i - \mathbf{x}_j|| = h_{ij}$ and $|N(h_{ij})|$ is the number of distinct pairs in this set. Typically there are few, if any, pairs of data that are exactly a distance h_{ij} apart. Thus lag classes with class midpoints $h_m, m = 1, \ldots, M$ are formed using a histogram approach (Diggle and Riberio, 2007). This practice is referred to as *binning*.

If there exists a trend in the spatial process, that is the mean is not constant and $Z(\cdot)$ follows the model (2.5)

$$\mathbf{Z} = \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \sim (\mathbf{0}, \mathbf{V}(\boldsymbol{\theta})),$$

the empirical semivariogram based on the data $\mathbf{Z}$ cannot be used in the estimation of $\boldsymbol{\theta}$. This is because in such instances the empirical semivariogram of $\mathbf{Z}$ is dominated by the trend and hence fitting a curve to it leads to poor estimates for $\boldsymbol{\theta}$ (Cressie, 1993; Schabenberger and Gotway, 2005; Diggle and Riberio, 2007). To obtain accurate estimates of $\boldsymbol{\theta}$ the empirical semivariogram of the mean corrected data $\mathbf{Z} - \mathbf{F}\tilde{\boldsymbol{\beta}}$, which are the residuals obtained by fitting an appropriate trend, should be used. The fitting process can be expressed as an algorithm, see Schabenberger and Gotway (2005, p257). Firstly $\boldsymbol{\beta}$ is estimated by the OLS estimator of the mean $\hat{\boldsymbol{\beta}}_{OLS} = (\mathbf{F}^\top\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{Z}$. This then allows the residuals to be estimated as $\mathbf{r} = \mathbf{Z} - \mathbf{F}\hat{\boldsymbol{\beta}}_{OLS}$. Next the empirical semivariogram of the residuals is computed and a generic semivariogram model $\gamma(h)$ is fitted by one of the LS fitting criterion, discussed below, to obtain $\tilde{\boldsymbol{\theta}}$. The covariance parameter estimates $\tilde{\boldsymbol{\theta}}$ are then plugged into the LS estimate for $\boldsymbol{\beta}$ to give $\tilde{\boldsymbol{\beta}} = (\mathbf{F}^\top\mathbf{V}(\tilde{\boldsymbol{\theta}})^{-1}\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{V}(\tilde{\boldsymbol{\theta}})^{-1}\mathbf{Z}$ as an improved estimate of the mean. The estimation of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ may be repeated until the differences between the estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ for successive iterations of the algorithm are suitably small (Schabenberger and Gotway, 2005).

As mentioned earlier, LS methods of fitting semivariogram models to the empirical semivariogram include OLS, GLS and WLS. In OLS the fitting criterion is the sum of squares between observed and fitted semivariogram values

$$(\tilde{\boldsymbol{\gamma}} - \boldsymbol{\gamma})^\top(\tilde{\boldsymbol{\gamma}} - \boldsymbol{\gamma}),$$

which must be minimized with respect to $\boldsymbol{\theta}$. Although the OLS method of semivariogram model fitting is easily implemented, its major shortcoming is its inability to take into account the distributional variation and covariation of the generic estimator $\tilde{\boldsymbol{\gamma}}$ (Cressie, 1985).

The GLS method for fitting curves to the empirical semivariogram is intended to overcome the shortcoming of the OLS method. In GLS the fitting criterion is

$$(\tilde{\boldsymbol{\gamma}} - \boldsymbol{\gamma})^\top\mathbf{V}_\gamma(\boldsymbol{\theta})^{-1}(\tilde{\boldsymbol{\gamma}} - \boldsymbol{\gamma}),$$

where $\mathbf{V}_\gamma(\boldsymbol{\theta}) = \mathrm{Var}\{\tilde{\boldsymbol{\gamma}}\}$ is the variance-covariance matrix of $\tilde{\boldsymbol{\gamma}}$. The minimization process is carried out iteratively if the variance-covariance $\mathbf{V}_\gamma$ does not depend on $\boldsymbol{\theta}$. However, the generalized sum of squares can be difficult to implement as it is not straight forward to obtain $\mathbf{V}_\gamma$ (Cressie, 1985; 1993).

The formulation of the WLS fitting criterion is similar to that for GLS semivariogram fitting. However, the WLS approach uses only the diagonals of $\mathbf{V}_\gamma(\boldsymbol{\theta})$ which can be approximated as

$$\text{Var}\{\tilde{\gamma}(h_m)\} \approx 2\frac{\gamma(h_m)^2}{|N(h_m)|},$$

where $|N(h_m)|$ expresses the number of pairs in lag class h_m, $m = 1, \ldots, M$ (Cressie, 1985). Thus the variance-covariance matrix of $\tilde{\boldsymbol{\gamma}}$, $\mathbf{V}_\gamma(\boldsymbol{\theta})$, is replaced by a diagonal matrix $\mathbf{W}_\gamma(\boldsymbol{\theta}) = \text{diag}\{\text{Var}\{\tilde{\gamma}(h_m)\}\}$. The WLS fitting criterion becomes

$$(\tilde{\boldsymbol{\gamma}} - \boldsymbol{\gamma})^\top \mathbf{W}_\gamma(\boldsymbol{\theta})^{-1}(\tilde{\boldsymbol{\gamma}} - \boldsymbol{\gamma}).$$

The above expression can be minimized through any iterative non-linear estimation algorithm. An attractive feature of WLS semivariogram model fitting is that the weighting scheme automatically assigns more weight to small lags and to lags where there are more pairs and down-weights lags with a small number of pairs (Cressie, 1985).

The performance of the various LS methods of model fitting to the empirical semivariogram is comparable. For instance, OLS and WLS methods perform more or less similarly (Zimmerman and Zimmerman, 1991). Although $\mathbf{W}_\gamma(\boldsymbol{\theta})$ gives a poor approximation to $\mathbf{V}_\gamma(\boldsymbol{\theta})$, Cressie and Grondona (1992) show that there is not much loss in efficiency from using WLS instead of GLS in terms of estimating the parameters. Therefore no method is uniformly superior. There are other more complex LS methods of fitting semivariogram models, all of which are based on these three variants, OLS, GLS and WLS, of least squares fitting. These methods are often superior to their simpler counterparts but they are complex and difficult to implement (Zimmerman and Zimmerman, 1991).

Least squares (LS) methods of estimating $\boldsymbol{\theta}$ are popular for several reasons (Cressie, 1993; Schabenberger and Gotway, 2005);

- They make no distributional assumptions about the data $\mathbf{Z}$ or empirical semivariogram $\tilde{\boldsymbol{\gamma}}$.
- They are quick and easy to use as they do not require much computational power.
- They produce semivariogram models that give visually appealing fits to the empirical semivariogram.

A major shortcoming of LS methods of model fitting is that estimation of the semivariogram at the origin involves extrapolation if no duplicate measurements are available at some of the observation locations (Diggle and Riberio, 2007). Other criticisms of LS methods are due to the *binning* process. The *binning* process gives rise to serious concern. It results in semivariogram models being fitted to pseudo-data instead of the actual data $\mathbf{Z} = [Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_n)]^\top$. This can lead to an analyst

inadvertently "constructing" data that fit a particular semivariogram model instead of looking for a model which fits the data (Stein, 1999; Schabenberger and Gotway, 2005).

2.3.2 Likelihood methods

The most widely used likelihood methods for obtaining estimates of the parameters in a spatial linear model are maximum likelihood (ML) and restricted maximum likelihood (REML) estimation. Likelihood methods require that the spatial distribution of $Z(\cdot)$ be known and a Gaussian distribution is usually assumed (Schabenberger and Gotway, 2005). An assumption of second-order stationarity is not made explicitly but is implied in that for the Gaussian model, intrinsic and second order stationarity coincide (see Section 2.2.1). The likelihood approach to estimation is the same whether the model has a constant or a non-constant mean, so a general development of the methods is given here.

For ML estimation, we assume the model

$$\mathbf{Z} \sim G(\mathbf{F}\boldsymbol{\beta}, \mathbf{V}(\boldsymbol{\theta})),$$

which indicates that the data are assumed to follow a Gaussian distribution with mean $\mathbf{F}\boldsymbol{\beta}$ and variance-covariance $\mathbf{V}(\boldsymbol{\theta})$. Estimation of $\boldsymbol{\beta}$ and $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_q]^\top$ involves maximizing the log-likelihood

$$l(\boldsymbol{\theta}; \boldsymbol{\beta}, \mathbf{Z}) = -\frac{1}{2}\Big[n \ln\{2\pi\} + \ln\{|\mathbf{V}(\boldsymbol{\theta})|\} + (\mathbf{Z} - \mathbf{F}\boldsymbol{\beta})^\top \mathbf{V}(\boldsymbol{\theta})^{-\mathbf{1}}(\mathbf{Z} - \mathbf{F}\boldsymbol{\beta})\Big]. \quad (2.22)$$

If $\mathbf{F}$ is an $n \times (K+1)$ matrix of rank $K+1$, then the optimization problem involves $q + (K+1)$ parameters.

To ease the computational burden the mean $\boldsymbol{\beta}$ and partial sill σ_0^2 may be profiled from (2.22) so that numerical optimization is done over a reduced parameter space with only $q-1$ dimensions instead of the $q + (K+1)$ dimensions of the original problem. Profiling is achieved by making use of the fact that there exists closed form expressions for the ML estimates of $\boldsymbol{\beta}$ and σ_0^2 in terms of the other parameters. These are

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}^\top \mathbf{V}(\boldsymbol{\theta}^\dagger)^{-1}\mathbf{F})^{-1}\mathbf{F}^\top \mathbf{V}(\boldsymbol{\theta}^\dagger)^{-1}\mathbf{Z} \quad (2.23)$$

and

$$\hat{\sigma}_0^2 = \frac{1}{n}(\mathbf{Z} - \mathbf{F}\hat{\boldsymbol{\beta}})^\top \mathbf{V}(\boldsymbol{\theta}^\dagger)^{-1}(\mathbf{Z} - \mathbf{F}\hat{\boldsymbol{\beta}}), \quad (2.24)$$

respectively, where $\boldsymbol{\theta}^\dagger$ is the $q-1$ vector of the parameters with the elements altered to reflect the factoring of σ_0^2. The matrix $\mathbf{V}(\boldsymbol{\theta}^\dagger) = \mathbf{R}(\phi) + \nu^2\mathbf{I}_n$ is obtained by defining $\nu^2 = \tau^2/\sigma_0^2$ so that $\mathbf{V}(\boldsymbol{\theta}) = \sigma_0^2\mathbf{V}(\boldsymbol{\theta}^\dagger)$ (Diggle and Riberio, 2007).

The expressions for $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}_0^2$ in (2.23) and (2.24) are substituted into the log-likelihood (2.22) to give

$$l_{\beta,\sigma_0^2}(\boldsymbol{\theta}^\dagger; \mathbf{Z}) = -\frac{1}{2}\Big[n(1 + \ln\{2\pi\} + \ln\{\hat{\sigma}_0^2\}) + \ln\{|\mathbf{V}(\boldsymbol{\theta}^\dagger)^{-1}|\}\Big], \quad (2.25)$$

the objective function profiled for $\boldsymbol{\beta}$ and σ_0^2. Numerical optimization is then carried out on (2.25) with respect to $\boldsymbol{\theta}^\dagger$ to give ML estimates for $\boldsymbol{\theta}^\dagger$ as $\hat{\boldsymbol{\theta}}^\dagger_{ML}$. Upon convergence the ML estimate of the mean $\hat{\boldsymbol{\beta}}_{ML}$ is obtained by evaluating (2.23) at $\hat{\boldsymbol{\theta}}^\dagger_{ML}$. Then, the ML estimate of σ_0^2 is obtained by evaluating (2.24) at $\hat{\boldsymbol{\beta}}_{ML}$ and $\hat{\boldsymbol{\theta}}^\dagger_{ML}$. Besides reducing the optimization problem to a smaller and more easily managed problem, profiling is also beneficial in that maximizing the full log-likelihood (2.22) can lead to numerical instabilities (Fang *et al*, 2006).

Maximum likelihood estimators (MLE's) have many desirable statistical properties. For instance, they are asymptotically Gaussian and efficient under standard regularity conditions (Diggle and Riberio, 2007). However, the ML estimates for the variance components $\hat{\boldsymbol{\theta}}_{ML}$ are usually biased downwards. This bias arises from the fact that ML estimation does not take into consideration the loss of degrees of freedom in estimating $\boldsymbol{\theta}$ that is incurred when simultaneously estimating mean parameters (Cressie, 1993; Schabenberger and Gotway, 2005).

To overcome the problem of biased estimates for $\boldsymbol{\theta}$ arising from ML estimation restricted maximum likelihood estimation (REML) can be used instead. REML controls the bias of the MLE's and in some cases completely eliminates it (Patterson and Thompson, 1971). REML estimators are obtained by applying ML estimation to transformed data $\mathbf{KZ}$, where $\mathbf{K}$ is an $(n-K-1)\times n$ matrix of error contrasts, instead of the actual data $\mathbf{Z}$. The matrix $\mathbf{K}$ is chosen so that

$$\mathrm{E}\{\mathbf{KZ}\} = \mathbf{0}$$

and

$$\mathrm{rank}\{\mathbf{K}\} = n - K - 1,$$

that is, such that $\mathbf{K}$ is an $(n-K-1)\times n$ matrix whose rows are linearly independent of the rows of

$$\mathbf{I}_n - \mathbf{F}(\mathbf{F}^\top\mathbf{F})^{-1}\mathbf{F}^\top. \tag{2.26}$$

Such a matrix will not depend on $\boldsymbol{\theta}$ or $\boldsymbol{\beta}$ (Harville, 1977).

For REML estimation we thus have the model

$$\mathbf{KZ} \sim G(\mathbf{0}, \mathbf{KV}(\boldsymbol{\theta})\mathbf{K}^\top)$$

and the log-likelihood function which must be minimized is

$$l_r(\boldsymbol{\theta}; \mathbf{KZ}) = -\frac{1}{2}\Big[(n-K-1)\ln\{2\pi\} + \ln\{|\mathbf{KV}(\boldsymbol{\theta})\mathbf{K}^\top|\} + \mathbf{Z}^\top\mathbf{K}^\top(\mathbf{KV}(\boldsymbol{\theta})\mathbf{K}^\top)^{-1}\mathbf{KZ}\Big].$$

This REML objective function is independent of the choice of $\mathbf{K}$ and can be expressed by eliminating $\mathbf{K}$ to give

$$\begin{aligned} l_r(\boldsymbol{\theta}; \mathbf{KZ}) \;=\; & -\frac{1}{2}\Big[(n-K-1)\ln\{2\pi\} + \ln\{|\mathbf{F}^\top\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{F}|\} + \ln\{|\mathbf{V}(\boldsymbol{\theta})|\} \\ & - \ln\{|\mathbf{F}^\top\mathbf{F}|\} + (\mathbf{Z}-\mathbf{F}\hat{\boldsymbol{\beta}})^\top\mathbf{V}(\boldsymbol{\theta})^{-1}(\mathbf{Z}-\mathbf{F}\hat{\boldsymbol{\beta}})\Big]. \end{aligned} \tag{2.27}$$

As in ML estimation, σ_0^2 may be profiled from the REML objective function (2.27). This process is similar to profiling in ML estimation as it also exploits the closed form expression for σ_0^2 in terms of $\hat{\boldsymbol{\theta}}^\dagger$ and $\hat{\boldsymbol{\beta}}$, which in this case is

$$\hat{\sigma}_0^2 = \frac{1}{n-K-1}(\mathbf{Z}-\mathbf{F}\hat{\boldsymbol{\beta}})^\top \mathbf{V}(\hat{\boldsymbol{\theta}}^\dagger)^{-1}(\mathbf{Z}-\mathbf{F}\hat{\boldsymbol{\beta}}).$$

The divisor $(n-K-1)$ in $\hat{\sigma}_0^2$ adjusts the estimate of σ_0^2 for the $(K+1)$ mean parameters $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_K]^\top$. Substitution of the expression for the REML estimate of σ_0^2 into (2.27) then results in the profile restricted log-likelihood

$$\begin{aligned} l_r(\boldsymbol{\theta}^\dagger; \mathbf{K}Z) &= -\frac{1}{2}\Big[(n-K-1)(1+\ln\{2\pi\}+\ln\{\hat{\sigma}_0^2\}) + \ln\{|\mathbf{V}(\boldsymbol{\theta}^\dagger)|\} \\ &\quad + \ln\{|\mathbf{F}^\top \mathbf{V}(\boldsymbol{\theta}^\dagger)^{-1}\mathbf{F}|\}\Big] \end{aligned} \tag{2.28}$$

which is a function of $\boldsymbol{\theta}^\dagger$ only. The REML estimate of the mean $\hat{\boldsymbol{\beta}}_{REML}$ is obtained by evaluating (2.23) at $\hat{\boldsymbol{\theta}}^\dagger_{REML}$. A major shortcoming of REML estimation is the fact the it is developed only for the case where the mean is a linear function of the parameters, that is for Gaussian linear models (Schabenberger and Gotway, 2005). Thus if the mean is not a linear function it is uncertain how the matrix of error contrasts $\mathbf{K}$ may be constructed.

Clearly likelihood methods are heavily dependent on the Gaussian assumption. However this is not a major problem as transformations, such as the Box-Cox family of transformations, can be applied to the data so that $\mathbf{Z}$ is approximately Gaussian (Cressie, 1993; Schabenberger and Pierce, 2002; Diggle and Riberio, 2007). Also, if the data are non-Gaussian but from an exponential family, generalized linear models (GLM) may be introduced to estimate the parameters of $Z(\cdot)$ (Diggle and Riberio, 2007). The only real problems of likelihood estimation are computational, especially for large data sets (Diggle and Riberio, 2007).

Inspite of the problems of likelihood estimation methods of model fitting, the approach possesses some very desirable properties.

- For likelihood methods $\boldsymbol{\theta}$ is estimated based on the actual data instead of pseudo-data by avoiding the *binning* process in constructing an empirical semivariogram (Stein, 1999). The methods thus optimize an objective function, such as (2.22) or (2.27), in the space of the data $\mathbf{Z}$ and the model residuals have a sensible interpretation (Todini and Ferraresi, 1996).
- Likelihood methods lead to a statistical basis for inferences on parameters. Specifically the standard errors of the estimates of the parameters can be obtained as the squared rooted terms on the diagonal of the large sample covariance matrix. In addition, confidence bounds can be estimated. This is typically done using the profile likelihood (Diggle and Riberio, 2007).

- Likelihood methods are the only methods that produce a reliable basis for model comparisons based on statistical grounds (Schabenberger and Gotway, 2005). Models can be compared using the values of the Akaike Information Criterion (AIC)

$$\text{AIC} = -2\ln\{\hat{L}\} + 2p$$

or the Bayesian Information Criterion (AIC)

$$\text{BIC} = -2\ln\{\hat{L}\} + p\log\{n\},$$

where $\hat{L}$ is the value of the maximized log likelihood, p is the number of parameters in the model and n is the number of observations. Using these criteria a model is said to be the best fitting model for a data set if it has the smallest AIC or BIC value amongst competing models.

2.4 The Estimated Kriging Variance

Recall that if the covariance parameter vector $\boldsymbol{\theta}$ is known, the kriging predictor at the location $\mathbf{x}_0$ is given by

$$\begin{aligned}\hat{Z}(\mathbf{x}_0) &= \hat{Z}_1(\mathbf{x}_0;\boldsymbol{\theta}) \\ &= \mathbf{f}_0^\top\hat{\boldsymbol{\beta}} + \mathbf{v}(\boldsymbol{\theta})^\top\mathbf{V}(\boldsymbol{\theta})^{-1}(\mathbf{Z}-\mathbf{F}\hat{\boldsymbol{\beta}})\end{aligned}$$

where $\hat{\boldsymbol{\beta}} = (\mathbf{F}^\top\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{Z}$, and is termed the best linear unbiased predictor (BLUP). This predictor has mean squared prediction error (MSPE)

$$m_1(\boldsymbol{\theta}) = \text{E}\{[\hat{Z}_1(\mathbf{x}_0;\boldsymbol{\theta}) - Z(\mathbf{x}_0)]^2\}$$

which is given by (2.13), that is

$$\sigma^2 - \mathbf{v}(\boldsymbol{\theta})^\top\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{v}(\boldsymbol{\theta}) + (\mathbf{f}_0^\top - \mathbf{v}(\boldsymbol{\theta})^\top\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{F})(\mathbf{F}^\top\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{F})^{-1}(\mathbf{f}_0 - \mathbf{F}^\top\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{v}(\boldsymbol{\theta})).$$

Also recall that since the parameter vector $\boldsymbol{\theta}$ is in general unknown, it has to be estimated by, say, one of the methods described in Section 2.3. The kriging predictor at $\mathbf{x}_0$ is then obtained by plugging-in the estimate of $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}$, into (2.12) to produce $\hat{Z}_2(\mathbf{x}_0;\hat{\boldsymbol{\theta}})$ which is termed the empirical best linear unbiased predictor (EBLUP). The estimator $\hat{\boldsymbol{\theta}}$ is commonly a translation-invariant estimator, that is $\hat{\boldsymbol{\theta}}(\mathbf{Z}+\mathbf{F}\boldsymbol{\psi}) = \hat{\boldsymbol{\theta}}(\mathbf{Z})$ for every vector $\boldsymbol{\psi}$ and all values of $\mathbf{Z}$, such as $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$. The EBLUP has MSPE error

$$m_2(\boldsymbol{\theta}) = \text{E}\{[\hat{Z}_2(\mathbf{x}_0;\hat{\boldsymbol{\theta}}) - Z(\mathbf{x}_0)]^2\}$$

which is a function of the unknown parameters $\boldsymbol{\theta}$.

Following Kacker and Harville (1984) the prediction error of the EBLUP can be seen to consist of two parts:

$$\hat{Z}_2(\mathbf{x}_0) - Z(\mathbf{x}_0) = [\hat{Z}_1(\mathbf{x}_0) - Z(\mathbf{x}_0)] + [\hat{Z}_2(\mathbf{x}_0) - \hat{Z}_1(\mathbf{x}_0)], \qquad (2.29)$$

where $\hat{Z}_2(\mathbf{x}_0)$ and $\hat{Z}_1(\mathbf{x}_0)$ are the EBLUP and BLUP respectively, expressed without $\boldsymbol{\theta}$ for economy of notation. Following from (2.29) we have

$$\begin{aligned} m_2(\boldsymbol{\theta}) &= m_1(\boldsymbol{\theta}) + \mathrm{E}\{[\hat{Z}_2(\mathbf{x}_0) - \hat{Z}_1(\mathbf{x}_0)]^2\} + 2\mathrm{E}\{\hat{Z}_1(\mathbf{x}_0) - Z(\mathbf{x}_0), \hat{Z}_2(\mathbf{x}_0) - \hat{Z}_1(\mathbf{x}_0)\} \\ &= m_1(\boldsymbol{\theta}) + \mathrm{Var}\{[\hat{Z}_2(\mathbf{x}_0) - \hat{Z}_1(\mathbf{x}_0)]\} + 2\mathrm{Cov}\{\hat{Z}_1(\mathbf{x}_0) - Z(\mathbf{x}_0), \hat{Z}_2(\mathbf{x}_0) - \hat{Z}_1(\mathbf{x}_0)\} \end{aligned}$$

since $\hat{Z}_1(\mathbf{x}_0)$ is unbiased, that is $\mathrm{E}\{\hat{Z}_1(\mathbf{x}_0)\} = \mathbf{f}_0^\top \boldsymbol{\beta}$. If $\mathbf{Z}$ is Gaussian then

$$\mathrm{Cov}\{\hat{Z}_1(\mathbf{x}_0) - Z(\mathbf{x}_0), \hat{Z}_2(\mathbf{x}_0) - \hat{Z}_1(\mathbf{x}_0)\} = 0$$

and $m_2(\boldsymbol{\theta})$ can be simply be expressed as

$$m_2(\boldsymbol{\theta}) = m_1(\boldsymbol{\theta}) + \mathrm{Var}\{[\hat{Z}_2(\mathbf{x}_0) - \hat{Z}_1(\mathbf{x}_0)]\}. \tag{2.30}$$

The first term in (2.30) is the prediction error of the BLUP and the second term is the error incurred in the estimation of $\boldsymbol{\theta}$ with $m_2(\boldsymbol{\theta}) \geq m_1(\boldsymbol{\theta})$ (Zimmerman and Cressie, 1992).

Since $m_2(\boldsymbol{\theta})$ in (2.30) cannot be evaluated directly, at least in general, it is commonly estimated by the empirical mean squared prediction error (EMSPE). This traditional estimator of $m_2(\boldsymbol{\theta})$ is obtained by plugging in $\hat{\boldsymbol{\theta}}$ into (2.13) to give

$$\begin{aligned} m_1(\hat{\boldsymbol{\theta}}) &= \sigma^2 - \mathbf{v}(\hat{\boldsymbol{\theta}})^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1} \mathbf{v}(\hat{\boldsymbol{\theta}}) + (\mathbf{f}_0^\top - \mathbf{v}(\hat{\boldsymbol{\theta}})^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1} \mathbf{F})(\mathbf{F}^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1} \mathbf{F})^{-1} \\ &\quad \times (\mathbf{f}_0 - \mathbf{F}^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1} \mathbf{v}(\hat{\boldsymbol{\theta}})). \end{aligned}$$

However, a major shortcoming of $m_1(\hat{\boldsymbol{\theta}})$ as an estimator of $m_2(\boldsymbol{\theta})$ is that on average it results in the underestimation of the prediction error of the EBLUP (Zimmerman and Cressie, 1992; Cressie, 1993; Wang and Wall, 2003; Schabenberger and Gotway, 2005; den Hertog *et al*; 2006). This is because it does not take into account the variability due to the estimation of $\boldsymbol{\theta}$ that is introduced into the EBLUP . It is also evident that $m_1(\hat{\boldsymbol{\theta}})$ estimates $\mathrm{E}\{[\hat{Z}_1(\mathbf{x}_0) - Z(\mathbf{x}_0)]^2\}$ the prediction error of the BLUP and not $m_2(\boldsymbol{\theta})$, the mean squared prediction error *it should be estimating* (Zimmerman and Cressie, 1992; Schabenberger and Gotway, 2005). Therefore $m_1(\hat{\boldsymbol{\theta}})$ is not only an underestimator of $m_2(\boldsymbol{\theta})$ but it is in fact an estimator of an incorrect quantity.

Other results obtained by Zimmerman and Cressie (1992) which relate $\mathrm{E}\{m_1(\boldsymbol{\theta})\}$ to $m_1(\hat{\boldsymbol{\theta}})$ and $m_2(\boldsymbol{\theta})$ and thus give an indication of the bias in using $m_1(\boldsymbol{\theta})$ to estimate $m_2(\hat{\boldsymbol{\theta}})$ are:

- If the covariance function $C(h)$ is linear in the elements $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ is unbiased, then
$$\mathrm{E}\{m_1(\hat{\boldsymbol{\theta}})\} \leq m_1(\boldsymbol{\theta}),$$
that is the EMSPE underestimates the MSPE itself.

- If $\mathbf{Z}$ is Gaussian, $C(h)$ is linear in the elements $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ is unbiased, then
$$\mathrm{E}\{m_1(\hat{\boldsymbol{\theta}})\} \leq m_1(\boldsymbol{\theta}) \leq m_2(\boldsymbol{\theta}),$$

- If $\mathbf{Z}$ is Gaussian, $C(h)$ is linear in the elements $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ is complete and sufficient, then

$$\mathrm{E}\{m_1(\hat{\boldsymbol{\theta}})\} = m_2(\boldsymbol{\theta}) - 2[m_2(\boldsymbol{\theta}) - m_1(\boldsymbol{\theta})].$$

The final result implies that the bias of $m_1(\hat{\boldsymbol{\theta}})$ is of magnitude $2[m_2(\boldsymbol{\theta}) - m_1(\boldsymbol{\theta})]$ as bias due to $\mathrm{E}\{m_1(\hat{\boldsymbol{\theta}})\} \leq m_2(\boldsymbol{\theta})$ and the fact that $m_1(\boldsymbol{\theta}) \leq m_2(\boldsymbol{\theta})$ contribute equally to the total bias of $m_1(\hat{\boldsymbol{\theta}})$. The result also gives an idea of the form of an unbiased estimator of $m_2(\boldsymbol{\theta})$. However, the conditions under which such an estimator can be obtained are stringent and rarely hold.

2.4.1 Alternate analytical estimators to the EMSPE

From the discussion above it is clear that $m_1(\hat{\boldsymbol{\theta}})$, the EMSPE is an unsatisfactory estimator of $m_2(\boldsymbol{\theta})$, the MSPE of the EBLUP. There is therefore a need for alternative methods of estimating $m_2(\boldsymbol{\theta})$. One such estimator is based on an approximation that originates from work done on mixed linear models where it was first suggested by Kacker and Harville (1984) for use in mixed linear models with estimated variance parameters. Later Harville and Jeske (1992) as well as Zimmerman and Cressie (1992) gave details and suggested its use in general linear models and spatial linear models with estimated covariance parameters respectively. The approximation is developed as follows.

Firstly, the EBLUP is approximated by a first-order Taylor series expansion of $\hat{Z}_2(\mathbf{x}_0)$ about $\boldsymbol{\theta}$. The Taylor series expansion leads to

$$\hat{Z}_2(\mathbf{x}_0) \approx \hat{Z}_1(\mathbf{x}_0) + \frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\bigg|_{\theta}^{\top} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}),$$

where $\partial \hat{Z}_1(\mathbf{x}_0)/\partial \boldsymbol{\theta}$ is a $q \times 1$ vector with i^{th} element $\partial \hat{Z}_1(\mathbf{x}_0)/\partial \theta_i$. Thus

$$\hat{Z}_2(\mathbf{x}_0) - Z(\mathbf{x}_0) \approx \hat{Z}_1(\mathbf{x}_0) - \hat{Z}(\mathbf{x}_0) + \left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)^{\top} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}).$$

Squaring on both sides and taking expectations then leads to the approximation

$$\begin{aligned} m_2(\boldsymbol{\theta}) \;\approx\; & m_1(\boldsymbol{\theta}) + \mathrm{E}\left\{\left[\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)^{\top} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right]^2\right\} \\ & + 2\ \mathrm{E}\left\{[\hat{Z}_1(\mathbf{x}_0) - Z(\mathbf{x}_0)][\mathbf{Z}^{\top}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})]\right\}. \end{aligned} \tag{2.31}$$

If it is assumed that $\hat{\boldsymbol{\theta}}$ was obtained from previous data as in Kacker and Harville (1984) or if it is assumed that $\hat{\boldsymbol{\theta}}$ and $\mathbf{Z}$ are independent (Harville and Jeske, 1992; Zimmerman and Cressie, 1992), then the approximation can be simply expressed as

$$m_2(\boldsymbol{\theta}) \approx m_1(\boldsymbol{\theta}) + \mathrm{E}\left\{\left[\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)^{\top} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right]^2\right\}. \tag{2.32}$$

Although the assumption of independence between $\hat{\boldsymbol{\theta}}$ and $\mathbf{Z}$ may be unrealistic, the ability of (2.32) to satisfactorily approximate $m_2(\boldsymbol{\theta})$ is not adversely affected. Abt (1999) found that estimators based on approximation (2.32) perform similarly to estimators based on the full approximation of $m_2(\boldsymbol{\theta})$ given in (2.31).

Note that

$$\begin{aligned}
\mathrm{E}\left\{\left[\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)^\top(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})\right]^2\right\} &= \mathrm{E}\left\{\mathrm{tr}\left\{\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)^\top \right.\right. \\
&\quad \left.\left. \times\, (\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^\top\right\}\right\} \\
&= \mathrm{tr}\left\{\mathrm{E}\left\{\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)^\top\right\}\right. \\
&\quad \left. \times\, \mathrm{E}\{(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^\top\}\right\}.
\end{aligned}$$

Now $\mathrm{E}\{\hat{Z}(\mathbf{x}_0)\} = \mathbf{f}_0^\top\boldsymbol{\beta}$ so

$$\mathrm{E}\left\{\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right\} = \frac{\partial}{\partial \boldsymbol{\theta}}\mathrm{E}\{\hat{Z}_1(\mathbf{x}_0)\} = 0,$$

and it follows that

$$\mathrm{E}\left\{\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)^\top\right\} = \mathrm{Var}\left\{\left(\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right)\right\} = \mathbf{A}(\boldsymbol{\theta}).$$

Also

$$\mathrm{E}\{(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^\top\} = \mathrm{MSE}(\hat{\boldsymbol{\theta}}) = \mathbf{B}(\boldsymbol{\theta}),$$

hence the overall approximation is

$$m_2(\boldsymbol{\theta}) \doteq m_1(\boldsymbol{\theta}) + \mathrm{tr}\{\mathbf{A}(\boldsymbol{\theta})\mathbf{B}(\boldsymbol{\theta})\}. \tag{2.33}$$

The second term in (2.33), $\mathrm{tr}\{\mathbf{A}(\boldsymbol{\theta})\mathbf{B}(\boldsymbol{\theta})\}$, approximates $\mathrm{Var}\{[Z_2(\mathbf{x}_0) - Z_1(\mathbf{x}_0)]\}$ the error incurred by the EBLUP due to $\boldsymbol{\theta}$ being unknown.

Now consider the form of the variance matrix $\mathbf{A}(\boldsymbol{\theta})$. Recall that $\hat{Z}_1(\mathbf{x}_0) = \boldsymbol{\lambda}^\top\mathbf{Z}$ where the kriging weights are given by (2.11) as

$$\boldsymbol{\lambda} = \mathbf{V^{-1}v} - \mathbf{V^{-1}F\Delta^{-1}F^\top V^{-1}}(\mathbf{f_0} - \mathbf{F^\top V^{-1}v})$$

with $\boldsymbol{\Delta} = \mathbf{F}^\top\mathbf{V}^{-1}\mathbf{F}$. In the discussion that follows the parameter vector $\boldsymbol{\theta}$ is omitted from $\mathbf{v}(\boldsymbol{\theta})$ and $\mathbf{V}(\boldsymbol{\theta})$ for ease of notation. It thus follows that

$$\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\theta}}\right)^\top\mathbf{Z} \tag{2.34}$$

where $\partial\boldsymbol{\lambda}/\partial\boldsymbol{\theta}$ is an $n \times q$ matrix with columns $\partial\boldsymbol{\lambda}/\partial\theta_i$, $i = 1, \ldots, q$. It can be shown that

$$\frac{\partial \boldsymbol{\lambda}}{\partial \theta_i} = \mathbf{M}_1\left(\frac{\partial \mathbf{v}}{\partial \theta_i} - \frac{\partial \mathbf{V}}{\partial \theta_i}(\mathbf{M}_2\mathbf{f}_0 + \mathbf{M}_1\mathbf{v})\right),$$

where

$$\mathbf{M}_1 = \mathbf{V^{-1}F}(\mathbf{F}^\top\mathbf{V^{-1}F})^{-1} \quad \text{and} \quad \mathbf{M}_2 = \mathbf{V^{-1}} - \mathbf{M_1F}^\top\mathbf{V^{-1}}.$$

It thus follows immediately from (2.34) that

$$\text{Var}\left\{\frac{\partial \hat{Z}_1(\mathbf{x}_0)}{\partial \boldsymbol{\theta}}\right\} = \mathbf{A}(\boldsymbol{\theta}) = \left(\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\theta}}\right)^\top \mathbf{V}\left(\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\theta}}\right). \tag{2.35}$$

Specifically the ij^{th} element of the $q \times q$ symmetric matrix $\mathbf{A}(\boldsymbol{\theta})$ can be seen to be

$$a_{ij} = \left(\frac{\partial \boldsymbol{\lambda}}{\partial \theta_i}\right)^\top \mathbf{V}(\boldsymbol{\theta})\frac{\partial \boldsymbol{\lambda}}{\partial \theta_j} \quad i, j = 1, \ldots, q.$$

Consider now the $q \times q$ mean square matrix $\mathbf{B}(\boldsymbol{\theta})$. This matrix is typically intractable and is usually approximated by the large sample covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ of $\hat{\boldsymbol{\theta}}$ and hence (2.33) takes on the form

$$m_2(\boldsymbol{\theta}) \doteq m_1(\boldsymbol{\theta}) + \text{tr}\{\mathbf{A}(\boldsymbol{\theta})\boldsymbol{\Sigma}(\boldsymbol{\theta})\}. \tag{2.36}$$

The large sample covariance matrix of variance components is, under certain regularity conditions, given by the inverse of the information matrix of $\boldsymbol{\theta}$, $\mathbf{I}(\boldsymbol{\theta})$ (Zimmerman and Cressie, 1992; Abt 1998; Zhu and Stein 2005; Zimmerman, 2006) such that

$$\mathbf{B}(\boldsymbol{\theta}) \doteqdot \boldsymbol{\Sigma}(\boldsymbol{\theta}) \backsimeq \mathbf{I}(\boldsymbol{\theta})^{-1}. \tag{2.37}$$

For $\mathbf{I}(\boldsymbol{\theta})$ associated with $\hat{\boldsymbol{\theta}}_{ML}$ the ij^{th} element of the information matrix is

$$\text{I}_{ij}(\boldsymbol{\theta}) = \frac{1}{2}\text{tr}\left\{\mathbf{V}^{-1}\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{V}^{-1}\frac{\partial \mathbf{V}}{\partial \theta_j}\right\}.$$

For $\mathbf{I}(\boldsymbol{\theta})$ associated with $\hat{\boldsymbol{\theta}}_{REML}$ the ij^{th} element of the information matrix is

$$\text{I}_{ij}(\boldsymbol{\theta}) = \frac{1}{2}\text{tr}\left\{\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_j}\right\}$$

with $\mathbf{P} = \mathbf{V^{-1}} - \mathbf{V^{-1}F}(\mathbf{F}^\top\mathbf{V^{-1}F})^{-1}\mathbf{F}^\top\mathbf{V^{-1}}$ (Harville, 1977; Zimmerman, 2006). The estimator based on approximation (2.36) is termed the Kacker-Harville estimator. If we further use approximation (2.37) then the estimator is given as

$$\sigma^2(\hat{\boldsymbol{\theta}})_{KH} \;=\; m_1(\hat{\boldsymbol{\theta}}) + \text{tr}\{\mathbf{A}(\hat{\boldsymbol{\theta}})\mathbf{I}(\hat{\boldsymbol{\theta}})^{-1}\}, \tag{2.38}$$

and is obtained by plugging in the appropriate $\boldsymbol{\theta}$ estimate into $m_1(\boldsymbol{\theta})$ and $\mathbf{A}(\boldsymbol{\theta})$ and using the Hessian matrix

$$-\left(\frac{\partial^2 l}{\partial \hat{\boldsymbol{\theta}} \partial \hat{\boldsymbol{\theta}}^\top}\right) = -\mathbf{H}(\hat{\boldsymbol{\theta}}) = \mathbf{I}(\hat{\boldsymbol{\theta}}),$$

that is the matrix of second order derivatives of the appropriate likelihood function. The Hessian is usually found numerically. For ML, $\mathbf{I}(\hat{\boldsymbol{\theta}})$ is the Hessian of the Gaussian log-likelihood (2.22) evaluated at $\hat{\boldsymbol{\theta}}_{ML}$ and for REML it is the Hessian of the restricted log-likelihood (2.27) evaluated at $\hat{\boldsymbol{\theta}}_{REML}$.

Abt and Welch (1998) and others (Abt, 1998; Zhu and Stein, 2005; Zimmerman, 2006) have shown that the use of the inverse of $\mathbf{I}(\boldsymbol{\theta})$ to estimate $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ and therefore $\mathbf{B}(\boldsymbol{\theta})$ is appropriate. In a simulation study Zhu and Stein (2005) found that $\mathbf{I}(\boldsymbol{\theta})^{-1}$ provides a fair approximation to $\mathbf{B}(\boldsymbol{\theta})$ when the sample size is "moderately large". However, they also found that when the sample size is small the approximation is poor. If the parameter estimates are on the boundary of the parameter space, namely when the estimated nugget is zero, then $\mathbf{I}(\hat{\boldsymbol{\theta}})^{-1}$ tends to overestimate the standard errors of $\hat{\boldsymbol{\theta}}$ (Abt, 1998; Abt and Welch; 1998). In such situations, Abt (1998) recommends using $\mathbf{I}(\boldsymbol{\theta})^{-1}$ with the first row and first column deleted. That is, deleting all elements involving derivatives of the nugget. Another approach for approximating $\mathbf{B}(\boldsymbol{\theta})$ in (2.33) involves the use of bootstrapping as in Zhu and Stein (2005), Zimmerman (2006) and Gonzáles-Manteiga *et al* (2008). In this approach $\mathbf{B}(\boldsymbol{\theta})$ is approximated as

$$\mathbf{B}^*(\boldsymbol{\theta}) = \frac{1}{B}\sum_{b=1}^{B}(\hat{\boldsymbol{\theta}}_b^* - \hat{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}}_b^* - \hat{\boldsymbol{\theta}})^\top, \quad b = 1, \ldots, B,$$

where $\hat{\boldsymbol{\theta}}$ is the original vector of likelihood parameter estimates and $\hat{\boldsymbol{\theta}}_b^*$ is the parameter vector of the likelihood estimates for the b^{th} bootstrap sample.

Although an improvement on $m_1(\hat{\boldsymbol{\theta}})$, the estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ in (2.38) also results in the underestimation of $m_2(\boldsymbol{\theta})$ if $m_1(\hat{\boldsymbol{\theta}})$ underestimates $m_1(\boldsymbol{\theta})$ (Harville and Jeske, 1992; Zimmerman and Cressie, 1992). A modified estimator for $m_2(\boldsymbol{\theta})$, first proposed by Prasad and Rao (1986; 1990) and then by Harville and Jeske (1992), which is intended to eliminate the negative bias of $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ is the Prasad-Rao estimator

$$\sigma^2(\hat{\boldsymbol{\theta}})_{PR} = m_1(\hat{\boldsymbol{\theta}}) + 2\text{tr}\{\mathbf{A}(\hat{\boldsymbol{\theta}})\mathbf{I}(\hat{\boldsymbol{\theta}})^{-1}\}. \tag{2.39}$$

Although $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$, the Prasad-Rao estimator, reduces the bias seen in $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ the Kacker-Harville estimator, it is also biased as it tends to overestimate the MSPE of the EBLUP $m_2(\boldsymbol{\theta})$ (Zimmerman and Cressie, 1992). The magnitude of the bias varies depending on the closeness of the estimated parameters $\hat{\boldsymbol{\theta}}$ to the true but unknown parameters $\boldsymbol{\theta}$ (Harville and Jeske, 1992).

Other results regarding the performance of $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ are not clear and may be considered somewhat inconclusive. It is reported that the unbiasedness of these estimators in estimating the MSPE of the BLUP, $m_2(\boldsymbol{\theta})$, and the degree of improvement on the EMSPE, $m_1(\hat{\boldsymbol{\theta}})$, depend on how many of and which of the following assumptions are true (Prasad and Rao, 1986; 1990; Harville and Jeske, 1992; Zimmerman and Cressie, 1992):

- The observations $\mathbf{Z}$ follow a multivariate Gaussian distribution.
- $\hat{\boldsymbol{\theta}}$ is unbiased.
- $\hat{\boldsymbol{\theta}}(\mathbf{Z})$ is a complete, sufficient statistic for the distribution of $\mathbf{Z}$.

- The generic covariance function $C(h)$ is a linear function of the elements of $\boldsymbol{\theta}$.

The number of data n, their spatial configuration and the strength of the spatial correlation also play a major role on the quality of estimates obtained from $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ (Zimmerman and Cressie, 1992; Abt, 1999). According to Zimmerman and Cressie (1992), to avoid obtaining estimates that are more biased than those of the EMSPE $m_1(\hat{\boldsymbol{\theta}})$, the Kacker-Hariville and Prasad-Rao estimators should only be used if

- $\mathbf{V}(\boldsymbol{\theta}) - \mathrm{E}[\mathbf{V}(\hat{\boldsymbol{\theta}})]$ is non-negative definite.
- The spatial correlation is weak.

The second condition is based on the notion that moderate-to-strong spatial correlation compensates for the inherent negative bias of $m_1(\hat{\boldsymbol{\theta}})$ (Zimmerman and Zimmerman, 1991; Zimmerman and Cressie; 1992). Thus adding the addtional terms of (2.38) and (2.39) to $m_1(\hat{\boldsymbol{\theta}})$ only serves to worsen the precision of the estimates. However, when the observations $\mathbf{Z}$ are very weakly correlated, the large sample covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta}) \backsimeq \mathbf{I}(\boldsymbol{\theta})^{-1}$ in the term $\mathrm{tr}\{\mathbf{A}(\boldsymbol{\theta})\boldsymbol{\Sigma}(\boldsymbol{\theta})\} \approx \mathrm{Var}\{[\hat{Z}_2(\mathbf{x}_0) - \hat{Z}_1(\mathbf{x}_0)]\}$ overestimates the variability of the parameter estimates such that (2.38) and (2.39) overestimate $m_2(\boldsymbol{\theta})$ (Abt, 1998). Abt (1998) mentions that this may however be countered by increasing n, the size of the sample and by the inclusion of small lags $h_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||$ for model fitting. If feasible, one may also use different designs for model fitting and prediction.

2.4.2 Bootstrap alternatives

Wang and Wall (2003) were the first to propose bootstrapping for estimating the MSPE of the EBLUP, $m_2(\boldsymbol{\theta})$, in the light of some of the constraints associated with the Kacker-Harville and Prasad-Rao estimators as given by (2.38) and (2.39) respectively. The first concern is that in order to compute (2.38) and (2.39), $\hat{\boldsymbol{\theta}}$ must be an estimator for which $\mathbf{B}(\boldsymbol{\theta})$ can be calculated. Secondly, the assumption that the covariance function $C(h)$ is linear in $\boldsymbol{\theta}$ needed for the estimators to be unbiased is not valid for most of the commonly used models of covariance functions such as the exponential, Gaussian and spherical covariance models (Wang and Wall, 2003).

In one approach, Wang and Wall (2003) use bootstrapping for estimating the term $\mathrm{tr}\{\mathbf{A}(\boldsymbol{\theta})\mathbf{B}(\boldsymbol{\theta})\} \approx \mathrm{Var}\{[\hat{Z}_2(\mathbf{x}_0;\hat{\boldsymbol{\theta}}) - \hat{Z}_1(\mathbf{x}_0;\boldsymbol{\theta})]\}$ in (2.33). The term is approximated as

$$s^2 = \frac{1}{B-1}\sum_{b=1}^{B}\{\hat{Z}(\mathbf{x}_0;\hat{\boldsymbol{\theta}}_{\boldsymbol{b}}^{*}) - \hat{Z}_2(\mathbf{x}_0;\hat{\boldsymbol{\theta}})\}^2, \quad b = 1,\ldots,B, \tag{2.40}$$

where $\hat{Z}_2(\mathbf{x}_0;\hat{\boldsymbol{\theta}}) = \boldsymbol{\lambda}^{\top}(\hat{\boldsymbol{\theta}})\mathbf{Z}$ is the EBLUP, with the dependence of the kriging weights on $\boldsymbol{\theta}$ indicated explicitly, and $\hat{Z}(\mathbf{x}_0;\hat{\boldsymbol{\theta}}_b^{*}) = \boldsymbol{\lambda}^{\top}(\hat{\boldsymbol{\theta}}_b^{*})\mathbf{Z}$ is the kriging prediction made

using a kriging model fitted with the estimated parameters $\hat{\boldsymbol{\theta}}_b^*$ from the b^{th} bootstrap sample. Note that $\hat{Z}(\mathbf{x}_0; \hat{\boldsymbol{\theta}}_b^*)$ uses the original and not the bootstrapped data. According to Wang and Wall (2003), making the prediction with the bootstrapped parameter estimates $\hat{\boldsymbol{\theta}}_{\boldsymbol{b}}^*$ and the original data $\mathbf{Z}$ instead of the bootstrapped data $\mathbf{Z}^*$ captures the variability due to $\hat{\boldsymbol{\theta}}$ only such that

$$s^2 \approx \text{Var}\{[\hat{Z}_2(\mathbf{x}_0; \hat{\boldsymbol{\theta}}) - \hat{Z}_1(\mathbf{x}_0; \boldsymbol{\theta})]\}.$$

The approximation s^2 can then be used to calculate bootstrap equivalents of the $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ in (2.38) and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ in (2.39). The bootstrap version of the Kacker-Harville estimator is then given as $m_1(\hat{\boldsymbol{\theta}}) + s^2$ whilst the bootstrap version of the Prasad-Rao estimator is given as $m_1(\hat{\boldsymbol{\theta}}) + 2s^2$. This approach of obtaining the Kacker-Harville and Prasad-Rao estimators seems, arguably, rather contrived and appears not have been used by other researchers.

In another approach Wang and Wall (2003) use bootstrapping, in particular parametric simulation, to estimate $m_2(\boldsymbol{\theta})$ directly by estimating the distribution of $\text{E}\{[\hat{Z}_2(\mathbf{x}_0) - Z(\mathbf{x}_0)]^2\}$. This approach has been adopted by Zimmerman (2006) and den Hertog *et al* (2006). The generic algorithm behind the approach is straight forward. Suppose that data $\mathbf{Z}$ are observed at the n locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Firstly, the distribution of $\mathbf{Z}$ is estimated. To calculate the estimated kriging variance at a test point, $\mathbf{x}_0$, the algorithm proceeds by drawing the b^{th} parametric bootstrap sample in the observed data locations and at $\mathbf{x}_0$ simultaneously from the fitted distributional model. Next a kriging model is fitted to the bootstrapped data corresponding to the observation locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and used to make a prediction $\hat{Z}_b^*(\mathbf{x}_0)$ at the test point $\mathbf{x}_0$. Lastly, the squared error between the predicted value $\hat{Z}_b^*(\mathbf{x}_0)$ and the simulated value $Z_b^*(\mathbf{x}_0)$ at the test point $\mathbf{x}_0$ is calculated. These steps are repeated for $b = 1, \ldots, B$ and the squared errors between the kriging prediction at $\mathbf{x}_0$ for the b^{th} simulation and the b^{th} simulated value at $\mathbf{x}_0$ are then averaged over the number of iterations B to give a bootstrap estimate of the kriging variance at the test location $\mathbf{x}_0$. The generic bootstrap kriging variance estimator is thus given by

$$\sigma^2(\mathbf{x}_0; \hat{\boldsymbol{\theta}})_{BS} = \frac{1}{B} \sum_{b=1}^{B} \{\hat{Z}_b^*(\mathbf{x}_0) - Z_b^*(\mathbf{x}_0)\}^2, \quad b = 1, \ldots, B. \tag{2.41}$$

Zimmerman (2006) calls this type of bootstrapped MSPE estimate of the EBLUP the "finite sample prediction error variance". Wang and Wall (2003) use the bootstrapped MSPE estimate to set confidence bounds for predictions and to examine the coverage probabilities of these bounds.

Confidence bounds for the bootstrapped MSPE itself may also be set based on the central limit theorem (see Efron and Tibshirani, 1993). Den Hertog *et al* (2006) give the $(1 - \alpha)100\%$ bootstrap confidence interval for $\sigma^2(\mathbf{x}_0; \hat{\boldsymbol{\theta}})_{BS}$ as

$$\sigma^2(\mathbf{x}_0; \hat{\boldsymbol{\theta}})_{BS} \pm z_{\alpha/2} \sqrt{\frac{\hat{\sigma}^2_{SE}\{\hat{Z}^*(\mathbf{x}_0)\}}{B}}, \tag{2.42}$$

where $\hat{\sigma}^2_{SE}\{\hat{Z}^*(\mathbf{x}_0)\}$ is the sample variance of the bootstrapped squared errors, that is

$$\hat{\sigma}^2_{SE}\{\hat{Z}^*(\mathbf{x}_0)\} = \frac{1}{B-1}\sum_{b=1}^{B}\left(\{\hat{Z}^*_b(\mathbf{x}_0) - Z^*_b(\mathbf{x}_0)\}^2 - \sigma^2(\mathbf{x}_0;\hat{\boldsymbol{\theta}})_{BS}\right)^2. \quad (2.43)$$

The performance of the bootstrap method and therefore the estimator (2.41) relies on whether the distribution from which the bootstrap samples are generated is correctly specified (Wang and Wall, 2003). Commonly, the spatial process $Z(\cdot)$ is assumed to be a Gaussian process and the observations $\mathbf{Z}$ obtained at the locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are assumed to come from a multivariate Gaussian distribution, that is $\mathbf{Z} \sim G(\mathbf{F}\boldsymbol{\beta}, \mathbf{V}(\boldsymbol{\theta}))$. According to Wang and Wall (2003) this method performs well for normal data as well as for exponentiated data which are incorrectly assumed to be Gaussian, if the sample is large. They warn, however, that the method may not perform as favourably for other types of non-normal data if the distribution from which bootstrap samples are generated is incorrectly specified.

Den Hertog *et al* (2006) take the bootstrap approach of directly estimating $m_2(\boldsymbol{\theta})$ further by presenting various specific algorithms for generating the bootstrap samples. The algorithms are designed to be used in different practical applications and the major difference between these algorithms is in how and where bootstrap estimates of the kriging variance are to be made and thus how bootstrap samples are generated. There are three algorithms which are summarized as follows:

- Fixed test set algorithm
  For use when there is a finite set of prespecified prediction locations at which the kriging variance is to be estimated. Hence bootstrap samples are taken at the n observation locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and at m test locations $\mathbf{x}_{01}, \ldots, \mathbf{x}_{0m}$ simultaneously.

- Variable test set algorithm
  For use when the locations at which the kriging variance is to be estimated are not known beforehand. It is also used when the kriging variance has been estimated at some points and there is a need to estimate at further, that is additional, locations.

- Algorithm for adding test points one at a time
  For use when it is not known beforehand at which locations the kriging variance is to be estimated and such locations will be introduced later, one at a time.

In the next sub-sections the exact steps of two versions of the fixed test set algorithm are set out. The variable test set algorithm and the algorithm for adding test points one at a time are beyond the scope of the current research. The first algorithm is the original fixed test set algorithm of den Hertog *et al* (2006) where unconditional simulation is used in the bootstrap process. The second algorithm is a proposal of an alternate algorithm in which conditional instead of unconditional simulation is employed.

2.4.2.1 Fixed test set: Unconditional simulation

The unconditional bootstrap algorithm for the estimation of the MSPE of the EBLUP, $m_2(\boldsymbol{\theta})$, is presented in Table 2.3. The required bootstrap sample in Step 2 is denoted

$$\mathbf{Z}^* = [Z^*(\mathbf{x}_1), \ldots, Z^*(\mathbf{x}_n), Z^*(\mathbf{x}_{01}), \ldots, Z^*(\mathbf{x}_{0m})]^\top$$

and is obtained by sampling from the multivariate Gaussian distribution (den Hertog *et al*, 2006)

$$G(\mathbf{F}_s\hat{\boldsymbol{\beta}}, \mathbf{V}_s(\hat{\boldsymbol{\theta}})),$$

where

$$\mathbf{F}_s = \begin{bmatrix} 1 & f_1(\mathbf{x}_1) & \ldots & f_K(\mathbf{x}_1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & f_1(\mathbf{x}_n) & \ldots & f_K(\mathbf{x}_n) \\ 1 & f_1(\mathbf{x}_{01}) & \ldots & f_K(\mathbf{x}_{01}) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & f_1(\mathbf{x}_{0m}) & \ldots & f_K(\mathbf{x}_{0m}) \end{bmatrix}_{(n+m)\times(K+1)}$$

and the symmetric positive-definite variance-covariance matrix is

$$\mathbf{V}_s(\hat{\boldsymbol{\theta}}) = \begin{bmatrix} \mathbf{V}(\hat{\boldsymbol{\theta}}) & \mathbf{V}_{01}(\hat{\boldsymbol{\theta}}) \\ \mathbf{V}_{01}(\hat{\boldsymbol{\theta}})^\top & \mathbf{V}_0(\hat{\boldsymbol{\theta}}) \end{bmatrix}_{(n+m)\times(n+m)}$$

wherein $\mathbf{V}_{01}(\hat{\boldsymbol{\theta}})$ is the $n \times m$ variance-covariance matrix between realizations at the observed locations and at the predictions sites and $\mathbf{V}_0(\hat{\boldsymbol{\theta}})$ is the $m \times m$ variance-covariance matrix for realizations at the m prediction sites $\mathbf{x}_{01}, \ldots, \mathbf{x}_{0m}$.

Table 2.3: Unconditional Bootstrap Estimation of the Kriging Variance; fixed test set.

1. For a given covariance function and kriging model, estimate the mean $\boldsymbol{\mu}$ and the spatial dependence parameters $\boldsymbol{\theta}$ using the observations $\mathbf{Z}$ to obtain $G(\mathbf{F}\hat{\boldsymbol{\beta}}, \mathbf{V}(\hat{\boldsymbol{\theta}}))$ the estimated distribution of $Z(\cdot)$.

2. Sample from $G(\mathbf{F}_s\hat{\boldsymbol{\beta}}, \mathbf{V}_s(\hat{\boldsymbol{\theta}}))$ at the observation locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and at the test locations $\mathbf{x}_{01}, \ldots, \mathbf{x}_{0m}$ simultaneously.

3. Use the bootstrap sample at the observation locations to fit a kriging model.

4. Use the fitted kriging model to predict at the m test locations $\mathbf{x}_{01}, \ldots, \mathbf{x}_{0m}$.

5. Calculate the squared prediction error at each test point.

6. Repeat steps 2 to 5 B times and then compute the MSPEs at each test point.

The samples $\mathbf{Z}^*$ are obtained without regard to the original observations $\mathbf{Z}$, that is sampling is carried out unconditionally. A simulation is said to be unconditional if samples at the observation locations do not correspond to the observed data $\mathbf{Z}$ (Schabenberger and Gotway, 2005).

Yin *et al* (2008; 2009) note that for stochastic simulations, which is what the bootstrap samples generated by this algorithm essentially are, the range parameter ϕ may be estimated poorly such that it approaches infinity, that is the likelihood estimates $\hat{\phi}_b^*$ may become much larger than the true ϕ. If a large proportion of the bootstrap samples have such estimates, this may bias the bootstrap estimate of the MSPE. Hence the estimates of the range from the bootstrap samples have to be bounded within a sensible range.

2.4.2.2 Fixed test set: Conditional simulation

A modification of the fixed test set algorithm of den Hertog *et al* (2006) is proposed here and summarized in Table 2.4. In this modified fixed test set algorithm, values at the test locations $\{\mathbf{x}_{01}, \ldots, \mathbf{x}_{0m}\}$ are sampled conditionally on the original observed data $\mathbf{Z}$ at the observation locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

Table 2.4: Conditional Bootstrap Estimation of the Kriging Variance; fixed test set.

1. For a given covariance function and kriging model, estimate the mean $\boldsymbol{\mu}$ and the spatial dependence parameters $\boldsymbol{\theta}$ using the observations $\mathbf{Z}$ to obtain $G(\mathbf{F}\hat{\boldsymbol{\beta}}, \mathbf{V}(\hat{\boldsymbol{\theta}}))$, the estimated distribution of $Z(\cdot)$.

2. Use the kriging model fitted from the observations $\mathbf{Z}$ to make predictions at the m test locations $\mathbf{x}_{01}, \ldots, \mathbf{x}_{0m}$.

3. Take a bootstrap sample at the test locations $\mathbf{x}_{01}, \ldots, \mathbf{x}_{0m}$ conditional on the observations $\mathbf{Z}$, that is sample from the conditional distribution $\mathbf{Z}_0|\mathbf{Z} \sim G\Big(\boldsymbol{\mu}_0 + \mathbf{V}_{01}(\hat{\boldsymbol{\theta}})^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}(\mathbf{Z} - \boldsymbol{\mu}), \mathbf{V}_0(\hat{\boldsymbol{\theta}}) - \mathbf{V}_{01}(\hat{\boldsymbol{\theta}})^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{V}_{01}(\hat{\boldsymbol{\theta}})\Big)$.

4. Calculate the squared differences of predictions and the realizations obtained from steps 2 and 3 respectively.

5. Repeat steps 3 to 4 B times and then compute the mean squared prediction errors at each test point.

Conditional sampling involves the multivariate Gaussian distribution (Schabenberger and Gotway, 2005; den Hertog *et al*, 2006)

$$\begin{bmatrix} \mathbf{Z} \\ \mathbf{Z}_0 \end{bmatrix} = G\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_0 \end{bmatrix} \begin{bmatrix} \mathbf{V}(\boldsymbol{\theta}) & \mathbf{V}_{01}(\boldsymbol{\theta}) \\ \mathbf{V}_{01}(\boldsymbol{\theta})^\top & \mathbf{V}_0(\boldsymbol{\theta}) \end{bmatrix}\right)$$

where $\mathbf{Z}_0$ are observations at the prediction sites with sampling being carried out from the conditional distribution $\mathbf{Z}_0$ given the observed data $\mathbf{Z}$, that is

$$\mathbf{Z}_0|\mathbf{Z} \sim G\Big(\boldsymbol{\mu}_0 + \mathbf{V}_{01}(\boldsymbol{\theta})^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}(\mathbf{Z} - \boldsymbol{\mu}), \mathbf{V}_0(\boldsymbol{\theta}) - \mathbf{V}_{01}(\boldsymbol{\theta})^\top \mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{V}_{01}(\boldsymbol{\theta})\Big).$$

Chapter 3

Simulation Experiment

This chapter describes a simulation study designed to test the performance of various estimators of the mean squared prediction error (MSPE) of the empirical best linear unbiased predictor (EBLUP), $m_2(\boldsymbol{\theta})$. The estimators in question are the traditional estimator of $m_2(\boldsymbol{\theta})$, namely the empirical mean squared prediction error (EMSPE) $m_1(\hat{\boldsymbol{\theta}})$, the Kacker-Harville and Prasad-Rao estimators which are denoted $\sigma^2_{KH}(\hat{\boldsymbol{\theta}})$ and $\sigma^2_{PR}(\hat{\boldsymbol{\theta}})$ respectively and two bootstrap estimators, the unconditional and conditional bootstrap estimators, $\sigma^2_{UBS}(\hat{\boldsymbol{\theta}})$ and $\sigma^2_{CBS}(\hat{\boldsymbol{\theta}})$ respectively.

The performance of the estimators was investigated at several prediction locations in Gaussian random fields exhibiting different degrees of spatial correlation. Such a set up for the experiment was employed in order to identify an estimator that is optimal in some sense, namely has the smallest relative bias, for a range of prediction problems. For this study only the exponential covariance structure, a widely used covariance model in geostatistics, was employed. The parametrization

$$C(h) = \sigma_0^2 \exp\left\{ -\frac{h}{\phi} \right\}, \tag{3.1}$$

was used as it leads to a *well-conditioned* minimization problem for parameter estimation (Etman, 1994, p5). Though the exponential covariance model has been criticized by Whittle (1954) as being too "artificial" it was felt that its study here was justifiable as it is widely used.

The Gaussian and spherical covariance models, two of the other widely used covariance models in geostatistics, are not invoked here. Reasons for not using the Gaussian covariance model

$$C(h) = \sigma_0^2 \exp\left\{ -\left(\frac{h}{\phi}\right)^2 \right\},$$

in this study are as follows:

- The Gaussian covariance model has been criticized as being too smooth to represent physical processes as it is infinitely differentiable (Stein and Handcock, 1989; Schabenberger and Gotway, 2005).

- Variance-covariance matrices $\mathbf{V}(\boldsymbol{\theta})$ associated with the Gaussian covariance function are plagued by problems of computational singularity. This is a formidable concern in studies involving a significant amount of computation such as the simulation studies performed here.

Reasons for not using the spherical covariance model

$$\begin{aligned} C(h) &= \sigma_0^2 \left\{ 1 - 1.5\frac{h}{\phi} + 0.5\left(\frac{h}{\phi}\right)^3 \right\}, \quad 0 \leq h \leq \phi, \\ &= 0 \quad \text{otherwise} \end{aligned}$$

are as follows:

- Correlations are zero beyond the range, that is $C(h) = 0$ for $h \geq \phi$. This property of the spherical model has been criticized by Stein (1999) who does not perceive any apparent statistical benefit emanating from having a distance at which the covariances are exactly zero.
- Maximum likelihood (ML) estimation is sub-optimal when used to estimate the parameters of the spherical covariance model (Todini and Ferraresi, 1996; Diggle and Riberio, 2007). This is because for "ML to be minimum variance requires the likelihood function to be continuous and twice differentiable with respect to the parameters over the entire field of existence" (Todini and Ferraresi, 1996). This condition does not apply to the spherical covariance model because of its discontinuity in the first derivative at $h = \phi$.

3.1 Setup

3.1.1 Domain

A square domain $D = [0, 15] \times [0, 15]$ was set up. A grid of 112 observation locations specified by the set

$$S = \{\mathbf{x}_i = (u, v) : u = 2k, v = 2l{+}1 \text{ and } u = 2l{+}1, v = 2k;\ k = 1, \ldots 7,\ l = 0, 1, \ldots 7\}$$

were then introduced into the domain. A regular sampling design for the observation locations $\mathbf{x}_1, \ldots, \mathbf{x}_{112}$ was chosen to ensure even coverage of the study domain.

A set of five points $\{\mathbf{x}_{01}, \ldots, \mathbf{x}_{05}\} = \{(3, 1), (13, 0), (8, 8), (3, 13), (15, 15)\}$ was also introduced into the study domain D as prediction locations. The $m = 5$ prediction locations were located so as to enable the effective study of location and data configuration on the various $m_2(\boldsymbol{\theta})$ estimators. Points $\mathbf{x}_{01}, \mathbf{x}_{02}$ and $\mathbf{x}_{05}$ were located such that they had different data-to-point variance-covariance vectors $\mathbf{v}(\boldsymbol{\theta})$, whilst $\mathbf{x}_{03}$ and $\mathbf{x}_{04}$ had similar $\mathbf{v}(\boldsymbol{\theta})$ vectors. The study domain D with the configuration of the $n = 112$ observation locations and $m = 5$ prediction locations is shown in Figure 3.1.

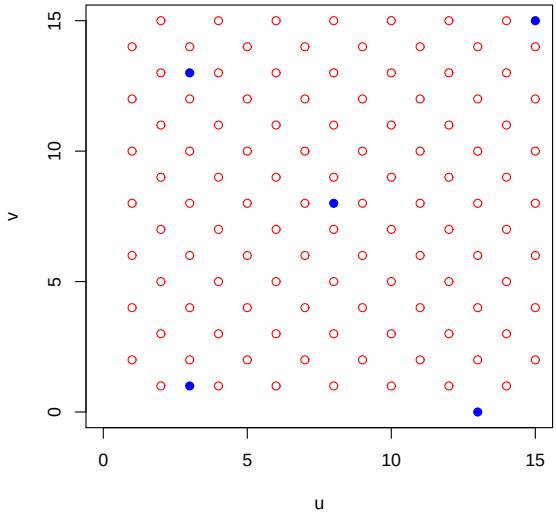


Figure 3.1: Spatial configuration of the $n = 112$ observation locations and $m = 5$ test points. Observation locations are indicated by open circles and test points by solid circles.

3.1.2 Model settings

The performance of the various kriging variance estimators was studied for three Gaussian random fields with the form

$$\mathbf{Z} = \mathbf{1}\beta + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \sim G(\mathbf{0}, \mathbf{V}(\boldsymbol{\theta}))$$

and different specifications for the parameter vector $\boldsymbol{\theta} = (\tau^2, \sigma_0^2, \phi)$. The only variation in the parameter vectors for the fields under consideration was in the values of the range ϕ since various authors, such as Zimmerman and Cressie (1992) and Abt (1999) have found the strength of the spatial correlation to be one of the key factors that influences the performance of kriging variance estimators. The same values for the partial sill σ_0^2 and nugget τ^2 values were used as changing these parameters has been found to only lead to rescaling of the kriging variance estimates (Abt, 1999).

The mean for each of the fields was specified as $\beta = 0$ and partial sill and nugget values of $\sigma_0^2 = 1$ and $\tau^2 = 0.25$ respectively were used. The nugget value $\tau^2 = 0.25$ was chosen such that it was 20% of the total variance, that is the sill $\sigma^2 = \tau^2 + \sigma_0^2$. In passing it is noted that Zimmerman (2006) used τ^2 values that were 25% and 50% of σ^2 in simulation studies on optimal designs involving two of our estimators of interest, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$, the Kacker-Harville estimator and $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$, the unconditional bootstrap estimator. Range values of $\phi = 1.5$, $\phi = 2.5$ and $\phi = 4.2$ were selected to be used in the experiment. These values were intended to correspond to weak, moderate and strong spatial correlations respectively. The choice of $\phi = 1.5$ as the smallest range value was made to avoid computational problems in the estimation of the parameters, such as near-singularity in the variance-covariance matrices. The

three random fields $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$ studied with their attendant parameter vectors $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_3$ are summarized in Table 3.1.

Table 3.1: Specifications of the three Gaussian random fields $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$ used in the simulation.

Field	β	τ^2	σ_0^2	ϕ	Correlation Description
$\mathbf{Z}_1$	0	0.25	1	1.5	Weak
$\mathbf{Z}_2$	0	0.25	1	2.5	Moderate
$\mathbf{Z}_3$	0	0.25	1	4.2	Strong

3.2 Approach

The simulation experiment was carried out in two parts. One part of the simulation was done in order to estimate through stochastic simulation, that is parametric bootstrapping, the expected values at the prediction locations for each of the estimators $m_2(\boldsymbol{\theta})$ under investigation, that is the EMSPE

$$\begin{aligned} m_1(\hat{\boldsymbol{\theta}}) &= \hat{\sigma}^2 - \mathbf{v}(\hat{\boldsymbol{\theta}})^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{v}(\hat{\boldsymbol{\theta}}) + (\mathbf{f}_0^\top - \mathbf{v}(\hat{\boldsymbol{\theta}})^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{F}) \\ &\quad \times (\mathbf{F}^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{F})^{-1}(\mathbf{f}_0 - \mathbf{F}^\top \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{v}(\hat{\boldsymbol{\theta}})), \end{aligned}$$

the Kacker-Harville estimator

$$\sigma^2(\hat{\boldsymbol{\theta}})_{KH} = m_1(\hat{\boldsymbol{\theta}}) + \mathrm{tr}\{\mathbf{A}(\hat{\boldsymbol{\theta}})\mathbf{I}(\hat{\boldsymbol{\theta}})^{-1}\},$$

the Prasad-Rao estimator

$$\sigma^2(\hat{\boldsymbol{\theta}})_{PR} = m_1(\hat{\boldsymbol{\theta}}) + 2\mathrm{tr}\{\mathbf{A}(\hat{\boldsymbol{\theta}})\mathbf{I}(\hat{\boldsymbol{\theta}})^{-1}\},$$

the unconditional bootstrap estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ which is calculated using the algorithm in Table 2.3 and the conditional bootstrap estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ which is calculated using the algorithm given in Table 2.4. The second part of the simulation experiment involved numerical simulation in order to obtain a Monte Carlo (MC) estimate for $m_2(\boldsymbol{\theta})$ the true MSPE of the EBLUP at each of the prediction locations.

All simulations, that is realizations from the Gaussian random fields $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$, were generated by multiplying a vector of standardized normal errors by a square root of the variance-covariance matrix $\mathbf{V}(\boldsymbol{\theta})$ and adding the result to a mean function. This method is well known and its description can be found in most texts on geostatistics, such as in Cressie (1993), Chilès and Delfiner (1999) and Schabenberger and Gotway (2005).

Since the square root of a matrix is not uniquely defined it may be found using various methods. One such method is through the Cholesky decomposition of $\mathbf{V}(\boldsymbol{\theta})$, which was employed for the simulations here. In the Cholesky method of simulating Gaussian spatial processes, a realization of a random field is obtained from the model

$$\mathbf{Z} = \mathbf{F}\boldsymbol{\beta} + \mathbf{L}\boldsymbol{\varepsilon}^* \tag{3.2}$$

where $\boldsymbol{\varepsilon}^*$ is the vector of standard normal random errors $\boldsymbol{\varepsilon}^* \sim N(\mathbf{0}, \mathbf{I}_n)$ and $\mathbf{L}$, the square root of $\mathbf{V}(\boldsymbol{\theta})$, is a lower triangular matrix obtained from the Cholesky decomposition of $\mathbf{V}(\boldsymbol{\theta})$ so that $\text{Var}\{\mathbf{Z}\} = \mathbf{L}\mathbf{L}^\top = \mathbf{V}(\boldsymbol{\theta})$.

The entire simulation experiment was executed in R (R Development Core Team, 2010) with the aid of functions from the geostatistics package **geoR** (Riberio and Diggle, 2001). The generation of Gaussian random fields for given covariance parameters was carried out by use of the **geoR** function `grf()`. Purpose built functions were written to compute the various kriging variance estimators, namely the bias corrected estimators $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ and the bootstrap estimators $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$. Codes for each of the functions for calculating the above mentioned estimators are found in Appendix A and their attendant documentation in Appendix B.

3.2.1 Parametric bootstrap procedures for the estimators of the kriging variance

For each random field $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$, 1000 realizations were generated at the observation locations and for each of the realizations kriging models were fitted by estimating $\boldsymbol{\theta}$ as both $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$. For each realization, the kriging variances $m_2(\boldsymbol{\theta})$ at the $m = 5$ prediction locations were then estimated via each of the 5 kriging variance estimators under study. The bootstrap estimators $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ were calculated over $B = 1000$ iterations for each of the 1000 realizations. At each prediction location the estimates of $m_2(\boldsymbol{\theta})$ from each of the five estimators were then averaged over the 1000 realizations of each field. These values are reported in Tables 3.4 and 3.5 which are given towards the end of the chapter.

Because 3 parameter vectors, 5 kriging variance estimators, 2 parameter estimation methods and 5 test points were used in the numerical experiment, the experiment was of order $3 \times 5 \times 2 \times 5 = 150$. Computing the bootstrap estimates, namely unconditional bootstrap estimates $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ over $B = 1000$ iterations for each of the 1000 realizations of the numerical experiments and also $\hat{\boldsymbol{\theta}}_{ML}$ plus $\hat{\boldsymbol{\theta}}_{REML}$ meant the experiment was computationally expensive as $1000 \times 1000 \times 2 = 2000000$ kriging models had to be fitted. This consideration limited the number of realizations that could be used for each numerical experiment to only 1000 instead of the 5000 executed by Zimmerman and Cressie (1992, Section 4.3) in a similar but less comprehensive study. Other key differences between this study and that of Zimmerman and Cressie (1992) are as follows.

1. They use a 1-dimensional example whilst we use a 2-dimensional example.

2. They introduced a generalized covariance function which is not often used in practice whilst we use the widely used exponential covariance function.

3. They only apply restricted maximum likelihood (REML) estimators of the parameters $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}_{REML}$, whilst we apply maximum likelihood estimators (MLE) $\hat{\boldsymbol{\theta}}_{ML}$ in addition to $\hat{\boldsymbol{\theta}}_{REML}$.

4. Their study only features 2 prediction points whilst this study features 5 prediction points.

5. They study the performance of 3 estimators whilst in this study we study the performance of their 3 estimators, the EMSPE $m_1(\hat{\boldsymbol{\theta}})$, the Kacker-Harville estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and the Prasad-Rao estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ and two additional ones, the unconditional bootstrap estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ and the conditional bootstrap estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$.

Lastly, as will be seen in the next section, and unlike in Zimmerman and Cressie (1992), simulations of the order of 60000 and not 5000 were used in the approximation of the MSPE of the BLUP, $m_2(\boldsymbol{\theta})$. This was done in order to minimize the sampling bias in the estimates of $m_2(\boldsymbol{\theta})$ which is our "gold standard", that is the values against which the various kriging variance estimators were to be compared to.

3.2.2 Monte Carlo approximation of $m_2(\boldsymbol{\theta})$

3.2.2.1 Algorithm

Since no analytical expression for

$$m_2(\boldsymbol{\theta}) = \mathrm{E}\{[\hat{Z}(\mathbf{x}_0;\hat{\boldsymbol{\theta}}) - Z(\mathbf{x}_0)]^2\}$$

where $\hat{Z}(\mathbf{x}_0;\hat{\boldsymbol{\theta}})$ is the EBLUP and $Z(\mathbf{x}_0)$ is the unknown value, is available, it was approximated by Monte Carlo simulation. For each Gaussian random field $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$, realizations were generated over the entire grid of observation and prediction locations in D. For each realization the variance parameters $\boldsymbol{\theta}$ along with the mean β were estimated via ML and REML estimation using the data at the observation locations. Next for each realization the EBLUP $\hat{Z}(\mathbf{x}_0;\hat{\boldsymbol{\theta}})$ for each prediction location in $\mathbf{x}_{01},\ldots,\mathbf{x}_{05}$ was computed. Differences between the the predicted values $\hat{Z}_2(\mathbf{x}_0)$ and the actual values $Z(\mathbf{x}_0)$ at the prediction locations were then computed. For each random field, these values were averaged over the realizations of that field to give numerical approximations to $m_2(\boldsymbol{\theta}) = \mathrm{E}\{[\hat{Z}(\mathbf{x}_0;\hat{\boldsymbol{\theta}}) - Z(\mathbf{x}_0)]^2\}$ at each prediction location $\mathbf{x}_{01},\ldots,\mathbf{x}_{05}$. The algorithm is outlined in Table 3.2.

Table 3.2: Algorithm used to approximate $m_2(\boldsymbol{\theta})$, the MSPE of the EBLUP, at the prediction locations for each of the Gaussian random fields $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$ used in the simulation experiment.

1. Set $r = 1$.

2. Generate a spatial process over the grid of observation and prediction locations.

3. For the r^{th} realization fit a kriging model by using the data at the observation locations to obtain estimates $\hat{\boldsymbol{\theta}}_r$ of $\boldsymbol{\theta}$ and $\hat{\beta}_r$ of β.

4. Calculate $[\hat{Z}(\mathbf{x}_0; \hat{\boldsymbol{\theta}}_r) - Z_r(\mathbf{x}_0)]^2$ the squared difference between $\hat{Z}(\mathbf{x}_0; \hat{\boldsymbol{\theta}}_r)$ the predicted value and $Z_r(\mathbf{x}_0)$ the observed value at $\mathbf{x}_0$.

5. Repeat steps 2 to 4 updating r to $r + 1$, R times and then compute the average squared differences between the predicted and simulated values at each prediction location $\mathbf{x}_0$ as

$$\frac{1}{R} \sum_{r=1}^{R} [\hat{Z}(\mathbf{x}_0; \hat{\boldsymbol{\theta}}_r) - Z_r(\mathbf{x}_0)]^2.$$

3.2.2.2 Pilot study

A pilot study was run in order to determine the number of iterations of the algorithm in Table 3.2 needed to approximate $m_2(\boldsymbol{\theta})$ with an accuracy comparable to the (mean) standard errors obtained for the 5 estimators in the stochastic simulation described in Section 3.2.1, that is to an accuracy of $d = \pm 0.004$. For each of the Gaussian random fields, test runs of the algorithm in Table 3.2 were carried out with 1000 iterations and the sample standard deviations s at each point were noted. These are given in Table 3.3 along with the average sample standard deviations $\bar{s}$ for each of the fields.

In order to determine the approximate sample size, that is the number of iterations of the algorithm needed to approximate $m_2(\boldsymbol{\theta})$ with an accuracy of $d \pm 0.004$ units, the following procedure was followed. Firstly the overall mean value for the sample standard deviations $\bar{\bar{s}}$ was calculated as 1.0178. Taking the standard error to be $d = s/\sqrt{n}$, it can be seen that the approximate sample size needed to estimate $m_2(\boldsymbol{\theta})$ with an average value of d for the standard error is given by $n = (s/d)^2$. Using this formula with $s = 1.0178$ and $d = 0.004$, we obtain n=64745. Thus 60000 iterations of the algorithm will produce estimates with a mean standard error of about ± 0.004.

The approximate $m_2(\boldsymbol{\theta})$ values at each prediction point calculated over 60000 iterations of the algorithm along with their standard errors obtained are reported in Tables 3.4 and 3.5 where $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$ are the ML and REML estimators of $\boldsymbol{\theta}$

Table 3.3: Sample standard deviations of the squared prediction error estimates at the prediction locations under ML and REML obtained from the pilot study for the various Gaussian random fields.

FIELD / Estimator / $\mathbf{x}_0$	$\mathbf{Z}_1$ $\hat{\boldsymbol{\theta}}_{ML}$	$\mathbf{Z}_1$ $\hat{\boldsymbol{\theta}}_{REML}$	$\mathbf{Z}_2$ $\hat{\boldsymbol{\theta}}_{ML}$	$\mathbf{Z}_2$ $\hat{\boldsymbol{\theta}}_{REML}$	$\mathbf{Z}_3$ $\hat{\boldsymbol{\theta}}_{ML}$	$\mathbf{Z}_3$ $\hat{\boldsymbol{\theta}}_{REML}$
(3,1)	1.1788	1.1752	0.9296	0.9365	0.7563	0.7601
(13,0)	1.6154	1.6374	1.3434	1.3558	1.0790	1.0863
(8,8)	1.0073	1.0134	0.7847	0.7876	0.6319	0.6295
(3,13)	1.0837	1.0918	0.8419	0.8454	0.6834	0.6832
(15,15)	1.2899	1.2981	1.0469	1.0506	0.8521	0.8538
Mean Sample Standard deviation ($\bar{s}$)	1.2350	1.2342	0.9893	1.0453	0.8006	0.8026

respectively.

3.3 Results

3.3.1 Presentation of results

Tables 3.4 and 3.5 report at each point across each of the random fields in the simulation, the values the MSPE of the BLUP, $m_1(\boldsymbol{\theta})$, the MSPE of the EBLUP, $m_2(\boldsymbol{\theta})$ and the EMSPE $m_1(\hat{\boldsymbol{\theta}})$ together with the values of the 4 alternate kriging variance estimators, namely $m_1(\hat{\boldsymbol{\theta}})$, $\sigma_0^2(\hat{\boldsymbol{\theta}})_{KH}$, $\sigma_0^2(\hat{\boldsymbol{\theta}})_{PR}$, $\sigma_0^2(\hat{\boldsymbol{\theta}})_{CBS}$ and $\sigma_0^2(\hat{\boldsymbol{\theta}})_{UBS}$, under both ML and REML estimation of the covariance parameters. Table 3.6 reports the relative bias values

$$\frac{\mathrm{E}\{\sigma_E^2(\hat{\boldsymbol{\theta}})\} - m_2(\boldsymbol{\theta})}{m_2(\boldsymbol{\theta})}, \tag{3.3}$$

where $\sigma_E^2(\hat{\boldsymbol{\theta}})$ is any generic estimator of $m_2(\boldsymbol{\theta})$, of the estimators also at each point across all the 3 random fields, $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$. The relative bias of the estimators is also represented diagrammatically in Figure 3.2 for each of the fields and kriging variance estimators under both estimators of $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$. Figures 3.3 and 3.4 display the absolute relative bias in percentages of the estimators, under $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$ respectively.

Table 3.4: Bootstrap estimates of $m_2(\boldsymbol{\theta})$ of the 5 kriging variance estimators at 5 test locations across the Gaussian random fields $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$ with $\hat{\boldsymbol{\theta}}_{ML}$ as the estimator of $\boldsymbol{\theta}$. Also shown are the calculated values of $m_1(\boldsymbol{\theta})$ and the MC estimates of $m_2(\boldsymbol{\theta})$. In parentheses are the standard errors of the estimates.

Random field	ϕ	$\mathbf{x}_0$	$m_1(\boldsymbol{\theta})$	$m_2(\boldsymbol{\theta})$	$m_1(\hat{\boldsymbol{\theta}})$	$\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$
		(3,1)	0.843	0.862 (0.0050)	0.836 (0.0084)	0.879 (0.0030)	0.922 (0.0053)	0.853 (0.0088)	0.837 (0.0087)
		(13,0)	1.054	1.067 (0.0062)	1.034 (0.0084)	1.049 (0.0090)	1.063 (0.0087)	1.050 (0.0090)	1.035 (0.0089)
$\mathbf{Z}_1$	1.5	(8,8)	0.789	0.806 (0.0047)	0.785 (0.0085)	0.837 (0.0375)	0.889 (0.0684)	0.801 (0.0091)	0.786 (0.0089)
		(3,13)	0.789	0.795 (0.0046)	0.786 (0.0086)	0.837 (0.0375)	0.888 (0.0684)	0.801 (0.0090)	0.786 (0.0090)
		(15,15)	0.934	0.941 (0.0054)	0.919 (0.0081)	0.954 (0.0219)	0.988 (0.0376)	0.938 (0.0090)	0.920 (0.0086)
		(3,1)	0.662	0.676 (0.0039)	0.643 (0.0050)	0.655 (0.0052)	0.668 (0.0054)	0.652 (0.0052)	0.643 (0.0051)
		(13,0)	0.868	0.880 (0.0051)	0.845 (0.0051)	0.858 (0.0052)	0.870 (0.0053)	0.857 (0.0055)	0.844 (0.0054)
$\mathbf{Z}_2$	2.5	(8,8)	0.613	0.625 (0.0036)	0.595 (0.0049)	0.607 (0.0052)	0.620 (0.0055)	0.605 (0.0052)	0.596 (0.0050)
		(3,13)	0.613	0.617 (0.0036)	0.595 (0.0049)	0.607 (0.0052)	0.620 (0.0055)	0.604 (0.0052)	0.596 (0.0051)
		(15,15)	0.758	0.765 (0.0044)	0.731 (0.0050)	0.745 (0.0051)	0.759 (0.0053)	0.743 (0.0053)	0.733 (0.0053)
		(3,1)	0.532	0.544 (0.0032)	0.509 (0.0040)	0.518 (0.0041)	0.528 (0.0043)	0.516 (0.0042)	0.509 (0.0041)
		(13,0)	0.694	0.708 (0.0041)	0.674 (0.0041)	0.685 (0.0042)	0.696 (0.0043)	0.685 (0.0044)	0.674 (0.0044)
$\mathbf{Z}_3$	4.2	(8,8)	0.492	0.502 (0.0029)	0.470 (0.0039)	0.479 (0.0041)	0.488 (0.0042)	0.478 (0.0042)	0.470 (0.0040)
		(3,13)	0.492	0.495 (0.0029)	0.470 (0.0039)	0.479 (0.0041)	0.488 (0.0042)	0.477(0.0041)	0.470 (0.0041)
		(15,15)	0.616	0.623 (0.0036)	0.586 (0.0040)	0.597 (0.0041)	0.608 (0.0043)	0.596 (0.0043)	0.587 (0.0043)

Table 3.5: Bootstrap estimates of $m_2(\boldsymbol{\theta})$ of the 5 kriging variance estimators at 5 test locations across the Gaussian random fields $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$ with $\hat{\boldsymbol{\theta}}_{REML}$ as the estimator of $\boldsymbol{\theta}$. Also shown are the calculated values of $m_1(\boldsymbol{\theta})$ and the MC estimates of $m_2(\boldsymbol{\theta})$. In parentheses are the standard errors of the estimates.

Random field	ϕ	$\mathbf{x}_0$	$m_1(\boldsymbol{\theta})$	$m_2(\boldsymbol{\theta})$	$m_1(\hat{\boldsymbol{\theta}})$	$\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$
		(3,1)	0.843	0.864 (0.0050)	0.830 (0.0736)	0.849 (0.0768)	0.867 (0.0805)	0.846 (0.0766)	0.831 (0.0755)
		(13,0)	1.054	1.070 (0.0062)	1.030 (0.0808)	1.045 (0.0819)	1.060 (0.0831)	1.049 (0.0833)	1.032 (0.0830)
$\mathbf{Z}_1$	1.5	(8,8)	0.789	0.809 (0.0047)	0.777 (0.0718)	0.797 (0.0758)	0.817 (0.0806)	0.794 (0.0757)	0.777 (0.0728)
		(3,13)	0.789	0.798 (0.0046)	0.778 (0.0720)	0.797 (0.0759)	0.816 (0.0805)	0.791 (0.0748)	0.777 (0.0729)
		(15,15)	0.934	0.945 (0.0055)	0.918 (0.0758)	0.936 (0.0782)	0.955 (0.0809)	0.937 (0.0797)	0.918 (0.0773)
		(3,1)	0.662	0.678 (0.0040)	0.648 (0.0072)	0.659 (0.0074)	0.670 (0.0076)	0.659 (0.0076)	0.647 (0.0075)
		(13,0)	0.868	0.883 (0.0051)	0.852 (0.0076)	0.865 (0.0077)	0.878 (0.0078)	0.868 (0.0081)	0.853 (0.0080)
$\mathbf{Z}_2$	2.5	(8,8)	0.613	0.627 (0.0036)	0.599 (0.0071)	0.610 (0.0073)	0.621 (0.0077)	0.607 (0.0073)	0.599 (0.0073)
		(3,13)	0.613	0.618 (0.0036)	0.599 (0.0071)	0.610 (0.0073)	0.621 (0.0075)	0.609 (0.0074)	0.598 (0.0072)
		(15,15)	0.758	0.767 (0.0044)	0.741 (0.0073)	0.755 (0.0075)	0.768 (0.0077)	0.755 (0.0079)	0.740 (0.0076)
		(3,1)	0.532	0.545 (0.0032)	0.517 (0.0041)	0.526 (0.0042)	0.534 (0.0043)	0.525 (0.0043)	0.518 (0.0042)
		(13,0)	0.694	0.708 (0.0041)	0.681 (0.0042)	0.692 (0.0042)	0.702 (0.0043)	0.693 (0.0045)	0.681 (0.0044)
$\mathbf{Z}_3$	4.2	(8,8)	0.492	0.503 (0.0029)	0.477 (0.0040)	0.485 (0.0041)	0.493 (0.0042)	0.486 (0.0042)	0.478 (0.0041)
		(3,13)	0.492	0.496 (0.0029)	0.478 (0.0040)	0.485 (0.0041)	0.493 (0.0042)	0.485 (0.0042)	0.478 (0.0042)
		(15,15)	0.616	0.624 (0.0036)	0.598 (0.0042)	0.609 (0.0043)	0.620 (0.0044)	0.610 (0.0044)	0.600 (0.0044)

Table 3.6: Approximate relative bias of the 5 kriging variance estimators at each point under ML and REML estimation of the covariance parameters across the Gaussian random fields $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$. Under each estimation method of $\boldsymbol{\theta}$ the best estimator and random field, the estimator with the smallest relative bias at each point is indicated in bold.

			$\hat{\boldsymbol{\theta}}_{ML}$					$\hat{\boldsymbol{\theta}}_{REML}$				
Random field	ϕ	$\mathbf{x}_0$	$m_1(\hat{\boldsymbol{\theta}})$	$\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$	$m_1(\hat{\boldsymbol{\theta}})$	$\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$	$\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$
		(3,1)	-0.030	+0.020	+0.070	**-0.010**	-0.029	-0.039	-0.017	**+0.003**	-0.021	-0.038
		(13,0)	-0.031	-0.017	**+0.004**	-0.016	-0.030	-0.037	0.023	**-0.009**	-0.020	-0.036
$\mathbf{Z}_1$	1.5	(8,8)	-0.026	+0.038	+0.103	**-0.006**	-0.025	-0.040	0.015	**+0.010**	-0.019	-0.040
		(3,13)	-0.011	+0.053	+0.117	**+0.008**	-0.011	-0.025	**+0.001**	0.023	-0.009	-0.026
		(15,15)	-0.023	+0.014	+0.050	**+0.003**	-0.022	-0.029	-0.010	+0.011	**-0.008**	-0.029
		(3,1)	- 0.049	-0.031	**-0.012**	-0.036	-0.049	-0.044	-0.028	**-0.012**	-0.028	-0.046
		(13,0)	-0.040	-0.025	**-0.011**	-0.026	-0.041	-0.035	-0.031	**-0.006**	-0.017	-0.034
$\mathbf{Z}_2$	2.5	(8,8)	-0.048	-0.029	**-0.008**	-0.032	-0.046	-0.045	-0.027	**-0.010**	-0.032	-0.045
		(3,13)	-0.036	-0.016	**+0.005**	-0.021	-0.034	-0.031	-0.013	**+0.005**	-0.015	-0.032
		(15,15)	-0.044	-0.026	**0.008**	-0.029	-0.042	-0.034	-0.016	**+0.001**	-0.016	-0.035
		(3,1)	-0.064	-0.048	**-0.029**	-0.051	-0.064	-0.051	-0.035	**-0.020**	-0.037	-0.050
		(13,0)	-0.048	-0.032	**-0.017**	-0.032	-0.048	-0.038	-0.023	**-0.008**	-0.021	-0.038
$\mathbf{Z}_3$	4.2	(8,8)	-0.064	-0.046	**-0.028**	-0.048	-0.064	-0.052	-0.036	**-0.020**	-0.034	-0.050
		(3,13)	-0.051	-0.032	**-0.014**	-0.036	-0.051	-0.036	-0.022	**-0.006**	-0.022	-0.036
		(15,15)	-0.059	-0.042	**-0.024**	-0.043	-0.058	-0.042	-0.024	**-0.006**	-0.022	-0.038

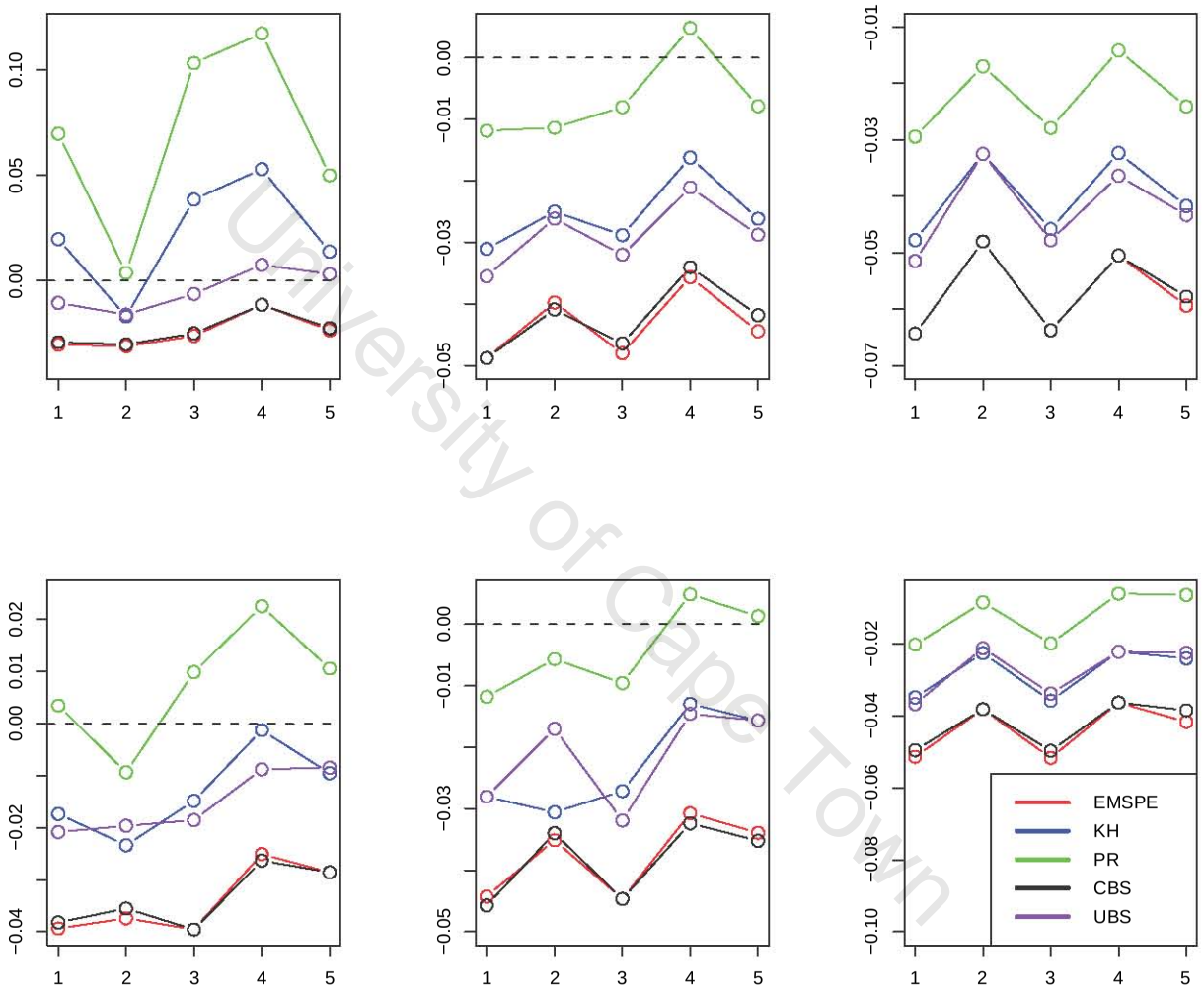


Figure 3.2: Graphical comparisons of the relative bias of the 5 estimators of $m_2(\boldsymbol{\theta})$ at the 5 prediction locations in the three fields, $\mathbf{Z}_1$ (left panels), $\mathbf{Z}_2$ (middle panels) and $\mathbf{Z}_3$ (right panels). Parameters estimated via $\hat{\boldsymbol{\theta}}_{ML}$ (first row) and $\hat{\boldsymbol{\theta}}_{REML}$ (second row). Relative bias plotted on the same scale across all three panels.

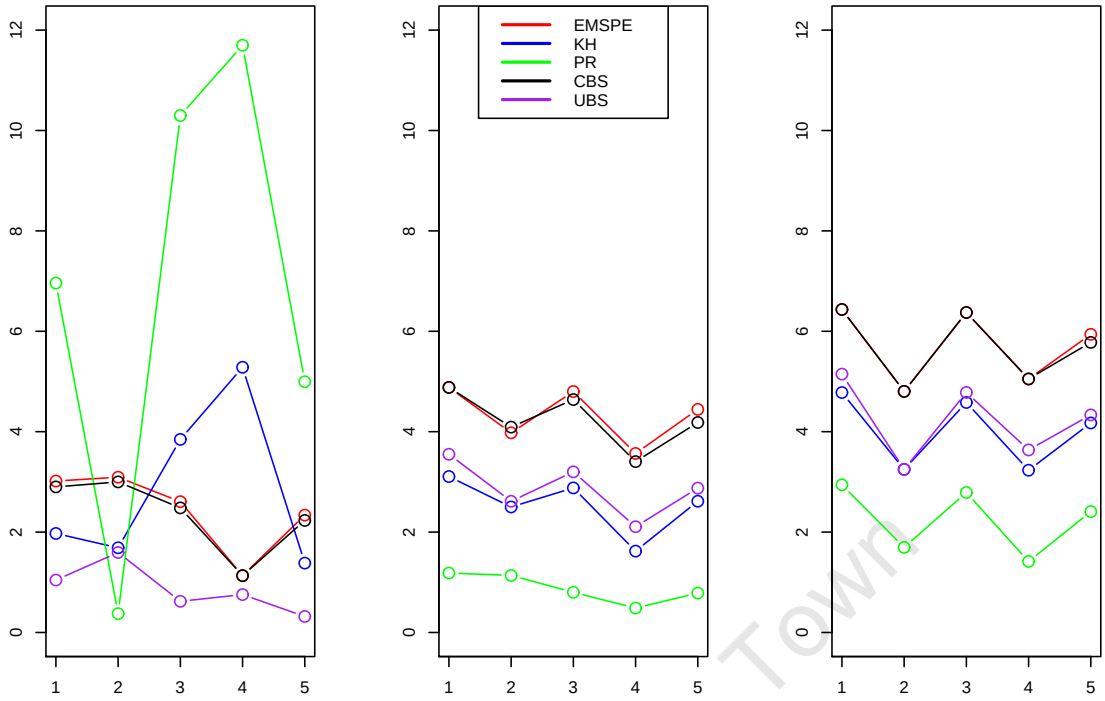


Figure 3.3: Graphical comparisons of the absolute relative bias (in percentages) of the 5 estimators of $m_2(\boldsymbol{\theta})$ at the 5 prediction locations in the three fields, $\mathbf{Z}_1$ (left panel), $\mathbf{Z}_2$ (middle panel) and $\mathbf{Z}_3$ (right panel). Parameters are estimated via ML estimation.

3.3.2 Findings

From Tables 3.4 and 3.5 it can be clearly seen that $m_1(\boldsymbol{\theta})$ is smaller than $m_2(\boldsymbol{\theta})$ whilst $m_1(\hat{\boldsymbol{\theta}})$ is smaller than both of these. Thus the inequality of Theorem 3.1 given by Zimmerman and Cressie (1992, Section 3.1)

$$m_1(\hat{\boldsymbol{\theta}}) \leq m_1(\boldsymbol{\theta}) \leq m_2(\boldsymbol{\theta})$$

is demonstrated. This is despite the conditions of the Theorem not being satisfied here, namely that of the covariance function $C(h)$ being linear in the elements of $\boldsymbol{\theta}$.

The EMSPE $m_1(\hat{\boldsymbol{\theta}})$ underestimates $m_2(\boldsymbol{\theta})$ by 1.1-6.4% for $\hat{\boldsymbol{\theta}}_{ML}$ and by 2.5-5.2% for $\hat{\boldsymbol{\theta}}_{REML}$, across the 3 Gaussian random fields, with mean absolute relative bias values of 4.16% and 3.85% respectively. The degree of underestimation of $m_2(\boldsymbol{\theta})$ seen in $m_1(\hat{\boldsymbol{\theta}})$ increases with an increase in the spatial correlation exhibited by the random fields, such that it is largest when the spatial correlation is strong. This effect is seen when both $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$ are used as estimators of $\boldsymbol{\theta}$.

The performance of the conditional bootstrap estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ is similar to that of $m_1(\hat{\boldsymbol{\theta}})$ at each point across all the random fields in the simulation and under both estimators $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$. In fact it can be seen from Figures 3.3 and 3.4 that the pattern of the absolute relative bias of these two estimators is almost

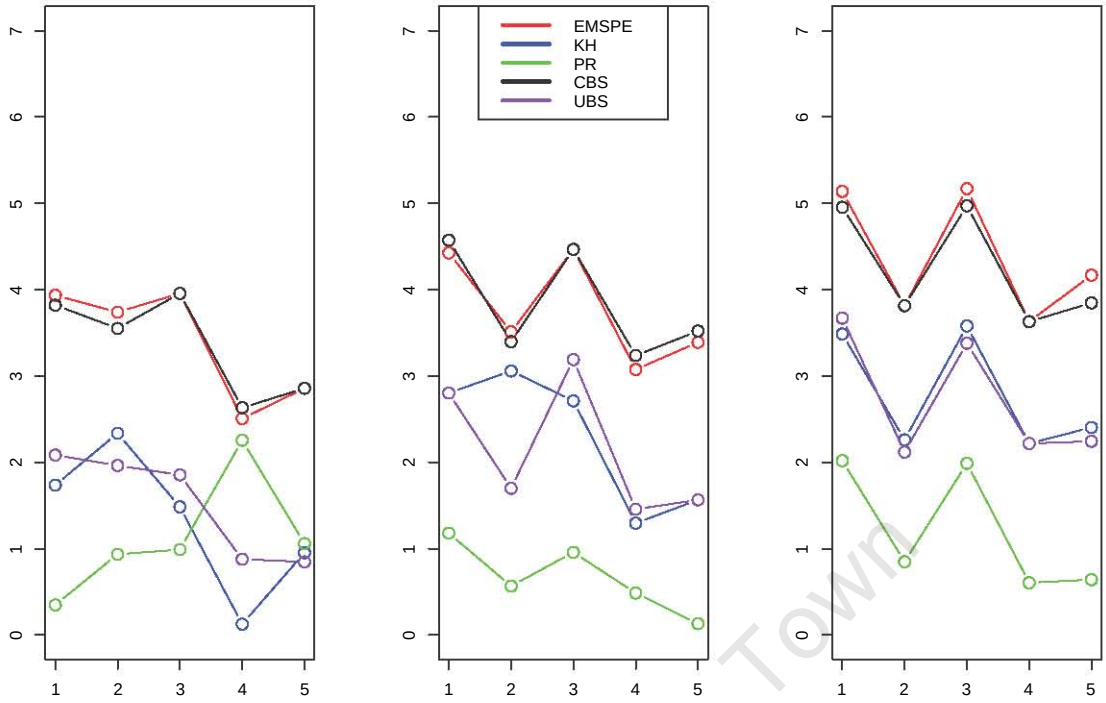


Figure 3.4: Graphical comparisons of the absolute relative bias (in percentages) of the 5 estimators of $m_2(\boldsymbol{\theta})$ at the 5 prediction locations in the three fields, $\mathbf{Z}_1$ (left panel), $\mathbf{Z}_2$ (middle panel) and $\mathbf{Z}_3$ (right panel). Parameters are estimated via REML estimation.

indistinguishable as is the degree to which they underestimate $m_2(\boldsymbol{\theta})$, see Figure 3.2.

The Kacker-Harville $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and unconditional bootstrap $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ estimators also exhibit very similar behaviour, but only when the spatial correlation is moderate to strong. As can be seen from Tables 3.4 to 3.6 and Figure 3.2, both estimators tend to underestimate $m_2(\boldsymbol{\theta})$ by the same degree under moderate to strong spatial correlation. When the correlation is weak $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ has a propensity to lead to overestimates of $m_2(\boldsymbol{\theta})$ when calculated with $\hat{\boldsymbol{\theta}}_{ML}$ whilst $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ exhibits a high degree of accuracy with an absolute relative bias as small as 0.3% at one point. With $\hat{\boldsymbol{\theta}}_{REML}$ the estimators are more evenly matched when the spatial correlation is weak. We note that in simulation studies on optimal designs in spatial statistics, that is optimal network designs, Zimmerman (2006) found that design criteria based on $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ lead to similar designs. In particular he found that designs that have small values for

$$\max_{\mathbf{s}\in S}\{\sigma^2(\hat{\boldsymbol{\theta}})_{KH}\}$$

coincide with designs that minimize

$$\max_{\mathbf{s}\in S}\{\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}\}$$

where S is the set of all possible sites and $\mathbf{s}$ are the sites that form the design, that is the set of observation locations.

From the simulation study the Prasad-Rao estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ does not seem to exhibit behaviour that is very close to any of the other four kriging variance estimators studied here. The estimator exhibits a large positive bias of up to 11.7% with $\hat{\boldsymbol{\theta}}_{ML}$ and 2.3% with $\hat{\boldsymbol{\theta}}_{REML}$ when the spatial correlation is weak. This is to be expected as it was noted earlier in Section 2.4.1, that when a random field exhibits very weak correlation the large sample covariance matrix which is given by the inverse of the information matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta}) \simeq \mathbf{I}(\boldsymbol{\theta})^{-1}$, overestimates the variability of the parameter vector $\boldsymbol{\theta}$ such that the second term in the expression

$$\sigma^2(\hat{\boldsymbol{\theta}})_{PR} \;=\; m_1(\hat{\boldsymbol{\theta}}) + 2\text{tr}\{\mathbf{A}(\hat{\boldsymbol{\theta}})\mathbf{I}(\hat{\boldsymbol{\theta}})^{-1}\}$$

is very large (Abt, 1998). Thus, as observed here, this leads to $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ overestimating $m_2(\boldsymbol{\theta})$. The same explanation can be given as to why $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ has a propensity to be positively biased when the spatial correlation is weak.

The Prasad-Rao estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ however appears to be the most optimal estimator when spatial correlation is moderate to strong as it exhibits the smallest absolute relative bias at all points when compared to the other 4 estimators of $m_2(\boldsymbol{\theta})$. The estimator performs well with both $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$ in these situations, with a bias as small as 0.5% and 0.1% under the respective estimators of $\boldsymbol{\theta}$ is some situations. The estimator exhibits mean absolute relative bias value of 3.33% with $\hat{\boldsymbol{\theta}}_{ML}$ and 1% with $\hat{\boldsymbol{\theta}}_{REML}$. This makes $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ unequivocally a superior alternative estimator to $m_1(\hat{\boldsymbol{\theta}})$ as it leads to reductions in the mean absolute relative bias of $m_1(\hat{\boldsymbol{\theta}})$ of 19.95% and 73.96% for $\hat{\boldsymbol{\theta}}_{ML}$and with $\hat{\boldsymbol{\theta}}_{REML}$ respectively.

3.3.3 Conclusions

Keeping in mind that "it is difficult to make general conclusions" as to how the performance of $m_1(\hat{\boldsymbol{\theta}})$ and alternate estimators is affected by a number of factors (Zimmerman and Cressie, 1992) the following tentative conclusions may be drawn from the results of the simulation experiment.

The EMSPE $m_1(\boldsymbol{\theta})$ results in severe underestimation of $m_2(\boldsymbol{\theta})$ especially when the spatial correlation is strong regardless of whether $\hat{\boldsymbol{\theta}}_{ML}$ or $\hat{\boldsymbol{\theta}}_{REML}$ is used as an estimator of $\boldsymbol{\theta}$. This finding that results in this conclusion is contrary to the findings of Zimmerman and Zimmerman (1991) as well as Zimmerman and Cressie (1992) who found the performance of $m_1(\hat{\boldsymbol{\theta}})$ to be, in the words of Zimmerman and Cressie (1992), "adequate" when the spatial correlation is strong.

The Prasad-Rao estimator with $\hat{\boldsymbol{\theta}}_{REML}$ as an estimator of $\boldsymbol{\theta}$ is the most optimal estimator of the 5 estimators studied here in all situations, that is, for Gaussian random fields with weak, moderate and strong spatial correlation.

3.3.4 General recommendations

In line with the above conclusions the following recommendations can be made concerning suitable estimators of $m_2(\boldsymbol{\theta})$.

- The use of the EMSPE $m_1(\hat{\boldsymbol{\theta}})$ should be avoided.
- The Prasad-Rao estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ should be employed when the spatial correlation is moderate to strong.
- When $\boldsymbol{\theta}$ is estimated by $\hat{\boldsymbol{\theta}}_{ML}$ and the spatial correlation is weak the unconditional bootstrap estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ should be employed to estimate $m_2(\boldsymbol{\theta})$ if $\boldsymbol{\theta}$ is estimated by $\hat{\boldsymbol{\theta}}_{REML}$ then $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ may be used.
- The use of $\hat{\boldsymbol{\theta}}_{REML}$ instead of $\hat{\boldsymbol{\theta}}_{ML}$ as an estimator of $\boldsymbol{\theta}$ is recommended. However, in some instances, for example the Meuse example in Chapter 4, the ML estimator may be more stable.

Chapter 4

Applications

4.1 River Meuse Heavy Metals Data

4.1.1 Description

The Meuse data set contains measurements of the concentration levels of heavy metals, in parts per million (ppm), from soil samples at 155 locations on the flood plain of the river Meuse near the village of Stein in the Netherlands. Also included in the data set is information on the flood frequency, elevation, land use, soil type and two "distance to the river" measures of the sample locations. A grid of prediction locations is also included, consisting of 3103 sites, and information comprising the flood frequency, soil type and one of the distance measures, *normalized distance*, recorded at the sample sites is also available for the prediction locations. According to Pebesma (2001), the normalized distances from the Meuse, which are accurate to 20 m, were obtained through a spread GIS operation and were scaled to lie between 0 and 1.

For the analysis carried out here, the data were obtained from the R package `gstat` (Pebesma, 2004) where it is used to demonstrate the functionality of the package. The data were however originally collected by Rikken and van Rijn (1993) with the intention of studying the concentrations of the heavy metals throughout the flood plain. This objective can be addressed using kriging as is done here.

Though the concentration of various heavy metals, namely cadmium (Cd), zinc (Zn), lead (Pb) and mercury (Hg), were measured at the sample sites, this research focuses solely on the Zn levels in the flood plain as in Burrough and McDonnell (1998) and Pebesma (2010). Table 4.1 shows an extract of the data. The coordinates of the data are in Rijksdriehoek (RDH), the Netherlands topographical map coordinates. The system is based on the reference, in metres, to a tower in Amersfoort, Netherlands. Also shown in Table 4.1 are the two distance measures, the raw distance in metres to the main river channel plus the normalized distance. Figure 4.1 shows all the 155 sample locations and the reported Zn content in ppm at those locations.

Table 4.1: A sample of the first 15 points of the Meuse data set with the coordinates, distance to the river in metres, normalized distance, Zinc (Zn) concentrations in ppm and log(Zn) values.

x (Easting)	y (Northing)	distance (m)	distance (normalized)	Zn (ppm)	ln(Zn)
181072	333611	50	0.00135803	1022	6.929517
181025	333558	30	0.01222430	1141	7.039660
181165	333537	150	0.10302900	640	6.461468
181298	333484	270	0.19009400	257	5.549076
181307	333330	380	0.27709000	269	5.594711
181390	333260	470	0.36406700	281	5.638355
181165	333370	240	0.19009400	346	5.846439
181027	333363	120	0.09215160	406	6.006353
181060	333231	240	0.18461400	347	5.849325
181232	333168	420	0.30970200	183	5.209486
181191	333115	400	0.31511600	189	5.241747
181032	333031	300	0.22812300	251	5.525453
180874	333339	20	0.00000000	1096	6.999422
180969	333252	130	0.11393200	504	6.222576
181011	333161	220	0.16833600	326	5.786897

The Zn, and indeed the other heavy metals, in the soil samples originate from polluted sediment that is carried by the river from up stream and then deposited on the flood plain. Since it appears that the governing process for the sediment deposition dictates that most of the polluted sediment is deposited close to the river bank (Pebesma, 2004) the concentration levels at each location should depend on their distance to the river. Following Pebesma (2004) the normalized distance to the river was therefore considered as an important covariate for all subsequent analysis. The raw distance to the river was not used as it is not given at the prediction locations as noted earlier.

4.1.2 Exploratory data analysis

An exploratory analysis of the Zn data using **geoR** (Riberio and Diggle, 2001) was firstly carried out. It was found that the concentration data have minimum and maximum values of 113.0 ppm and 1839.0 ppm respectively, with a median value of 326.0 ppm and a mean value of 469.0 ppm. The minimum value is located at (180328, 331158), a sample site that is at a considerably large distance from the river channel and the maximum value observed is at (179973, 332255), a sample site on the river bank.

A graphical exploratory data analysis that is produced by **geoR** is shown in Figure 4.2 In the upper-left panel of the figure, red crosses indicate data in the upper

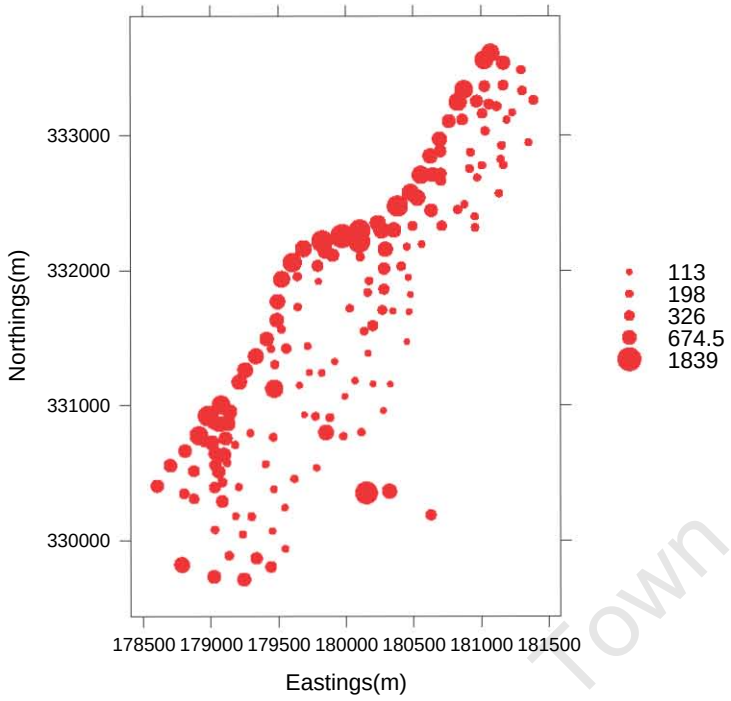


Figure 4.1: Zinc (Zn) concentration in the soil samples, in ppm, at 155 sample sites on the flood plain of the river Meuse (near Stein, Netherlands). The Meuse river is located on the west of the sample points. The area of the circles in the figure are proportional to the observed concentration levels at the locations. The legend gives the five number summary of the data.

quartile, blue circles, data in the lower quartile, yellow crosses and green rectangles data in the third and second quartiles, respectively. The locally weighted regression smoother (LOWESS) curves fitted to the scatter plots of the Zn concentrations against the spatial coordinates, shown in the upper-right and lower-lower panels of Figure 4.2, are used to determine whether the data exhibit a trend in that coordinate.

From the upper-left panel of Figure 4.2, it is apparent that data points with high values of Zn levels are mostly located on the west of the study area, that is on the bank of river, as is to be expected. This is also apparent from the lower-left panel of Figure 4.2. In this latter panel it is apparent that the LOWESS curve slopes downwards, which indicates that the concentrations decrease with an increase in the x coordinate, that is on moving away from the river. This suggests the inclusion of distance to the river as a covariate for kriging. The histogram of the data shown in the lower-right panel of Figure 4.2 reveals that the data are heavily skewed to the right, implying that the process generating the data does not follow a Gaussian distribution. This deviation of the data from a Gaussian distribution is confirmed by the normal Q-Q plot of the data, given in the left panel of Figure 4.3. This would suggest the need for transformation of the data in order to make it approximately Gaussian and thus enable the use of Gaussian kriging.

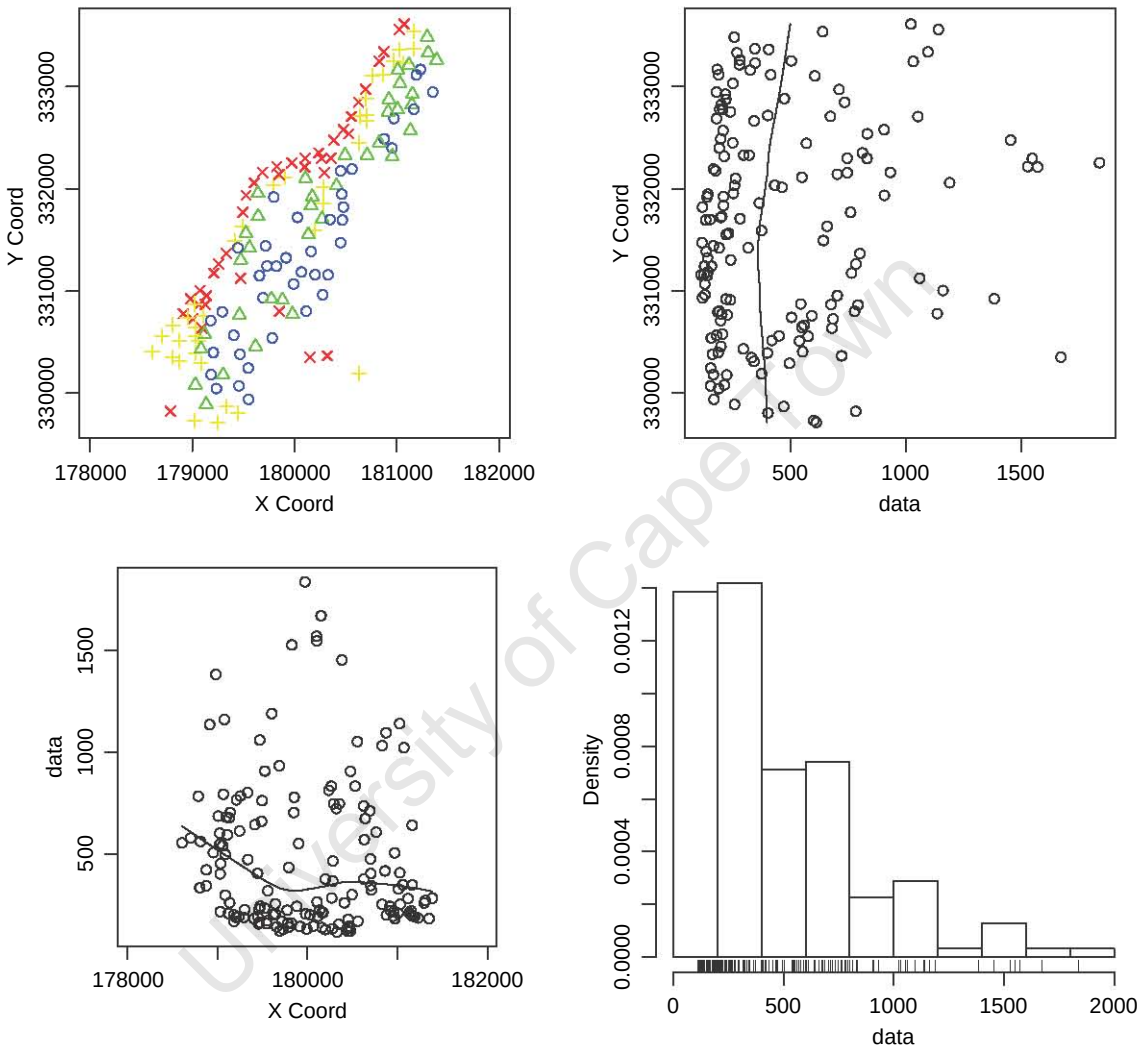
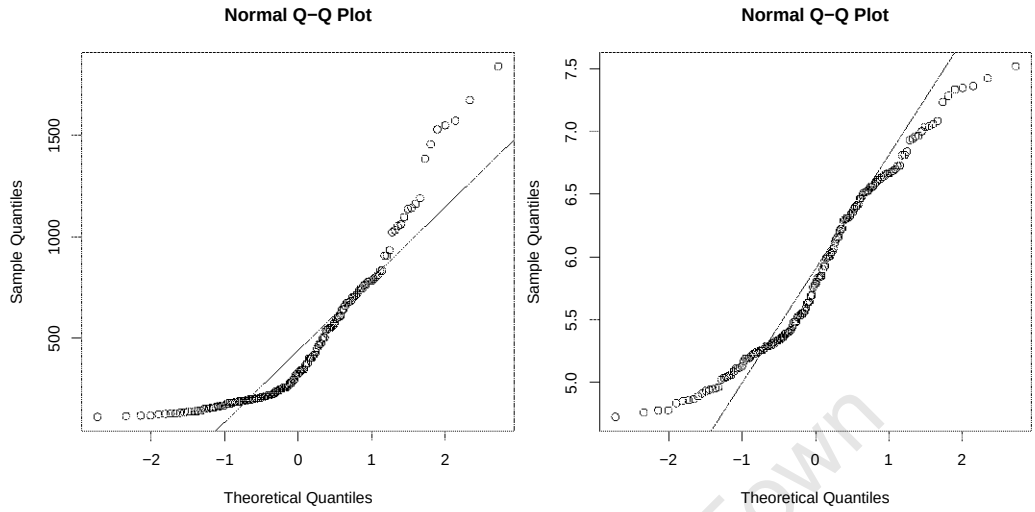


Figure 4.2: Upper-left panel shows the quartiles of the zinc data, the upper-right and lower-left panels show the data plotted against the x and y coordinates respectively, with LOWESS lines fitted to them. The lower-right panel shows the histogram of the zinc data.



(a) (b)

Figure 4.3: Normal Q-Q plots of (a) the zinc concentrations and (b) the log transformed zinc concentrations of the 151 observation locations in the Meuse data set.

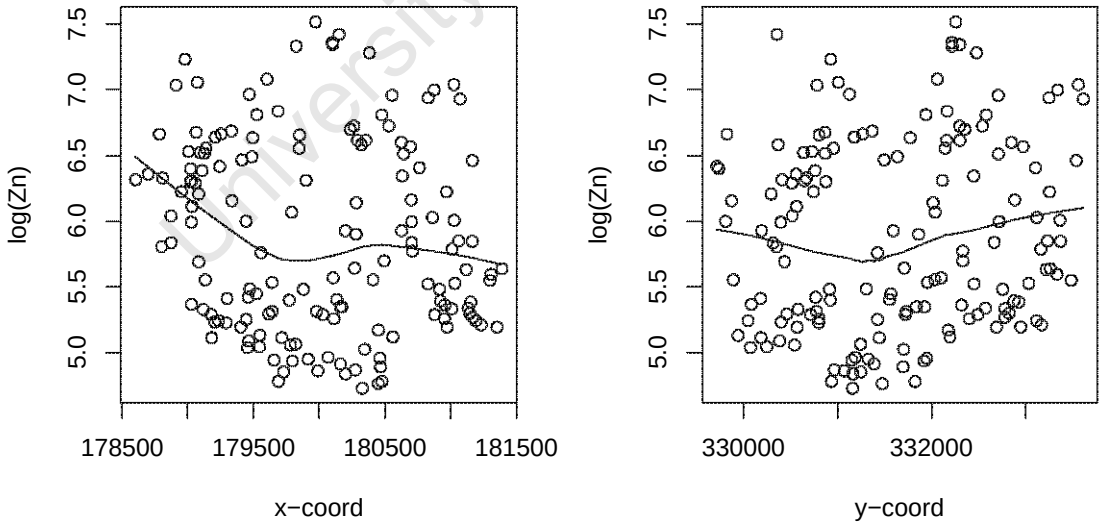


Figure 4.4: Log transformed zinc values, log(Zn), against the coordinates.

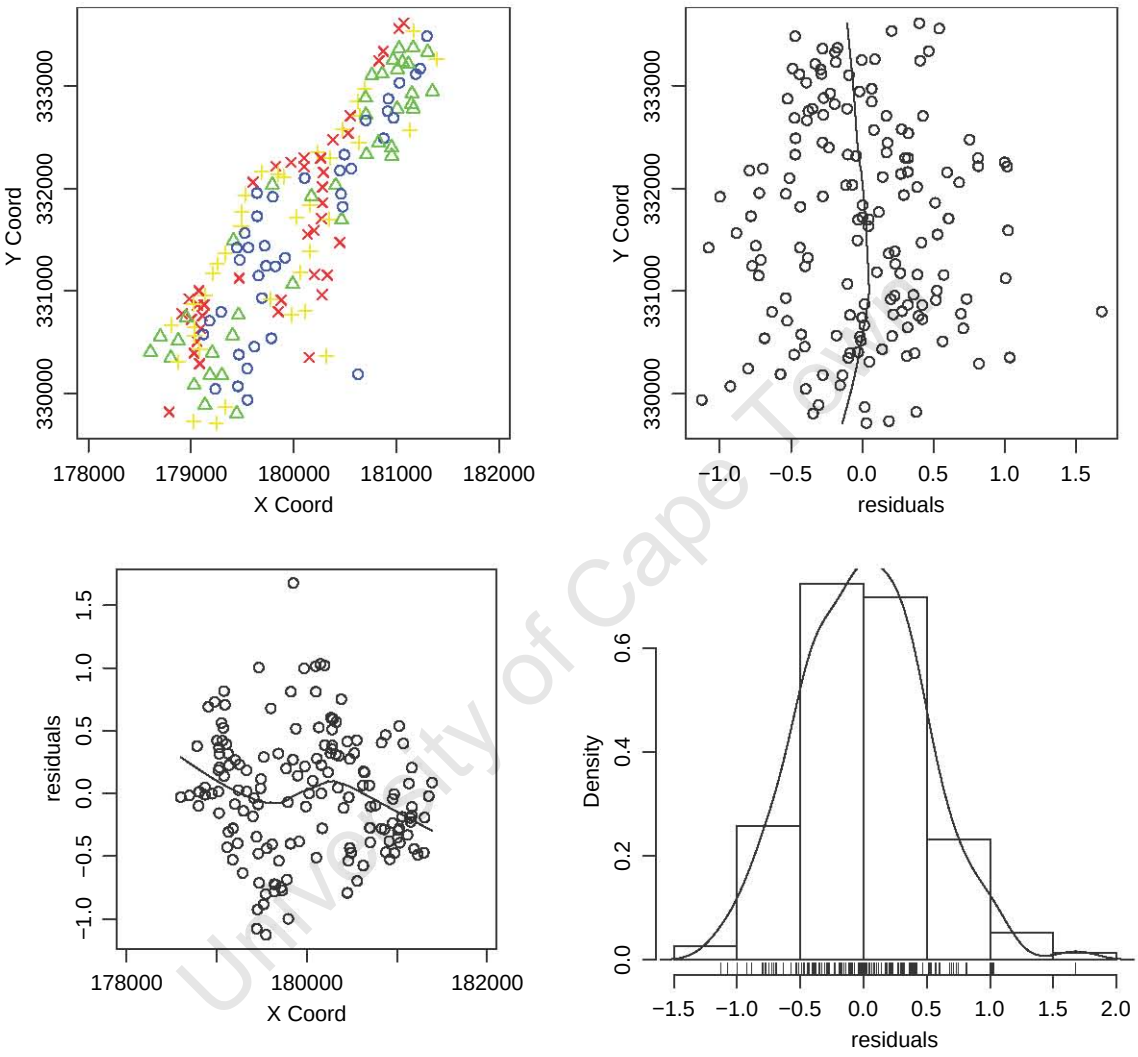


Figure 4.5: Upper-left panel shows the quartiles of the residuals resulting from removing the trend in the log(Zn) data, the upper-right and lower-left panels show the data plotted against the x and y coordinates respectively. The diagram in the lower-right panel shows the histogram of the residuals.

A log transform was applied to the data, as is done in the analysis by Pebesma (2010). The normal Q-Q plot of the log-transformed data, shown in the right-panel of Figure 4.3, suggests that the resulting data are not strictly Gaussian but that it is not unreasonable to assume that they were generated by a Gaussian process. The log transformed zinc data, log(Zn), were checked in order to determine if the trend seen in the Zn data persisted. This was done through a plot of log(Zn) against each of the coordinates as shown in Figure 4.4. These plots reveal that the trend in the x coordinate is still present. There is a trend in the y-coordinate but it is not clear. The normalized distance to the river, which accommodates the trend in the x-coordinate, was therefore included as covariate information similar to Pebesma (2010).

The log(Zn) data was detrended by regressing against the normalized distance to the river and an exploratory analysis performed on the residuals in order to confirm the near-Gaussian nature of the data The results of this analysis are shown in Figure 4.5. It can be seen from the plot of the residuals in the upper-left panel of the figure, that large and small values are more evenly distributed over the study area. The histogram, lower-right panel, suggests that the residuals follow an approximately Gaussian distribution.

4.1.3 Model fitting

Four ordinary kriging models with various covariance functions were fitted to the log-transformed data, log(Zn), via ML estimation. REML estimation of the covariance parameters was not employed as it resulted in multi-modal profile likelihoods. The covariance functions employed were an exponential, Gaussian, spherical and Matérn covariance functions with κ values of 1.5 and 2.5. Five universal kriging models with normalized distance to the river as a covariate were also fitted with the same covariance functions and using ML estimation. Parameter estimates, values of the maximized likelihood and the AIC and BIC values for the four resultant ordinary and the four universal kriging models are shown in Table 4.2. Plots of semivariograms obtained from fitting the various ordinary and universal kriging models are shown together with the corresponding empirical semivariograms in Figure 4.6.

From Table 4.2 it can be seen from the AIC and BIC values of the four ordinary kriging models that the model with a Matérn covariance function with κ=1.5 is the best fitting model to log(Zn). It can also be seen from the table that of the four universal kriging models, the model with a Gaussian covariance function is the best fitting model. Comparisons between ordinary and universal kriging models with similar covariance functions indicate that in each case the universal kriging models provide better fits with a marked reduction in the estimated nugget τ^2, partial sill σ_0^2 and range ϕ values. The reduction in the estimates of τ^2, σ_0^2 and ϕ reflect the fact the covariate, normalized distance to the river, is able to explain a substantial amount of the spatial variation.

Table 4.2: Estimates of the parameters of various ordinary and universal kriging models together with the maximized log-likelihood values and the AIC and BIC. The universal kriging models were formed by taking the normalized distance to the river as a covariate.

Model	κ	$\hat{\beta}$	$\hat{\tau}^2$	$\hat{\sigma}_0^2$	$\hat{\phi}$	L	AIC	BIC
Ordinary kriging models								
Exponential	-	6.5577	0.0335	1.4987	1700.0000	-99.16	206.3	218.5
Matérn	1.5	6.4910	0.0950	1.4140	440.1270	-97.38	202.8	214.9
	2.5	6.3780	0.1040	1.1470	259.5440	-97.82	203.6	215.8
Gaussian	-	6.2391	0.1146	0.8743	572.2880	-99.43	206.9	219.0
Spherical	-	6.1653	0.0332	0.6961	1200.5092	-97.88	203.8	215.9

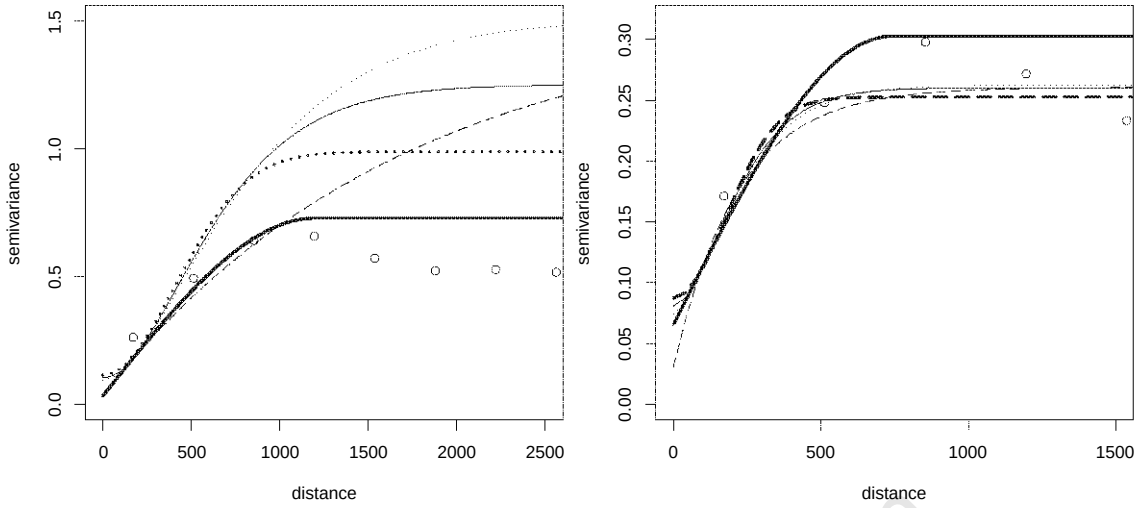
Model	κ	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\tau}^2$	$\hat{\sigma}_0^2$	$\hat{\phi}$	L	AIC	BIC
Universal kriging models									
Exponential	-	6.5958	-2.8186	0.0309	0.2298	220.8617	-86.01	182.0	197.2
Matérn	1.5	6.5912	-2.8201	0.0742	0.1882	122.9547	-84.78	179.6	194.8
	2.5	6.5853	-2.8138	0.0808	0.1793	90.4478	-84.48	179.0	194.2
Gaussian	-	6.5662	-2.7861	0.0873	0.1653	247.1482	-84.28	178.6	193.8
Spherical	-	6.6413	-2.9108	0.0660	0.2367	738.3997	-84.71	179.4	194.6

As noted earlier, of the kriging models fitted with covariate information, the kriging model with a Gaussian covariance is the best fitting model with AIC and BIC values of 178.6 and 193.8 respectively. However, the AIC and BIC indicate there is not much difference between this kriging model and the model fitted with Matérn covariance functions. Semivariograms of these three models are also fairly similar as shown in the right-hand panel of Figure 4.6. The kriging model with an exponential covariance function is the worst performing model with AIC and BIC values of 182.0 and 197.2. The best kriging model, that is the model with a Gaussian covariance, was thus selected for further investigations.

The profile likelihoods of each of the variance parameters of the selected kriging model, that is the universal kriging model with a Gaussian covariance structure

$$\tau^2 + \sigma_0^2 \exp^{-(h/\phi)^2},$$

were then calculated and plotted using the **geoR** functions `proflik` and `plot.prolik` respectively. The plotted profile likelihoods are shown in Figure 4.7. From the figures it can be seen that all the variance parameters have profile likelihoods that have clearly defined maxima. This indicates that all the parameters are estimated with good precision. Approximate 95% confidence intervals (CI), for the parameters in the universal kriging model with a Gaussian covariance function, based on the profile likelihoods, are given in in Table 4.3.



(a) Constant mean (b) Linear trend

Figure 4.6: Comparison of various fitted semivariogram models to the empirical semivariogram (a) of the data and (b) of the residuals from a trend in the x coordinate. The dashed lines correspond to fits with the exponential, solid lines to a Matérn with $\kappa = 1.5$, the dotted lines to a Matérn with $\kappa = 2.5$, the heavy dots to a model with a Gaussian and the heavy solid line to the spherical covariance model. The circles represent the empirical semivariograms.

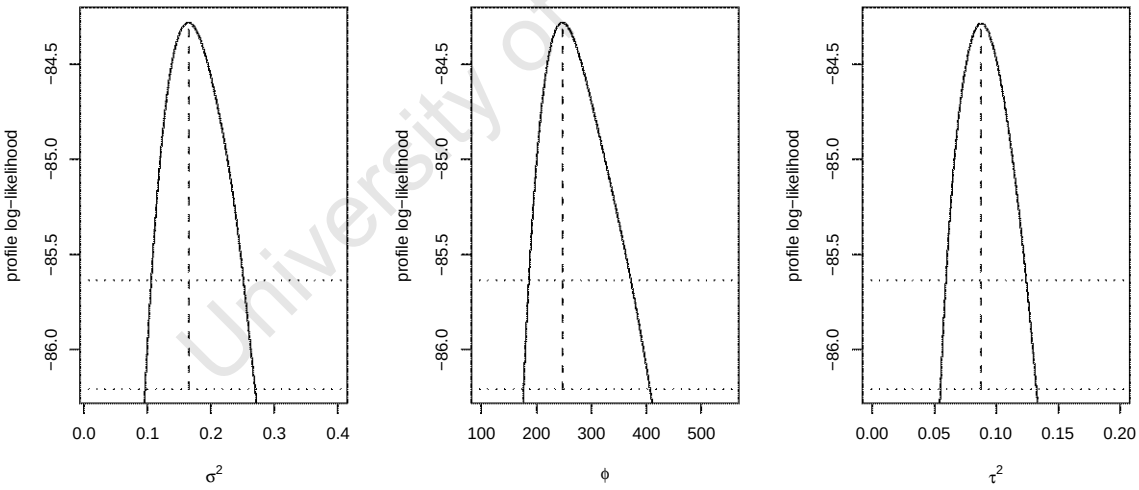


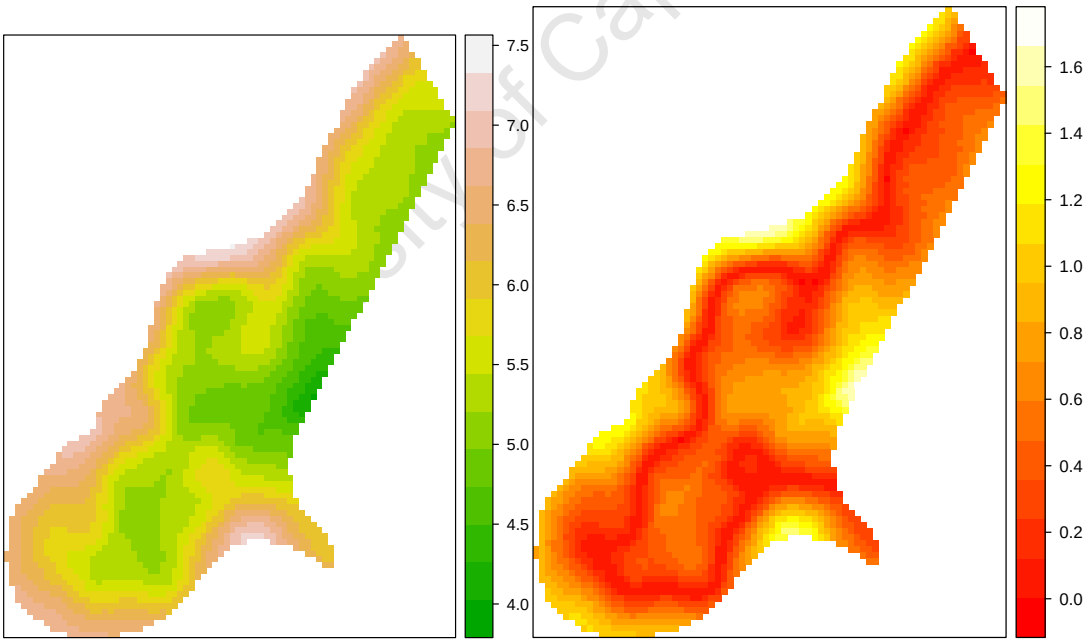
Figure 4.7: Profile likelihoods of σ_0^2 (left panel), ϕ (middle panel) and τ^2 (right panel) for the universal kriging models with a Gaussian covariance model. The two horizontal lines on each of the plots define approximate 90% and 95% confidence intervals of each of the parameters, based on the asymptotic $\frac{1}{2}\chi^2(1)$-distribution of the log likelihood ratio.

Table 4.3: Estimates of the variance parameters and their approximate 95% confidence bounds, in parenthesis, for the universal kriging model with a Gaussian covariance function.

	Parameters		
	τ^2	σ_0^2	ϕ
Estimates	0.0873	0.1653	247.1482
95% CI	(0.051, 0.180)	(0.10, 0.27)	(175, 400)

4.1.4 Kriging and kriging variance estimation

Kriging predictions at the 3103 locations forming the prediction grid were made using the universal kriging model fitted with a Gaussian covariance function model. The resulting predicted log(Zn) values were then mapped and the mapped values are shown in the left panel of Figure 4.8 and the associated kriging standard deviations in the right panel of that figure.



(a) Predictions (b) Standard deviations

Figure 4.8: (a) Universal kriging surface of the log-zinc values on the flood plain of the river Meuse obtained using the kriging model with a Gaussian covariance function and (b) a map of the corresponding kriging standard deviations.

The kriging variance at each of the 3103 prediction locations was then estimated via the empirical mean squared prediction error (EMSPE) $m_1(\hat{\boldsymbol{\theta}})$, the Kacker-Harville $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and Prasad-Rao $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ estimators as well as the conditional and unconditional bootstrap estimators $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$. The approximate 95% confidence bounds of the range parameter ϕ, as shown in Table 4.3, were used to bound the range estimates in the unconditional bootstrap so as to ensure that the kriging models fitted to the bootstrapped data were representative of the actual kriging models under study (see Section 2.4.2.1). The bootstrap estimators $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ were calculated over 5000 iterations of their respective algorithms.

Summary statistics of the kriging variance estimates are displayed in Table 4.4 and the associated notched box and whisker plots are shown in Figure 4.9. The nothches in the box plot tell us if the are any differences in the medians of the values they represent. If the notches of two plots do not overlap this is indicates that the two medians differ. Maps of the kriging variance estimates obtained using the various estimators are shown in Figure 4.10. Although an assessment of how well each of the estimators approximate the mean squared prediction error (MSPE) of the empirical best linear unbiased predictor (EBLUP), $m_2(\boldsymbol{\theta})$, cannot be made the following observations are noted.

Table 4.4: Summary statistics of the MSPE of the EBULP estimates at the 3103 prediction locations obtained using various kriging variance estimators. Bootstrap estimators, $\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{CBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{UBS}$, were calculated over 5000 iterations of their respective algorithms.

	KRIGING VARIANCE ESTIMATORS				
Summary Statistics	$m_1(\hat{\boldsymbol{\theta}}_{ML})$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{KH}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{PR}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{CBS}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{UBS}$
Minimum	0.1026	0.1065	0.1090	0.0989	0.0987
1st Quartile	0.1203	0.1254	0.1291	0.1205	0.1222
Median	0.1323	0.1364	0.1414	0.1324	0.1342
Mean	0.1456	0.1499	0.1534	0.1466	0.1476
3rd Quartile	0.1618	0.1635	0.1653	0.1612	0.1625
Maximum	0.2712	0.2719	0.2727	0.2790	0.2613
Standard deviation	0.0367	0.0355	0.0346	0.0369	0.0353

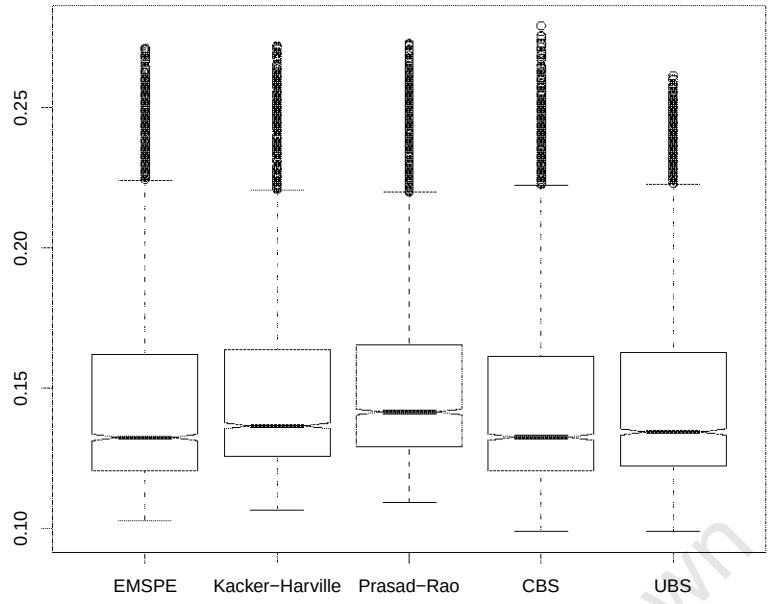


Figure 4.9: Notched box and whisker plot of the kriging variance estimates at the 3103 prediction locations obtained for the 5 kriging variance estimators.

4.1.4.1 Some observations on the estimates and estimators

Firstly, it is apparent from Table 4.4 and Figure 4.9 that the Prasad-Rao estimator, $\sigma^2_{PR}(\hat{\boldsymbol{\theta}})$ produces, on average, the highest estimates of the variance with a mean value of 0.1534. This observation of $\sigma^2_{PR}(\hat{\boldsymbol{\theta}})$ is consistent with the findings of the simulation experiment of Chapter 3. Further, from Table 4.4 and Figure 4.9 it can be seen that the EMSPE, $m_1(\hat{\boldsymbol{\theta}})$, and the conditional bootstrap estimator, $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$, produce the smallest kriging variance estimates on average and furthermore these are similar in value as can be seen by closeness of the mean and median values of these estimators which are 0.1456 and 0.1323, respectively for $m_1(\hat{\boldsymbol{\theta}})$ and 0.1466 along with 0.1324 for $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$. This again is in line with the results from the simulation experiments. In addition, the estimates obtained from the Kacker-Harville estimator, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$, and unconditional bootstrap estimator, $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$, appear from Figure 4.9 to be similar as was observed in the simulation experiment.

Since the kriging variance estimates for this log(Zn) data obtained from the various estimators exhibit behaviour, in terms of the relationships to one another, similar to that seen in the simulation experiment, it would lead one to suggest that the performance of the estimators in approximating $m_2(\boldsymbol{\theta})$ for the real data here is similar to that observed in the simulation experiment. That is, the Prasad-Rao is the best estimator of $m_2(\boldsymbol{\theta})$.

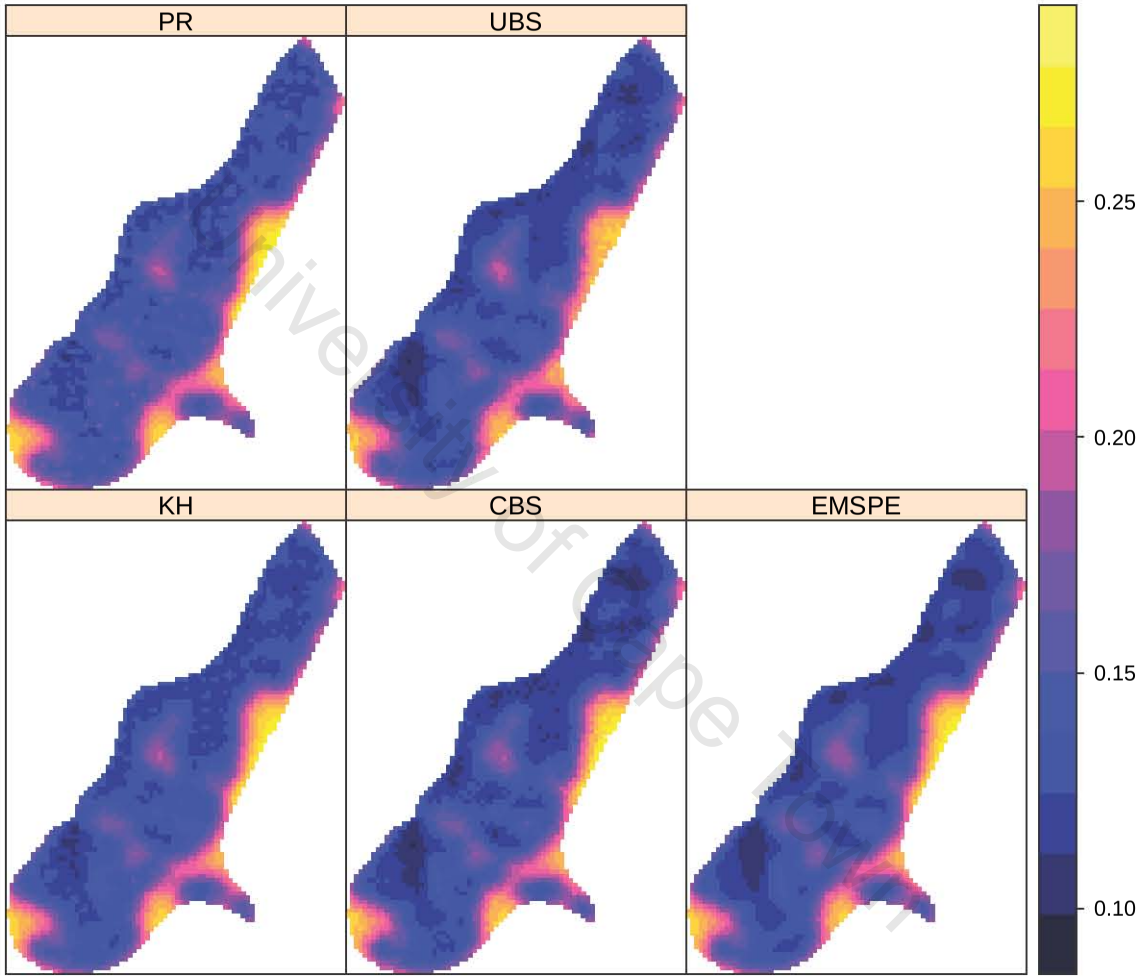


Figure 4.10: Maps showing surfaces of kriging variance estimates obtained using the Prasad-Rao (PR), the unconditional bootstrap (UBS), the Kacker-Harville (KH), the conditional bootstrap (CBS) and empirical mean squared prediction error (EMSPE) under the kriging model with a Gaussian covariance function.

A few observations regarding the kriging variance surfaces obtained from each of the estimators and shown in Figure 4.10 can also be made. From Figure 4.10 it is apparent that the estimates from $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ produce kriging variance surfaces that are smoother than the surface produced by $m_1(\hat{\boldsymbol{\theta}})$. It can also be seen that though the surface produced by $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ has characteristics that are very similar to those of the surface produced by $m_1(\hat{\boldsymbol{\theta}})$, the surface produced by the former estimator is less smooth than that obtained from the latter estimator. Similarly the surface produced by $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ seems rough as compared to those produced by the three analytic estimators $m_1(\hat{\boldsymbol{\theta}})$, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$.

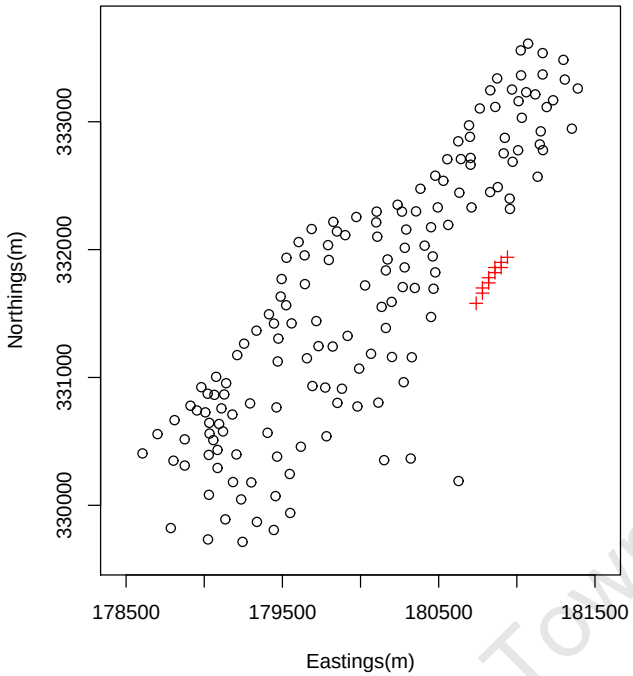
4.1.4.2 An brief exploration into optimal design

As noted earlier, the kriging variance is used in various ways as the basis for a design criterion in optimal sampling design (Cressie *et al*, 1990; Zimmerman, 2006). For instance, for any particular kriging variance estimator, a common approach to determining which points should be included in future sampling is to rank the estimates in terms of their magnitude. Those points with the highest kriging variance are then identified as future sample locations because the kriging predictions are associated with low precision. To examine how the different estimators studied here would affect the decision as to where future observations should be taken, the 10 locations with the highest kriging variance were identified for each estimator. The results are summarized in Table 4.5 and Figure 4.11.

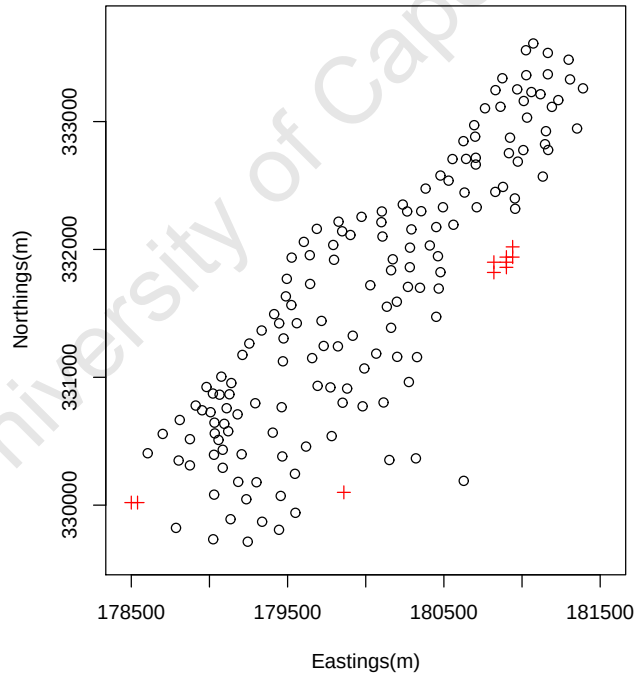
It is clear that the rankings of the kriging variance for the estimators $m_1(\hat{\boldsymbol{\theta}})$, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ are the same and that the maximum value occurs at (180900, 331860), a location far from the river and close to the edge of the study area. The conditional bootstrap estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ provides the same set of 10 locations with the highest kriging variance, but in a completely different order to that of the three analytic estimators $m_1(\hat{\boldsymbol{\theta}})$, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$. In contrast, although some of the set of 10 locations with the highest kriging variance values for $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ are also identified for the other estimators this set of locations does not coincide exactly with those for the other estimators.

Table 4.5: Top ten prediction locations and estimates with the highest kriging variance as identified by the various estimators. Locations given in parenthesis and variance estimates in bold.

$m_1(\hat{\boldsymbol{\theta}}_{ML})$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{KH}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{PR}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{CBS}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{UBS}$
(180900, 331860) **0.2712**	(180900, 331860) **0.2719**	(180900, 331860) **0.2727**	(180820, 331780) **0.2790**	(180900, 331940) **0.2613**
(180860, 331820) **0.2707**	(180860, 331820) **0.2715**	(180860, 331820) **0.2723**	(180780, 331660) **0.2755**	(180900, 331860) **0.2603**
(180820, 331740) **0.2703**	(180820, 331740) **0.2712**	(180820, 331740) **0.2721**	(180900, 331860) **0.2752**	(180900, 331900) **0.2598**
(180780, 331660) **0.2697**	(180780, 331660) **0.2707**	(180780, 331660) **0.2718**	(180860, 331820) **0.2743**	(180940, 331940) **0.2581**
(180900, 331900) **0.2690**	(180900, 331900) **0.2698**	(180900, 331900) **0.2707**	(180740, 331580) **0.2728**	(179860, 330100) **0.2577**
(180860, 331860) **0.2685**	(180860, 331860) **0.2694**	(180860, 331860) **0.2703**	(180780, 331700) **0.2724**	(178540, 330020) **0.2577**
(180940, 331940) **0.2684**	(180940, 331940) **0.2692**	(180940, 331940) **0.2701**	(180820, 331740) **0.2714**	(180820, 331900) **0.2576**
(180820, 331780) **0.2681**	(180820, 331780) **0.2690**	(180820, 331780) **0.2700**	(180780, 331740) **0.2702**	(180940, 332020) **0.2570**
(180780, 331700) **0.2674**	(180780, 331700) **0.2685**	(180780, 331700) **0.2696**	(180940, 331940) **0.2694**	(180820, 331820) **0.2567**
(180740, 331580) **0.2669**	(180740, 331580) **0.2681**	(180740, 331580) **0.2692**	(180860, 331860) **0.2692**	(178500, 330020) **0.2565**



(a)



(b)

Figure 4.11: Circles show sample locations and crosses show locations the 10 locations which have the highest kriging variance under: (a) the EMSPE, Kacker-Harville, Prasad-Rao and conditional bootstrap estimator, (b) unconditional bootstrap estimators .

4.2 South African Micronutrient Data

4.2.1 Description

The data set of 214 soil samples used here is part of data from a study by de Villiers *et al* (2010) in which surface soil samples were collected at approximately 1000 locations across South Africa between 2005 and 2008. The intention of the study was to estimate, by inverse distance interpolation (IDW), the micronutrient content in soils across the country. Though data on various micronutrients were collected, the study focused specifically on lead (Pb) and cadmium (Cd) content. The soil samples were collected at locations that did not exhibit signs of any prior human activity such as mining or agricultural activity. Each of the samples is a composite sample of subsamples taken within a 100 m^2 area from the upper 10 cm of the soil sequence and the reported micronutrient values of the samples were determined by use of the Mehlich-3 extraction method (de Villiers *et al*, 2010).

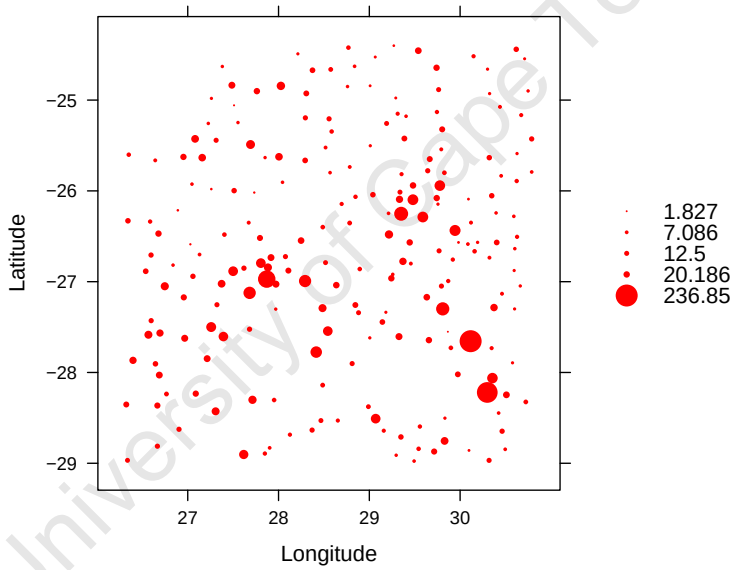


Figure 4.12: Iron (Fe) levels at the 214 sample locations between longitudes 26.32 to 30.79 and latitudes -28.98 to -24.40. The area of the circles in the figure are proportional to the observed concentration levels at the locations. The legend gives the five number summary of the data.

In this study, focus is on the amounts of iron (Fe) in the soil samples in and around the Witwatersrand area in the Gauteng province of South Africa. Figure 4.12 shows the study area which is between longitudes 26.32 and 30.79 and latitudes -28.98

Table 4.6: A sample of the iron (Fe) data set. The first 15 data points of the data, with the coordinates, Iron (Fe) concentrations and log(Fe) values.

Longitude	Latitude	Fe (ng/g)	log(Fe)
27.61660	-28.90322	42.13319	3.740836
27.84943	-28.89308	9.42192	2.243039
27.90203	-28.83018	7.07119	1.956029
28.12537	-28.68503	9.05852	2.203706
28.36962	-28.63455	13.38876	2.594416
28.46692	-28.52972	12.19404	2.500947
28.65308	-28.52965	8.64924	2.157471
28.98972	-28.37685	11.60196	2.451174
29.06987	-28.50855	45.84092	3.825177
29.15915	-28.64143	8.65618	2.158274
29.34805	-28.71025	14.72193	2.689338
29.55713	-28.59552	9.57662	2.259325
29.83258	-28.50325	5.6238	1.727008
29.97493	-28.02055	16.58198	2.808317
29.89837	-27.72882	10.83388	2.382678

and -24.40. The figure also shows the levels of Fe, in nanograms per gram (ng/g), measured in each of the soil samples at the 214 observation locations. On the side of the map is a legend illustrating the five number summary of the data. An extract of the data is presented in Table 4.6.

4.2.2 Exploratory data analysis

The data have minimum and maximum values of 1.827 ng/g and 236.900 ng/g respectively, with a mean value of 19.360 ng/g and a median value of 12.500 ng/g. From the upper-left panel in Figure 4.13 which shows a map of the quartiles of the data, the iron levels can be seen to be evenly spread over the study domain with a few clusters of similar values but no indication of the presence of a trend. The plots of the data against the coordinates shown in the upper-right and lower-left panels of Figure 4.13 also point towards there being no apparent trend but only to the data being heavily skewed. This is shown more clearly in the histogram of the data, in the lower-right panel of Figure 4.13. The histogram of the data indicates that the Fe data are markedly non-Gaussian as the histogram is heavily skewed to the right. The normal Q-Q plot, left panel Figure 4.14, also indicates departure of the data from a Gaussian distribution.

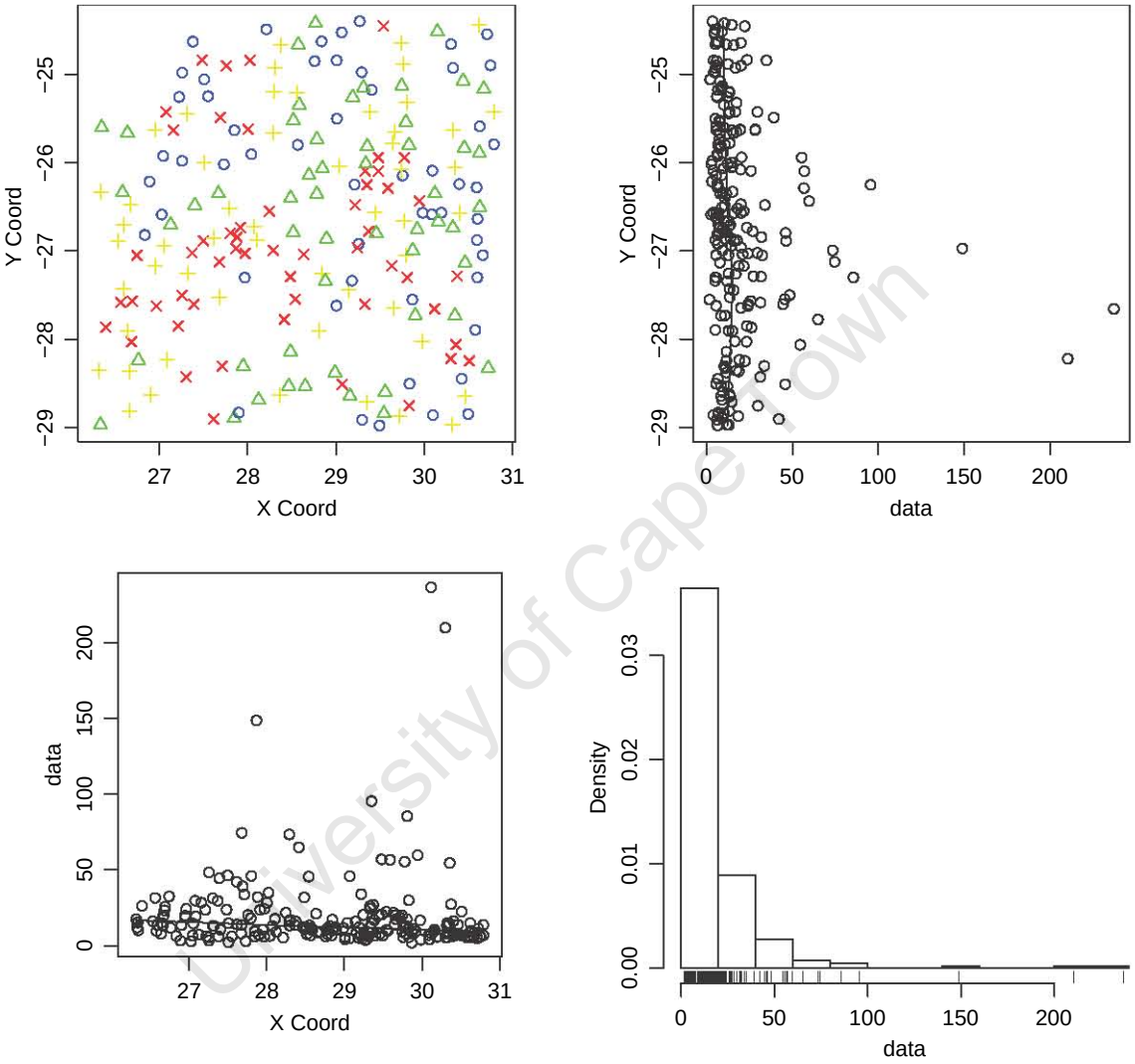


Figure 4.13: Upper-left panel shows a map of the quartiles of the iron data, the upper-right and lower-left panels show the data plotted against the y and x coordinates respectively. The diagram in the lower-right panel shows the histogram of the data.

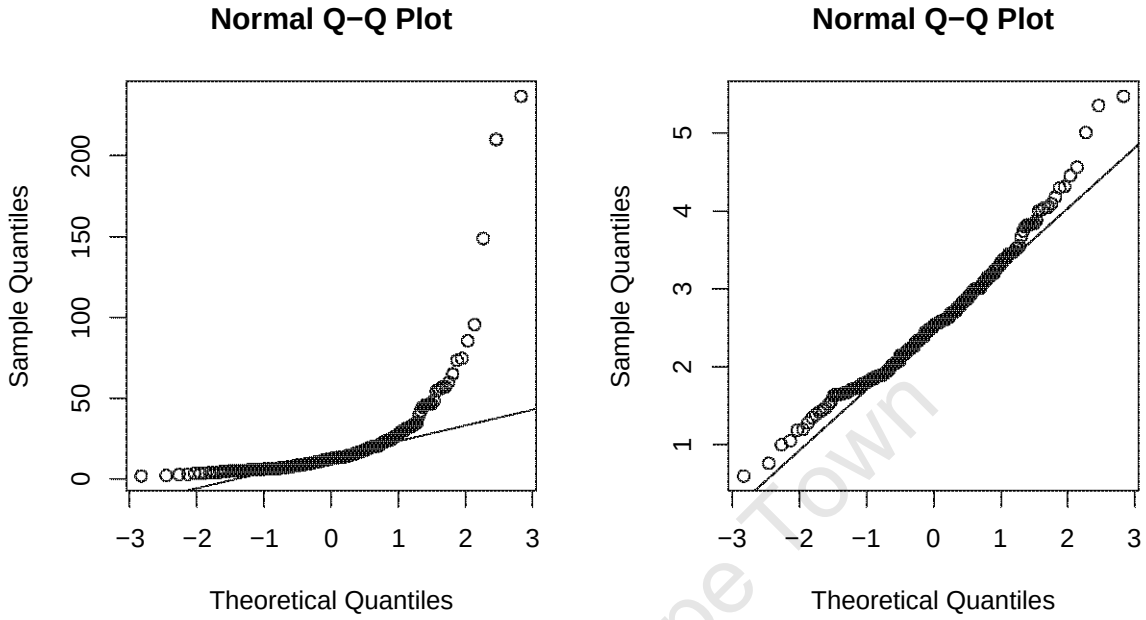


Figure 4.14: Normal Q-Q plot of iron measurements (left panel) and the log(Fe) values (right panel) at the 214 sample locations.

A log transformation was applied to the Fe data. Figure 4.15 shows a graphical exploratory data analysis (EDA) of the log-transformed data, termed log(Fe), similar to that shown in Figure 4.13. It can be seen from this figure that the histogram of the log(Fe), lower-right panel, is more symmetric than that of the original data. This indicates that the log transform of the data was adequate in normalizing the data, at least approximately. The normal Q-Q plot of the log(Fe) data, right panel of Figure 4.14 also confirms that the log-transform was adequate in generating approximately normally distributed data since the values of log(Fe) lie close to the diagonal axis of the plot. From the plots of log(Fe) against the coordinates shown in the upper-right and lower-left panels of Figure 4.15, it is clear that the log-transformed data does not seem to exhibit any trend. The map showing the quartiles of the log(Fe) data, upper-left panel of Figure 4.15, also indicates no trend. Hence a constant mean assumption, that is an ordinary kriging model, would seem to be applicable to the log(Fe) data.

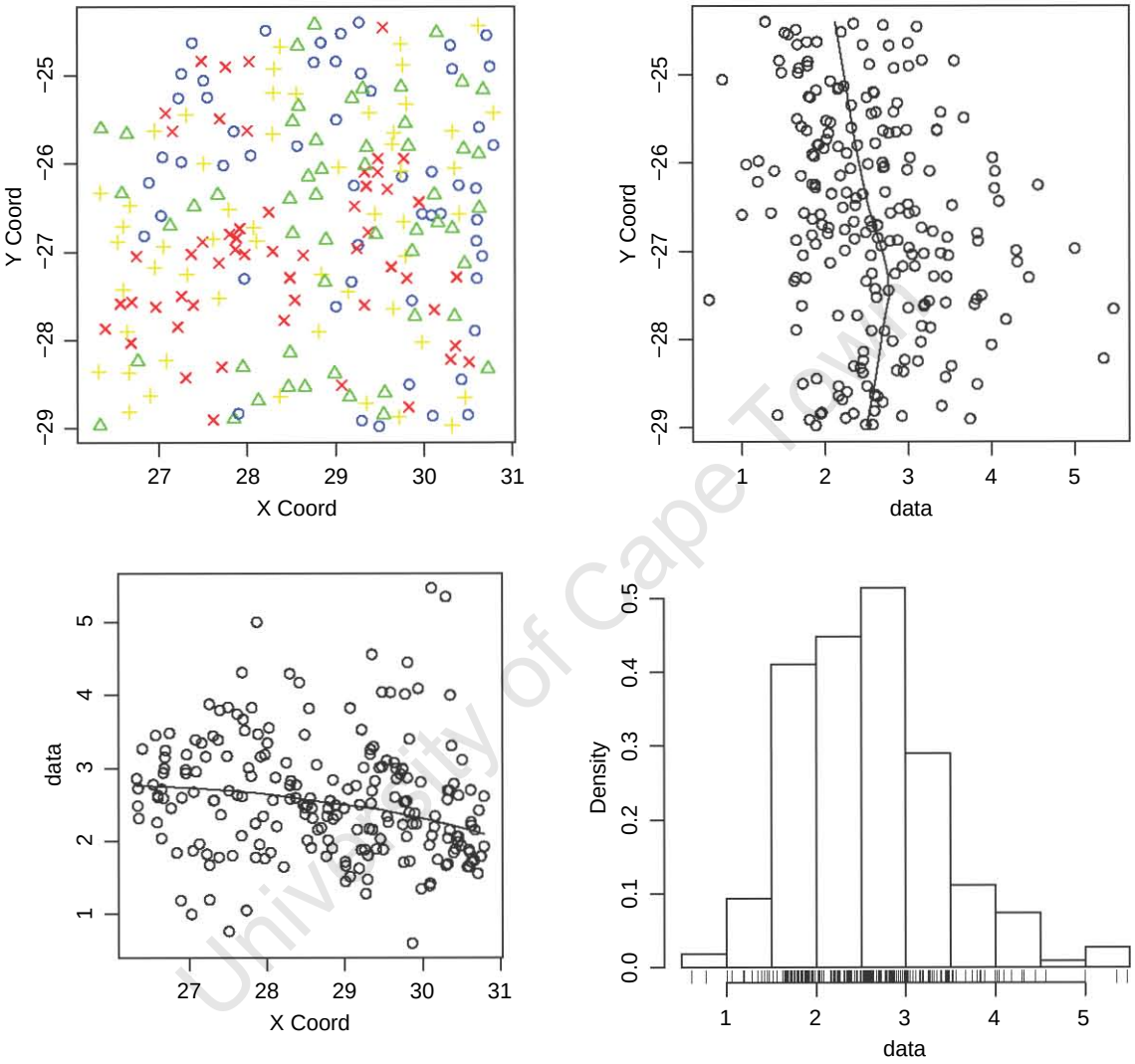


Figure 4.15: Upper-left panel shows a map of the quartiles of log(Fe) the log transformed iron levels, the upper-right and lower-left panels show log(Fe) plotted against the x and y coordinates respectively. The diagram in the lower-right panel shows the histogram of log(Fe).

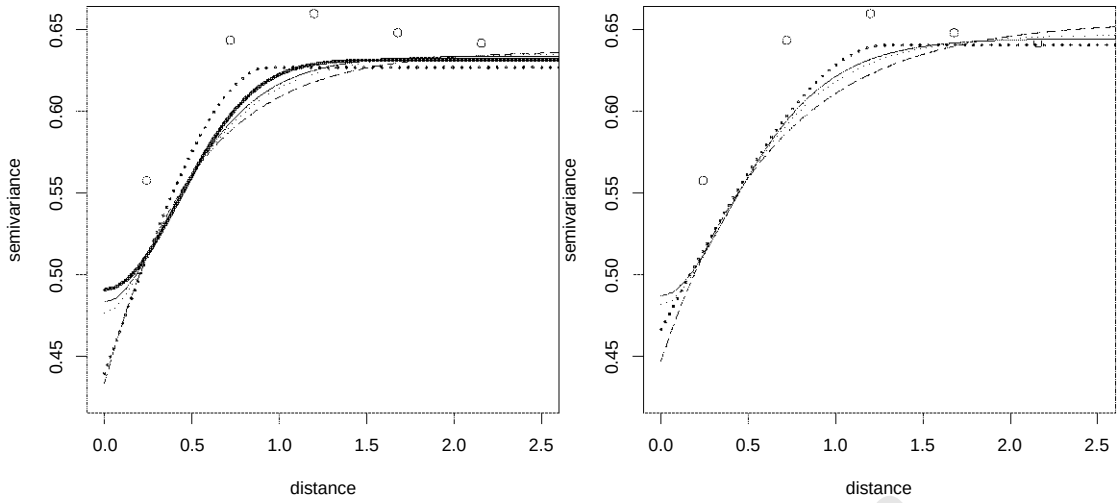
4.2.3 Model fitting

The spatial correlation structure for the log(Fe) data was modelled using three Matérn covariance functions with $\kappa = 0.5$, the exponential covariance, $\kappa = 1.5$ and $\kappa = 2.5$ and also the Gaussian and spherical covariance functions. Since no indication of any trend was found in the EDA, ordinary kriging models were fitted. Fitting of the models was done via ML and REML estimation. Table 4.7 gives a summary of the parameter estimates, the maximized log-likelihood value plus the information criteria, AIC and BIC, for the fitted ordinary kriging models. Figure 4.16 shows plots of the fitted semivariogram models together with the empirical semivariogram of the log(Fe) data.

Table 4.7: Estimates of the parameters of various ordinary kriging models of the log(Fe) data fitted with varying covariance functions together with the maximized log-likelihood values and the AIC and BIC.

Estimator	Covariance	κ	$\hat{\beta}$	$\hat{\tau}^2$	$\hat{\sigma}_0^2$	$\hat{\phi}$	L	AIC	BIC
ML	Exponential	-	2.5080	0.4340	0.2031	0.5085	-246.4134	500.8268	514.2907
	Matérn	1.5	2.5186	0.4766	0.1582	0.2813	-246.4293	500.8585	514.3224
		2.5	2.5223	0.4835	0.1504	0.2106	-246.4391	500.8782	514.3421
	Spherical	-	2.5379	0.4397	0.1871	0.9339	-246.8135	501.6270	515.0909
	Gaussian	-	2.5298	0.4909	0.1406	0.6034	-246.4829	500.9658	514.4297
REML	Exponential	-	2.4907	0.4474	0.2080	0.6494	-244.8845	497.7690	511.2329
	Matérn	1.5	2.5111	0.4817	0.1650	0.3138	-245.0273	498.0546	511.5185
		2.5	2.5169	0.4872	0.1572	0.2287	-245.0790	498.1581	511.6220
	Spherical	-	2.5277	0.4665	0.1740	1.2882	-245.3322	498.6645	512.1284
	Gaussian	-	2.5268	0.4937	0.1464	0.6374	-245.2094	498.4188	511.8827

It can be seen from Table 4.7 and Figure 4.16 that the 5 models fitted using ML estimation do not differ greatly especially with regards to estimates of the mean β, the nugget τ^2 and the partial sill σ_0^2. The fitted models can also be seen to have maximized log-likelihood values that are very close. From Figure 4.16, left panel, it can be seen that the fitted semivariogram models are almost indistinguishable from each other, at least visually. Inspection of the parameters of the models fitted by REML estimation reveals that the estimates of β and τ^2 do not differ greatly and it can be seen from the values of the maximized log likelihood values of the five fitted models that the resultant fitted models are similar. Furthermore, inspection of the semivariogram models fitted via REML, right panel of Figure 4.16, reveals the close similarity of the five fitted semivariograms.



(a) ML (b) REML

Figure 4.16: Comparison of various fitted semivariogram models together with the empirical semivariogram. The dashed lines correspond to fits with the exponential, dotted line to a Matérn with $\kappa = 1.5$, the solid line to a Matérn with $\kappa = 2.5$, heavy dotted line to a spherical and heavy solid line to a Gaussian. The circles correspond to the empirical semivariogram. Panels show (a) ML model results and (b) REML model results.

From Table 4.7 it is apparent that under both ML and REML estimation of the parameters, the kriging models fitted with the exponential covariance model,

$$\tau^2 + \sigma_0^2 \exp^{(-h/\phi)},$$

with AIC values of 500.8268 and 497.7690 for the model with $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$ respectively, are marginally better than the other models. Since the two models fitted with $\hat{\boldsymbol{\theta}}_{ML}$ and $\hat{\boldsymbol{\theta}}_{REML}$ are not very different in term the information criterion, AIC and BIC, and considering the fact that the differences between ML and REML estimation usually lead to only small differences in many applications (Diggle and Riberio, 2002) it was decided to use the kriging model with an exponential covariance function and $\hat{\boldsymbol{\theta}}_{ML}$ for the investigations conducted here. The profile likelihoods of the parameters for the kriging model with $\hat{\boldsymbol{\theta}}_{ML}$ are shown in Figure 4.17. The figure indicates that each of the parameters was estimated with reasonable precision as can be seen by the well defined maxima of each profile likelihood. The approximate 95% confidence intervals of the parameters which were obtained from the profile likelihoods, are displayed in Table 4.8.

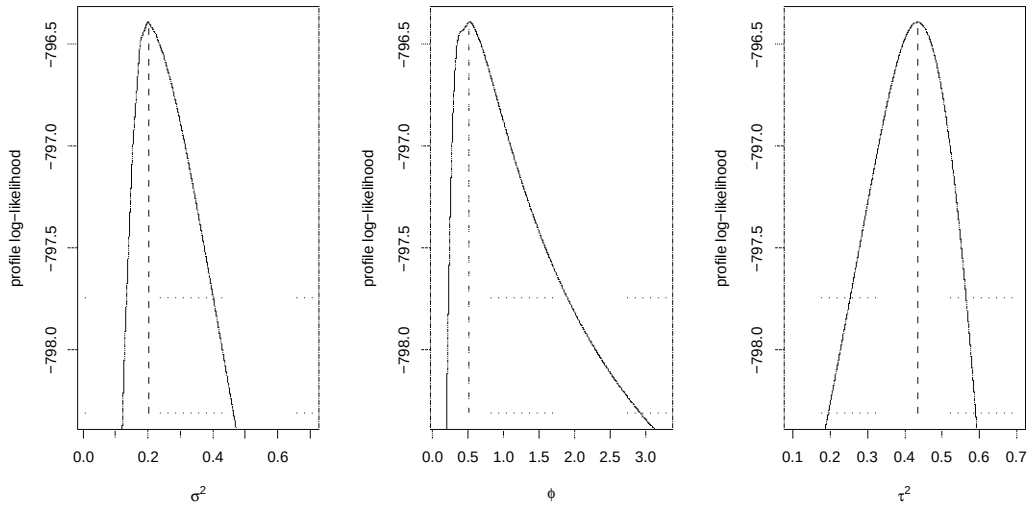


Figure 4.17: Profile likelihoods of σ_0^2 (left panel), ϕ (middle panel) and τ^2 (right panel) for the ordinary kriging model with an exponential function fitted using ML. The two horizontal lines on each of the plots define approximate 90% and 95% confidence intervals of each of the parameters, based on the asymptotic $\frac{1}{2}\chi^2(1)$-distribution of the log likelihood ratio.

Table 4.8: Estimates of the variance parameters and their approximate 95% confidence bounds, in parenthesis, for the ordinary kriging model with an exponential covariance function fitted via ML estimation.

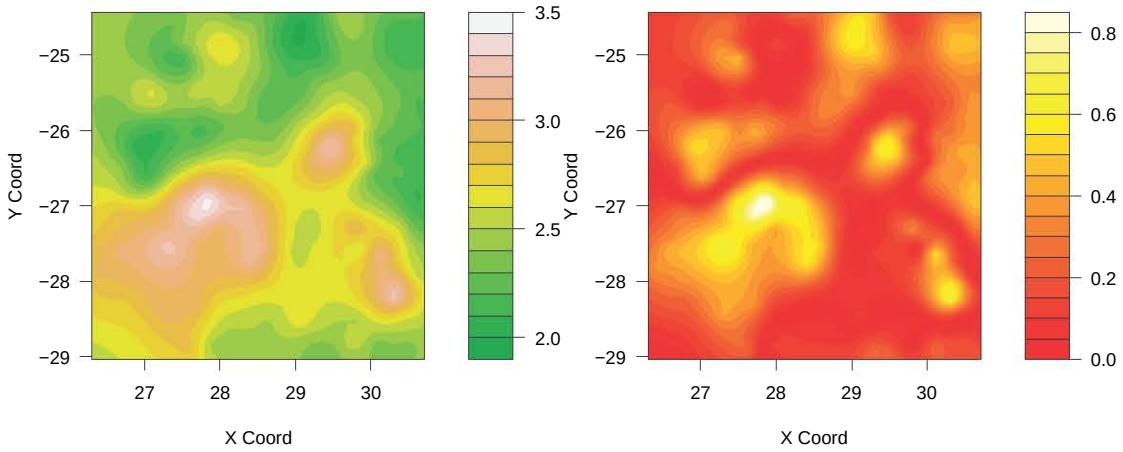
	Parameters		
	τ^2	σ_0^2	ϕ
Estimates	0.4340	0.2031	0.5085
95 % CI	(0.2,0.58)	(0.12,0.46)	(0.2,2.9)

4.2.4 Kriging and variance estimation

A grid of 2115 prediction locations specified by the set

$$S = \{\mathbf{x}_i = (u, v) : u = 26.30708{+}0.1k, k = 0, \ldots, 44; v = -29.03209{+}0.1l, l = 0, \ldots, 46\}$$

was set up in the study area. Predictions of the log(Fe) values at each of these locations were then made using the selected kriging model, that is the kriging model with an exponential covariance function fitted via ML estimation. Figure 4.18, right panel, shows the kriging surface of the predicted log(Fe) concentrations and the left panel shows associated kriging standard deviations.



(a) Predictions (b) Standard deviations

Figure 4.18: (a) Ordinary kriging surface obtained using the kriging model fitted with an exponential covariance function and (b) associated standard errors of the predictions.

Next the kriging variance at each of the prediction locations was calculated as in the Meuse data example using all the 5 kriging variance estimators, $m_1(\hat{\boldsymbol{\theta}})$, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$, $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$. The bootstrap estimates, $\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{CBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{UBS}$ were computed over 5000 iterations of their respective algorithms with the range parameter ϕ estimates in the $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ algorithm being constrained to $0.2 \leq \hat{\phi} \leq 2.90$, the approximate 95% confidence interval of the parameter as obtained from the profile likelihood and given in Table 4.8. Table 4.9 gives the summary statistics of the kriging variance estimates at the prediction 2115 locations obtained via the different kriging variance estimators as does Figure 4.19. Figure 4.20 gives a visual representation of the kriging estimates obtained by the various estimators as surfaces.

Table 4.9: Summary statistics of the kriging variance values at the 2115 prediction locations obtained using the various kriging variance estimators.

	KRIGING VARIANCE ESTIMATORS				
Summary Statistics	$m_1(\hat{\boldsymbol{\theta}}_{ML})$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{KH}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{PR}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{CBS}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{UBS}$
Min	0.5109	0.5180	0.5252	0.4958	0.5092
1st Quartile	0.5440	0.5508	0.5575	0.5415	0.5499
Median	0.5534	0.5595	0.5658	0.5541	0.5612
Mean	0.5574	0.5635	0.5695	0.5575	0.5643
3rd Quartile	0.5664	0.5717	0.5772	0.5703	0.5751
Max	0.6440	0.6476	0.6511	0.6671	0.6561
Standard deviation	0.0209	0.0201	0.0194	0.0237	0.0221

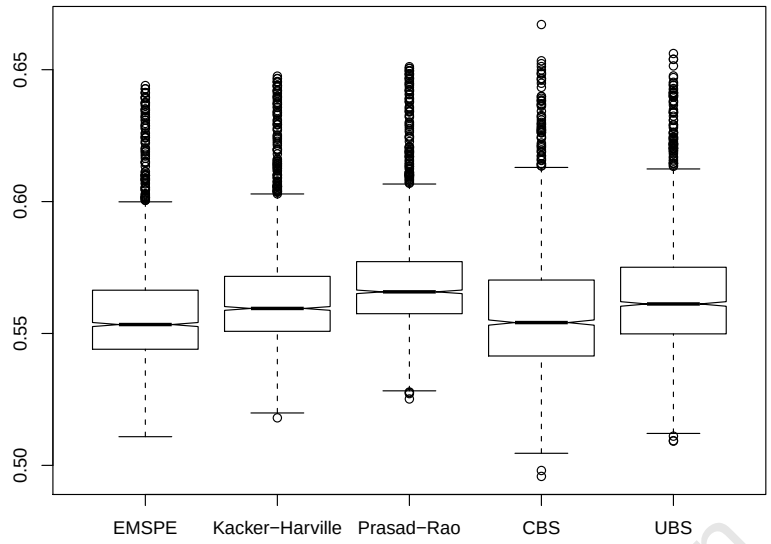


Figure 4.19: Notched box and whisker plot of the kriging variance estimates at the 3103 prediction locations obtained for the 5 kriging variance estimators: South African data.

It is clear from Table 4.9 and Figure 4.19 that the qualitative properties of the estimators observed in the simulation experiment also hold for this real data set. That is, $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ produced the highest estimates on average, $m_1(\hat{\boldsymbol{\theta}})$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ produced the smallest estimates and these estimates were relatively similar to each other and $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ produced estimates that were also very similar to each other.

Examination of Figure 4.20 reveals surfaces for $\sigma_0^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ that appear to be smooth as compared to the surface of the kriging variance estimates obtained for $m_1(\hat{\boldsymbol{\theta}})$ whilst $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ produce visibly rougher surfaces. It is interesting to note that these patterns were also observed in the previous application described in Section 4.1.

Table 4.10 shows the 10 points with the highest kriging variance estimates under each estimator, as well as the estimates. From this table it is apparent that the analytic estimators, $m_1(\hat{\boldsymbol{\theta}})$, $\sigma_0^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma_0^2(\hat{\boldsymbol{\theta}})_{PR}$, have the same set and ordering of 10 points with all these estimators identifying point (26.30708, -24.43209) as having the highest kriging variance. This point is located on the edge of the study area in the upper-left corner. The bootstrap estimators $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ identify points that are different from the each other and from the analytic estimators and unlike in the previous example it can be seen that $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ does not identify the same set of locations as $m_1(\hat{\boldsymbol{\theta}})$, $\sigma_0^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma_0^2(\hat{\boldsymbol{\theta}})_{PR}$. Figure 4.21 shows the locations of the set of 10 points with the highest kriging variance identified by the various estimators. The points identified by the various estimators lie in the same region, that is, at the edge of the figure. This may be an indication of an edge effect.

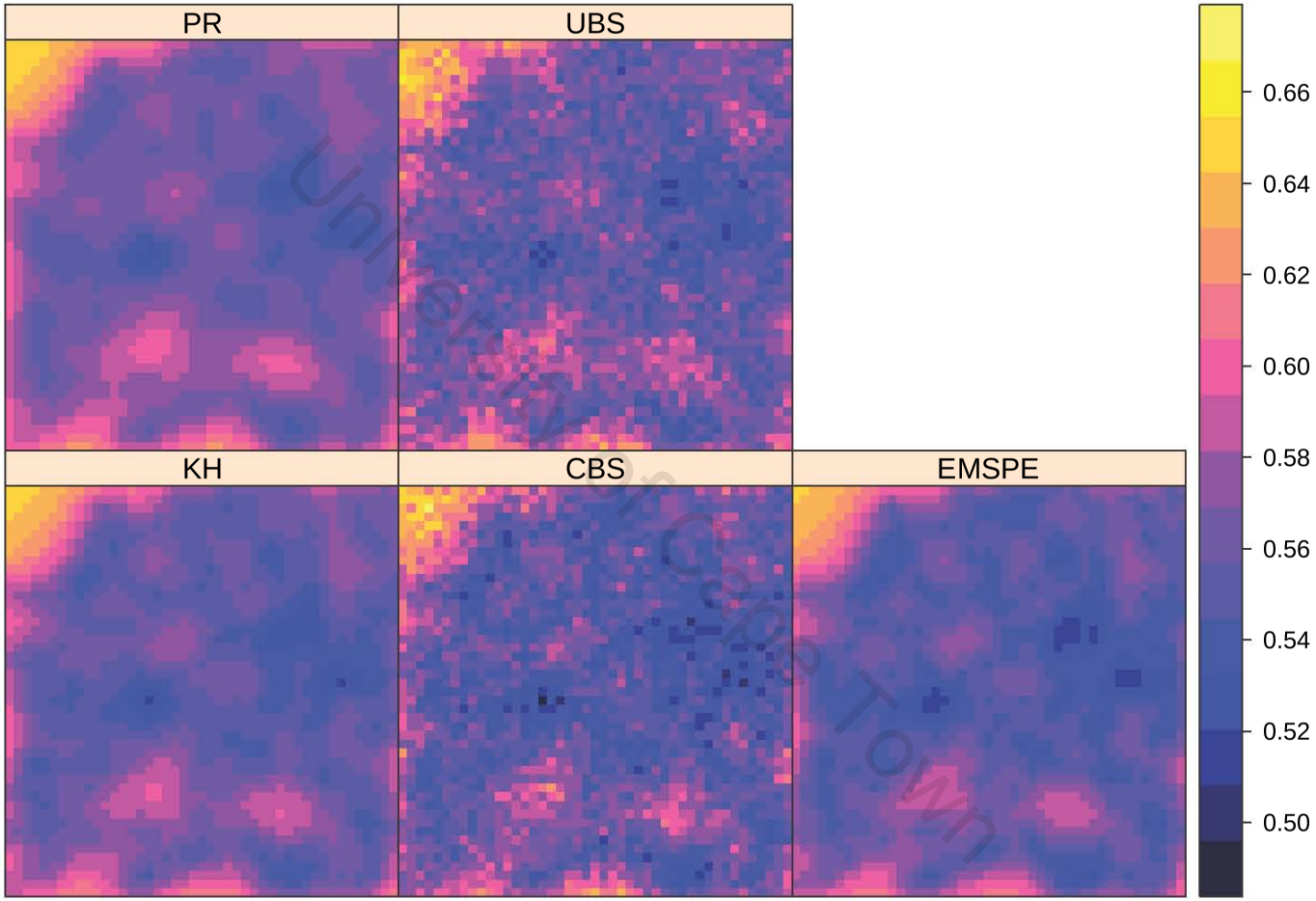


Figure 4.20: Maps showing surfaces of kriging variance estimates obtained using various estimation methods under the kriging model with an exponential covariance function and $\hat{\boldsymbol{\theta}}_{ML}$.

Table 4.10: Top ten prediction locations and estimates with the highest kriging variance as identified by the various estimators. Locations given in parenthesis and variance estimates in bold.

$m_1(\hat{\boldsymbol{\theta}}_{ML})$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{KH}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{PR}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{CBS}$	$\sigma^2(\hat{\boldsymbol{\theta}}_{ML})_{UBS}$
(26.30708, -24.43209) **0.6440**	(26.30708, -24.43209) **0.6476**	(26.30708, -24.43209)**0.6511**	(26.60708, -24.63209) **0.6671**	(26.30708 -24.83209) **0.6561**
(26.30708, -24.53209) **0.6428**	(26.30708, -24.53209) **0.6467**	(26.30708, -24.53209) **0.6505**	(26.50708, -24.63209) **0.6534**	(26.80708 -24.53209) **0.6540**
(26.40708, -24.43209) **0.6427**	(26.40708, -24.43209) **0.6466**	(26.40708, -24.43209) **0.6504**	(26.50708, -24.73209) **0.6523**	(26.30708 -24.53209) **0.6539**
(26.30708, -24.63209) **0.6414**	(26.30708, -24.63209) **0.6455**	(26.30708, -24.63209) **0.6497**	(26.50708, -24.93209) **0.6512**	(26.50708 -24.83209) **0.6514**
(26.40708, -24.53209) **0.6413**	(26.40708, -24.53209) **0.6454**	(26.40708, -24.53209) **0.6496**	(26.70708, -24.83209) **0.6499**	(26.70708 -24.63209) **0.6476**
(26.50708, -24.43209) **0.6411**	(26.50708, -24.43209) **0.6452**	(26.50708, -24.43209) **0.6494**	(26.30708, -24.43209) **0.6489**	(26.30708 -24.63209) **0.6470**
(26.40708, -24.63209) **0.6396**	(26.40708, -24.63209) **0.6440**	(26.40708, -24.63209) **0.6485**	(26.60708, -24.53209) **0.6485**	(26.30708 -24.73209) **0.6452**
(26.30708, -24.73209) **0.6396**	(26.30708, -24.73209) **0.6440**	(26.30708, -24.73209) **0.6484**	(26.80708, -24.53209) **0.6483**	(26.40708 -24.83209) **0.6448**
(26.50708, -24.53209) **0.6394**	(26.50708, -24.53209) **0.6438**	(26.50708, -24.53209) **0.6483**	(26.60708, -24.83209) **0.6467**	(26.70708 -25.03209) **0.6441**
(26.60708, -24.43209) **0.6390**	(26.60708, -24.43209) **0.6434**	(26.60708, -24.43209) **0.6478**	(26.30708, -24.53209) **0.6466**	(26.40708 -24.93209) **0.6441**

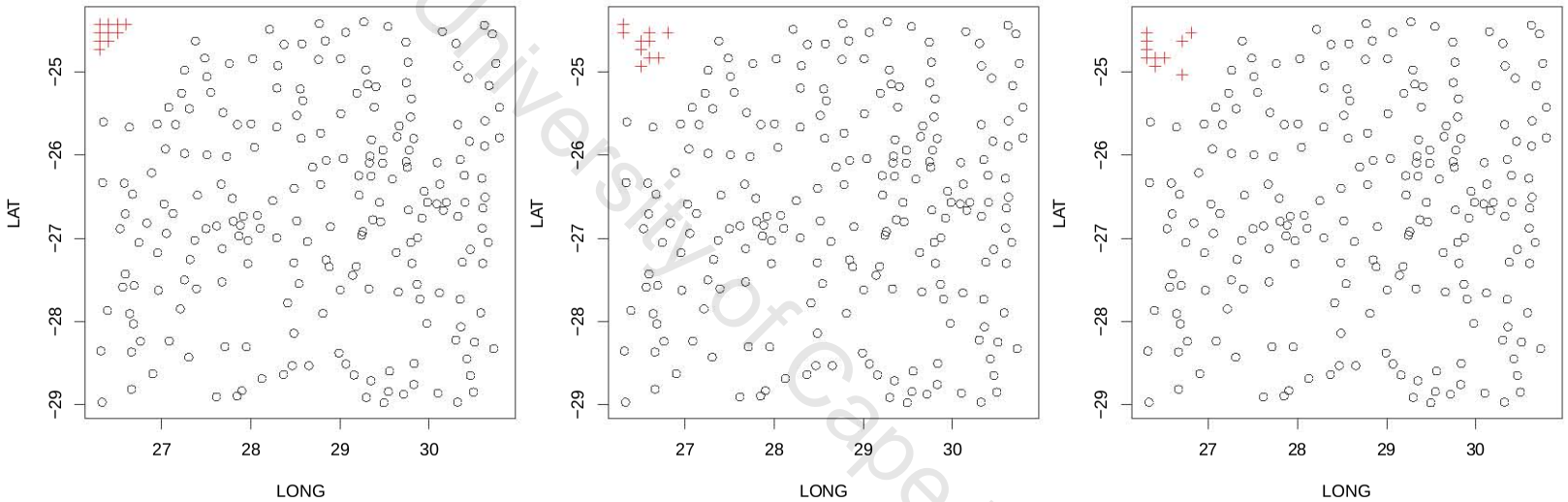


Figure 4.21: Circles show sample locations and crosses show the 10 locations which have the highest kriging variance under (a) the EMSPE, Kacker-Harville and Prasad-Rao estimators, (b) the conditional bootstrap estimator and (c) the unconditional bootstrap estimator.

4.3 Summary

Though the accuracy of each of the estimators can not be quantified as in the simulation experiment, it can be seen from both applications that the relationships between the estimators observed in the simulation experiment, such as the EMSPE and conditional bootstrap estimates being similar, also apply to real data sets. It can also be seen that each of the estimators are easily implemented even though the data sets may be large. All the estimators can also be seen to be applicable for a large variety of kriging models, such as the ordinary kriging models as in the South African data set and universal kriging models as in the Meuse data set. Though optimal designs in their truest sense have not been explored, it can be seen from both applications how the estimators may be used to identify an area of interest for sampling.

Chapter 5

Conclusions and Future Directions

5.1 Introduction

In this study various kriging variance estimators were explored and the performance investigated with the main aim of finding the most optimal estimator. This chapter evaluates the research objectives that were set out in the beginning of the study so as to ascertain whether they have been met. The objectives of the study as set out in Chapter 1 are listed here for reference. These were as follows:

1. The investigation of the performance of the empirical mean squared prediction error (EMSPE) in estimating the mean squared prediction error (MSPE) of the empirical best linear unbiased predictor (EBLUP).

2. The investigation and development of alternative methods to the EMSPE for estimating the MSPE of the EBLUP and the evaluation of the performance of these alternative estimators.

3. The development of portable programs in R (R Development Core Team, 2010) that provide the user with code for calculating the various estimators of the MSPE of the EBLUP.

4. The illustration and testing of the flexibility of the alternative methods of estimating the MSPE of the EBLUP on real data sets.

5. The examination of how sampling designs based on the the different kriging variance estimators as the basis of design criteria in optimal sampling design differ from each other.

Insights that arise from this study are also discussed and recommendations based on those insights are also made. Limitations of the study are briefly addressed and finally possible avenues for future research are highlighted.

5.2 Evaluation of Objectives

- **Objective 1**
  The performance of the EMPSE, denoted $m_1(\hat{\boldsymbol{\theta}})$, in estimating the MSPE of the EBLUP, $m_2(\boldsymbol{\theta})$, has been investigated methodically from both a theoretical and numerical perspective. In Chapter 2, Section 2.4, the reasons as to why $m_1(\hat{\boldsymbol{\theta}})$ is a biased estimator of $m_2(\boldsymbol{\theta})$ were set out and the direction of the bias was also noted. From the discussion in this section it was found that $m_1(\hat{\boldsymbol{\theta}})$ underestimates $m_2(\boldsymbol{\theta})$ as it fails to take into account the additional variability incurred by the EBLUP due to the covariance parameters, $\boldsymbol{\theta}$, being unknown and hence being estimated.

  The simulation experiment in Chapter 3 further showed that the bias of $m_1(\hat{\boldsymbol{\theta}})$ in estimating $m_2(\boldsymbol{\theta})$ can be quite large. From this experiment it was observed that this bias is greatest when the spatial correlation is strong, and less pronounced when the spatial correlation is weak. From the experiment it was also noted that, in comparable situations, the bias of $m_1(\hat{\boldsymbol{\theta}})$ tends to be larger when maximum likelihood (ML) estimates of $\boldsymbol{\theta}$, namely $\hat{\boldsymbol{\theta}}_{ML}$, are used in $m_1(\hat{\boldsymbol{\theta}})$ than when restricted maximum likelihood (REML) estimates, namely $\hat{\boldsymbol{\theta}}_{REML}$, are used.

- **Objective 2**
  In Chapter 2 existing alternative methods to the use of $m_1(\hat{\boldsymbol{\theta}})$ for estimating $m_2(\boldsymbol{\theta})$ were discussed and the development of these methods was thoroughly explored. Methods which appeared promising were identified. These included analytic forms of the Kacker-Harville and the Prasad-Rao estimators and parametric bootstrapping. In addition, a novel parametric bootstrapping algorithm for estimating $m_2(\boldsymbol{\theta})$, the conditional bootstrapping algorithm, was developed and also a slight detail, the constraining of the estimates of the range parameter ϕ from the samples generated in the unconditional bootstrap algorithm, was highlighted. As noted in Section 2.4.2.1, constraining the range is essential in ensuring that the bootstrapped kriging models are representative of the kriging model under study and hence the estimates produced by $\sigma_0^2(\hat{\boldsymbol{\theta}})_{UBS}$ are sensible.

  The estimators identified as being the most suitable alternatives to $m_1(\hat{\boldsymbol{\theta}})$, that is the Kacker-Harville $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$, the Prasad-Rao $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$, the unconditional bootstrap $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ and the conditional bootstrap estimators $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$ were then tested in the numerical experiment described in Chapter 3, in settings identical to those in which $m_1(\hat{\boldsymbol{\theta}})$ was tested, and the performance of the estimators recorded.

  From this experiment, it was observed that the performance of the estimator based on the conditional bootstrapping algorithm, $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$, was no better than $m_1(\hat{\boldsymbol{\theta}})$. From these experiments its was also found that the Prasad-Rao estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ is generally the most optimal estimator though its performance can be poor when the spatial correlation is weak. It was how-

ever also noted that with $\hat{\boldsymbol{\theta}}_{REML}$ as an estimator of $\boldsymbol{\theta}$ the Prasad-Rao estimator always produces estimates with very little bias no matter what the degree of the strength of the spatial correlation. Finally its was noted that the Kacker-Harville estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and the estimator based on an unconditional bootstrapping algorithm, $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$, perform more or less similarly. However, with $\hat{\boldsymbol{\theta}}_{REML}$ and when the spatial correlation is weak, $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ outperforms $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and in fact is the best estimator as it also outperforms $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$.

- **Objective 3**
  Programs in R (R Development Core Team, 2010), a widely used and very powerful statistical programming language and environment, were developed for calculating the various estimators investigated. These programs were linked to the R geostatistical package **geoR** (Riberio and Diggle, 2001) making them useful and accessible to a large number of geostatistical analysts. The programs were also completely documented so that they can be used with ease by anyone familiar with R and **geoR**.

- **Objective 4**
  The flexibility of the alternative estimators of $m_2(\boldsymbol{\theta})$ and the programs implementing them were demonstrated on two data sets, a data set widely investigated by other researchers and a local, South African data set, in Chapter 4. It was shown that these estimators and programs can be seamlessly incorporated into any routine and non-routine geostatistical analysis carried out in R using the geostatistical package **geoR** to calculate the various estimators.

- **Objective 5**
  In Chapter 4, using the two data sets mentioned above, we explored how the uses of the different estimators studied here may influence decisions in optimal sampling design, namely the decision as to where future samples should be taken. It was observed that $m_1(\hat{\boldsymbol{\theta}})$, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ identify the same locations as sites for future sampling. This is to be expected as these estimators are closely related. This finding means that when the ultimate goal of an analysis is to identify future sampling sites any of the three estimators, namely $m_1(\hat{\boldsymbol{\theta}})$, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$ and $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$, may be used as they will lead to the same decision. It was also observed that the conditional bootstrap estimator, $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$, may at times identify the same sites as those identified by the analytic estimators as future sampling sites, but this is not always the case. Finally it was also observed that the $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ usually identifies mostly different sites as future observation locations from $m_1(\hat{\boldsymbol{\theta}})$, $\sigma^2(\hat{\boldsymbol{\theta}})_{KH}$, $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$, and $\sigma^2(\hat{\boldsymbol{\theta}})_{CBS}$.

5.3 Limitations

Before drawing conclusions some of the most important limitations of this study need to be stated so as to put the conclusions into perspective. These limitations relate

mainly to the simulation experiment of Chapter 3. As noted earlier, there are many factors that influence the performance of the various kriging variance estimators (Zimmerman and Cressie, 1992). These include, in no order of importance, the following:

- Strength of the spatial correlation
- Type of covariance function employed
- Absence or presence of a trend and its degree
- Ratio of nugget to variance
- Spatial configuration of data and prediction locations
- Properties of the estimator of $\boldsymbol{\theta}$ employed
- Number of observations
- Density of the observation locations
- Size and shape of the observation domain

All these factors could not possibly be accounted for in the simulation experiment though every effort was made to make the experiment as comprehensive as possible.

5.4 Conclusions

In this section the main conclusions that can be drawn from this research are summarized as follows.

- *The EMSPE is an unsatisfactory estimator of the MSPE of the EBLUP.*
  In all cases studied here it was clearly seen that the EMSPE underestimates the MSPE of the EBLUP no matter what the degree of spatial correlation. It was also seen that in no situation will the EMSPE lead to the least biased estimates out of all the estimators studied here.

- *The Prasad-Rao estimator provides a real alternative to the EMSPE for estimating the MSPE of the EBLUP.*
  In most of the situations studied here the Prasad-Rao estimator was observed to provide the least biased best estimates of the MSPE of the EBLUP. Even in situations where the estimator may perform poorly, that is when the spatial correlation is weak, it was seen that the use of the REML estimator of $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}_{REML}$, instead of the ML estimator, $\hat{\boldsymbol{\theta}}_{ML}$, will result in estimates with satisfactory precision.

- *For any estimator of the MSPE of the EBLUP the use of $\hat{\boldsymbol{\theta}}_{REML}$ instead of $\hat{\boldsymbol{\theta}}_{ML}$ as an estimator of $\boldsymbol{\theta}$ will generally lead to less biased estimates.*
  Though under any situation all the estimators will exhibit some kind of bias, it has been observed that this bias can be reduced by using $\hat{\boldsymbol{\theta}}_{REML}$ instead of $\hat{\boldsymbol{\theta}}_{ML}$ in the calculation of the estimators.

- *Implementation of the alternative of estimators of the MSPE of the EBLUP is straight forward.*
  It has been seen from the applications in Chapter 4 that the estimation of the MSPE of the EBLUP using the alternative estimators is possible for practical problems involving large numbers of observation and prediction locations and different kriging models. It was also seen that the application of these methods to real data sets does not require any highly specialized algebra and programming.

5.5 Recommendations

- The use of the EMSPE $m_1(\hat{\boldsymbol{\theta}})$ should be avoided as it exhibits a large bias. However if the estimator is to be used it should be calculated using $\hat{\boldsymbol{\theta}}_{REML}$ as an estimator of $\boldsymbol{\theta}$ instead of $\hat{\boldsymbol{\theta}}_{ML}$ to mitigate the bias of the estimator.

- The Prasad-Rao estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ should be employed when the spatial correlation is moderate to strong.

- When $\boldsymbol{\theta}$ is estimated by $\hat{\boldsymbol{\theta}}_{ML}$ and the spatial correlation is weak, the unconditional bootstrap estimator $\sigma^2(\hat{\boldsymbol{\theta}})_{UBS}$ should be employed to estimate $m_2(\boldsymbol{\theta})$ but if $\boldsymbol{\theta}$ is estimated by $\hat{\boldsymbol{\theta}}_{REML}$ then $\sigma^2(\hat{\boldsymbol{\theta}})_{PR}$ may be used.

5.6 Further research

- Firstly the simulation experiment conducted in Chapter 3 could be extended to account for more of the factors noted under Section 5.3 as having an influence on the performance of estimators of the MSPE of the EBLUP, that is, it should include various covariance functions, different degrees of trend, various nugget to variance ratios, various estimators of $\boldsymbol{\theta}$, different sizes and shapes of the observation domain, different sampling densities and number of observations.

- The development of other bootstrapping algorithms for the estimation the MSPE of the EBLUP also needs further exploration and the performance of the estimators arising from these algorithms need to be investigated.

- Though the bootstrap versions of the Prasad-Rao and Kacker-Harville estimators suggested by Wang and Wall (2003) seem rather contrived it would be

interesting to explore how these estimators compare to their analytic counterparts and to the other estimators investigated in this study.

- The research carried out here could also be applied to the multivariate case as here only the univariate case is addressed.
- An in depth investigation on how the various estimators studied here can be used to be obtain optimal designs.

Bibliography

Abt, M. (1998). Approximating the mean squared prediction error in linear models under the family of exponential correlations. *Statistics Sinca*, **8**, 511-526.

Abt, M. (1999). Estimating the prediction mean squared error of a Gaussian stochastic process with exponential correlation structure. *Scandanavian Journal of Statistics*, **26**, 563-578.

Abt, M. and Welch, W.J. (1998). Fisher information and maximum likelihood estimation of covariance parameters in Gaussian stochastic processes. *Canadian Journal of Statistics*, **26**, 127-137.

Aptech (2008). *GAUSS mathematical and statistical system, system and graphics manual: version 9.0.* Aptech Systems, Maple Valley, WA.

Bernardo, M.C., Buck, R., Liu, L., Nazaret, W.A., Sacks, J. and Welch, W.J. (1992). Integrated circuit design optimization using a sequential strategy. In *IEEE Transactions on computer-aided design*, **11**, 361-372.

Booker, A.J., Dennis, J.E., Frank, P.D. Serafini, D.B., Torczon, V. and Trosset, M.W. (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, **17**, 113.

Burrough, P.A. and McDonnell, R.A. (1998). *Principles of Geographical Information Systems*, Oxford University Press.

Cahn, M.D., Hummel, J.W. and Brouer, B.H. (1994). Spatial analysis of soil fertility for site- specific crop management. *Soil Science Society of America Journal*, **58**, 1240-1248.

Carret, F. and Vellaron, A.J. (1992). Epidemiological mapping using the "kriging" method: application to an influenza-like illness epidemic in France. *American Journal of Epidemiology*, **135**, 1293-1300.

Chilès, J.P. and Delfiner, P. (1999). *Geostatistics : Modelling Spatail Uncertanity*. Wiley, New York.

Cressie, N.A.C. (1985). Fitting variogram models by weighted least squares. *Journal of the International Associstion of Mathmatical Geology*, **17**, 563-586.

Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, **22**, 239-252.

Cressie, N.A.C. (1993). *Statistics for Spatial Data* (revised edn.). John Wiley and Sons, New York.

Cressie, N., Gotway, C.A., and Grondona, M.O. (1990). Spatial prediction from networks. *Chemometrics and Intelligent Laboratory Systems*, **7**, 251-271.

Cressie, N. and Grondona, M.O. (1992). A Comparison of variogram estimation with covariogram estimation. *The Art of Statistical Science*, K.V. Mardia (ed.) Wiley, Chinster, 191-208.

Davis, J.C. (2002). *Statistical and data analysis in geology.* Wiley, New York.

DeVillers, S., Thiart, C. and Basson, N.C. (2010). Identification of sources of environmental lead in South Africa from surface soil geochemical maps. *Environmental Geochemistry and Health*, **32**, 451-459.

Den Hertog D., Kliejnen, J.P.C. and Siem. A.Y.D. (2006). The correct kriging variance estimated by bootstrapping. *Journal of the Operational Research Society*, **57**, 400-409.

Diggle, P.J. and Riberio, P.J. (2002). Bayesian inference in Gaussian model based geostatistics. *Geographical and Environmental Modelling*, **6**, 129-146.

Diggle, P.J. and Riberio, P.J. (2007). $Model-based\ Geostatistics$. Springer, New York.

Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap.* Chapman and Hall, New York.

Etman, L.F.P. (1994). *Design and analysis of computer experiments: The method of Sacks et al.*, Engineering Mechanics report WFW 94.098, Eindhoven University of Technology.

Fang, K., Li, R. and Sudjianto, A. (2006). *Design and Modelling for Computer Experiments*. Chapman and Hall/CRC, London.

Gilbert, P. (2006). *numDeriv: Accurate Numerical Derivatives.* R package version 2006.4-1. http://www.bank-banque-canada.ca/pgilbert

Gómez-Hernádez, J.J., Sahuquillo, A. and Capilla, J.E. (1997). Stochastic simulation of transmissivity fields conditional to both transmissivity and piezometric data: I. Theory. *Journal of Hydrology*, **203**, 162-174.

González-Manteiga, W., Lombardía, M.J., Molina, I., Morales, D. and Santamaría (2008). Analytic and bootstrap approximations of prediction errors under a multivariate Fay-Herriot model. *Computational Statistics and Data Analysis*, **52**, 5242-5252.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation.* Oxford University Press, Oxford.

Goovaerts, P. (1999). Geostatistics in soil science: state-of-the-art and perspectives. *Geoderma*, **89**, 1-45.

Goovaerts, P. (2001). Geostatistical modelling of uncertainty in soil science. *Geoderma*, **103**, 3-26

Hall, P. (1985). Resampling a coverage pattern. *Stochastic Processes and thier Applications*, **20**, 231-246.

Hall, P. and Jing B. (1996). On sample reuse methods for dependent data. *Journal of the Royal Statitical Society*, **58**, 727-737.

Harville, D.A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, **72**, 320-338.

Harville, D.A. (1985). Decomposition of prediction error. *Journal of the American Statistical Association*, **80**, 132-138.

Harville, D.A. and Jeske, D.R. (1992). Mean squared error of estimation or prediction under a general linear model. *Journel of the American Statistical Association*, **87**, 724-731.

Hohn, M.E. (1999). *Geostatistics and petroleum geology.* Dordrecht: Kluwer Academic Publishers, , 1999

Jones, D.R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journel of Global Optimization*, **13**, 455-492.

Jones, D.R, Schonlau, M. and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journel of Global Optimization*, **13**, 455-492.

Journel, A.G. (1986). Geostatistics: models and tools for the earth sciences. *Mathematical Geology*, **18**, 119-141.

Journel, A.G. and Huijbregts, C.J. (1978). *Mining Geostatistics*. Academic Press, New York.

Kacker, R.N. and Harville, D.A. (1984). Approximations for standard errors of estimators of fixed and random effects in mixed linear models. *Journel of the American Statistical Association*, **79**, 853-862.

Kintandis, P.K. and Vomoris, E.G. (1993). A geostatistical approach to the inverse problem of ground water modelling (steady state) and one dimensional simulations. *Water Resources Research*, **19**, 677-690.

Kleijnen, J.P.C. (1987). *Statistical Tools for Simulation Practitioners.* Marcel Decker, New York.

Kleijnen, J.P.C. (2009). Kriging metamodelling in simulation: a review. *Journal of the Operational Research Society*, **192**, 707-716.

Kleijnen, J.P.C. AND van Beers, W.C.M. (2004). Application-driven sequential designs for simulation experiments: kriging metamodelling. *Journal of the Operational Research Society*, **55**, 876-883.

Kolmogorov, A.N. (1941). Interpolation and extrapolation of stationary random sequences. *Izvestia Akademii Nauk SSSR, Seria Matematicheskiia*, **5**, 3-14.

Krige, D.G. (1951). A statistical approach to some basic mine evaluation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, **52**, 119-196.

Künsch, H.R. (1989). The jackknife and the bootstrap for general stationary observations. *Annals of Statistics*, **17**, 1217-1241.

Lahiri, S.N. (1999). Theocratical comparisons of block bootstrap methods. *Annals of Statistics*, **27**, 386-404.

Lahiri, S.N. (2000). *Resampling Methods for Dependent Data*. Springer, New York.

Lahiri, S.N., Lee, Y. and Cressie, N. (2002). On asymptotic distribution and asymptotic efficiency of least squares estimation of spatial variogram parameters. *Journal of Statistical Planning and Inference*, **103**, 65-85.

Martin, J.D. and Simpson (2005). Use of kriging models to approximate deterministic computer models. *AIAA Journal*, **43**, 853-863.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, **58**, 1246-1266.

Matheron, G. (1973). Intrinsic random functions and their applications. *Advances in Applied Probability*, **5**, 439-468.

Matheron, G. (1975). *Random Sets and Integral Geometry*, John Wiley and Sons, New York.

Matheron, G. (1989). *Estimating and Choosing*. Springer-Verlag, Berlin.

Pachepsky, Y. and Acock, B. (1998). Stochastic imaging of soil parameters to asses variability and uncertainty of crop yield estimates. *Geoderma*, **85**, 213-229.

Patterson, H.D. and Thompson, R. (1971). Recovery of interblock information when block sizes are unequal. *Biometrika*, **58**, 545-554.

Pebesma, E.J. (2004). Multivariable geostatistics in S: the gstat package. *Computers and Geosciences*, **30**, 683-691.

Pebesma, E.J. (2010). *The meuse data set: a tutorial for the gstat R package*.

Prasad, N.G.N. and Rao, J.N.K. (1986). *On the estimation of mean square error of small area predictors*. Technical Report 97, Carleton University, Laboratory for Research in Statistics and Probability, Ottawa.

Prasad, N.G.N. and Rao, J.N.K. (1990). On the estimation of the mean square error of small area predictors. *Journal of the American Statistical Association*, **85**, 163-171.

Philip, G.M. and Watson, D.F. (1986). Matheronian Geostatistics-Quo Vadis? *Mathematical Geology*, **18**, 93-115.

R Development Core Team (2010). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Riberio, P.J. Jr and Diggle, P. (2001). geoR: a package for geostatistical analysis. *R-News*, **2**, 14-18.

Riberio, P.J. Jr and Diggle, P. (2009). *The geoR package* (R manual).

Rikken, M.G.J. and Van Rijn, R.P.G. (1993). *Soil pollution with heavy metals - an inquiry into spatial variation, cost of mapping and the risk evaluation of copper, cadmium, lead and zinc in the floodplains of the Meuse west of Stein, the Netherlands.* Doctoraalveldwerkverslag, Dept. of Physical Geography, Utrecht University.

Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P. (1989). Design and analysis of computer experiments (with discussion). *Statistical Science*, **4**, 409-435.

Santer, T.J., Williams, B.J. and Notz, W.I. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York.

Sasena, M.J., Papalambros, P. and Goovaerts, P. (2002). Exploration of metamodelling sampling criteria for constrained global optimization. *Engineering Optimization* **34**, 263278.

Schabenberger, O. and Gotway, C.A. (2005). *Statistical Methods for Spatial Data Analysis*. Chapman and Hall, New York.

Schabenberger, O. and Pierce, F.J. (2002). *Contemporary Statistical Models for the Plant and Soil Sciences*. CRC Press, Boca Raton, FL.

Schott, J.R. (1997). *Matrix Analysis for Statistics*. Wiley, New York

Soil and Plant Analysis Council (2000). *Soil analysis: Handbook of Reference Methods*. Boca Raton: CRC Press.

Stein, A. (1999). *Interpolation of Spatial Data: some theory for kriging*. Springer, New York.

Stein, M. L. and Handcock, M. S. (1989). Some asymptotic properties of kriging when the covariance function is misspecified. *Mathematical Geology*, **21**, 171-190.

Todini, E. (2001). Influence of parameter estimation uncertainty in kriging: Part 1 - Theoretical development. *Hydrological and Earth System Sciences*, **5**, 215-223.

Todini, E. and Ferraresi, M. (1996). Influence of parameter estimation uncertainty in kriging. *Journal of Hydrology*, **175**, 555-566.

Todini, E., Pellegrini, F. and Mazzetti, C. (2001). Influence of parameter estimation uncertainty in kriging: Part 2 - Test and case study applications. *Hydrological and Earth Systems Sciences*, **5**, 225-232.

Wahba, G. (1990). *Spline models for observational data*. SIAM, Philadelphia.

Wang, F. and Wall, M.M. (2003). Incorporating parameter uncertainty into prediction intervals for spatial data modeled via a parametric variogram. *Journal of Agricultural, Biological and Environmental Statistics*, **8**, 296-309.

Webster, R. and Oliver, M. (2001). *Geostatistics for Environmental Scientists*. John Wiley and Sons, New York.

Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T. and Morris, M.D. (1992). Screening, predicting and computer experiments. *Technometrics*, **34**, 15-25.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, **41**, 434-449.

Wiener, N. (1949). *Extrapolation, Interpolation and Smoothing of Time Series*. MIT Press, Cambridge, MA.

Wold, H. (1938). *A Study in the Analysis of Stationary Time Series*, Almqvist and Wiksells, Uppsala.

Yin, J., Ng, S.H. and Ng, K.M. (2008). Kriging model with modified nugget effect for random simulation with heterogenous variances. In *Proceedings of IEEE International Conference on Industrial Engineering and Engineering Management*, 1714-1718.

Yin, J., Ng, S.H. and Ng, K.M. (2009). A study of the effects of parameter estimation on kriging model's prediction error in stochastic simulations. In *Proceedings of the 2009 Winter Simulation Conference*, Rossetti, M.D., Hill, R.R., Johansson, B. Dunkin, A. and Ingalls, R.G. (eds), 674-685.

Zhu, Z. and Stein, M.L. (2005). Spatial sampling design for parameter estimation of the covariance function. *Journal of Statistical Planning and Inference*, **134**, 583-603.

Zimmerman, D.L. (2006). Optimal network design for spatial prediction, covariance parameter estimation and empirical prediction. *Environmetrics*, **17**, 635-652.

Zimmerman, D.L. and Cressie, N. (1992). Mean squared prediction error in the spatial linear model with estimated covariance parameters. *Annals of the Institute of Statistical Mathematics*, **44**, 27-43.

Zimmerman, D.L. and Zimmerman M.B. (1991). A comparison of spatial semivariogram estimators and corresponding ordinary kriging predictors. *Technometrics*, **33**, 77-91.

Appendix A

R codes:

These functions were built in R 2.10.1 (R Development Core Team, 2010) and are composed mostly of functions from **geoR** 1.6-27 (Riberio and Diggle, 2001) and are thus dependent on this package. The bootstrapping functions `krige.uncond()` for bootstrap estimation of the kriging variance using unconditional simulation and `krige.cond()` which uses conditional simulation are strictly dependent on **geoR**. The functions `partial.deriv()` and `krige.approx()` though also dependent on **geoR** are also dependent on **numDeriv** (Gilbert, 2006) for the calculation of numerical derivatives. In addition the function `krige.approx()` requires the package **MASS** (Venables and Ripley, 2002) in order to calculate generalized inverses when `lik.method =''REML''`. The functions given in the last section `num.compare` and `num.solve()` have no practical use. The were used in conducting the simulation experiment and are only given here for completeness. Because they have no piratical use no documentation on them is provided.

A.1 Bootstrapping functions

A.1.1 Parametric bootstrapping of kriging variance; unconditional simulation

```
> krige.uncond <- function(geodata, coords, data, locations, cov.pars, cov.model =
+                          "matern", obj.model, mean = 0, nugget = 0, fix.nugget
+                          = FALSE, kappa = .5, fix.kappa = TRUE, lambda = 1,
+                          fix.lambda = TRUE, psiA = 0, fix.psiA = TRUE, psiR = 1,
+                          fix.psiR = TRUE, aniso.pars = NULL, lik.method = "ML",
+                          limits = pars.limits(), type.krige = "ok", trend.d =
+                          "cte", trend.l = "cte", micro.scale = 0, dist.epsilon =
+                          1e-10, output = output.control(messages = FALSE), nsim =
+                          1000, sim.method = "cholesky", con.level= 0.05, RF = TRUE,
+                          messages = FALSE, print.lik = TRUE)
+{
+     require(geoR)
```

```
+     if (missing(geodata) & missing(coords))
+         { stop ( "Need to specify data and coordinates") }
+     if (missing(geodata)) {
+         spatail.data <-cbind(coords,data)
+         geodata <- as.geodata(obj= spatail.data, coords.col=1:2, data.col = 3)
+                     n.obs <- nrow(geodata$coords) }
+     else { n.obs <- nrow(geodata$coords)}
+     if (missing(locations))
+            {  stop("need to specify prediction locations") }
+     else{
+           dimnames(geodata$coords) <- NULL
+           if (is.data.frame(locations) == TRUE)
+                 { locations <- as.matrix(locations) }
+           dimnames(locations) <- NULL
+           locs <- rbind(geodata$coords, locations)
+           dimnames(locs) <- NULL
+           locs <- unique(locs)
+           n.coords <- nrow(locs)
+           locations <- locs[(n.obs + 1) : n.coords,]
+           locations <- matrix(locations, ncol = 2, dimnames = NULL)
+           n.locs <- nrow(locations)
+       }
+     n.obs <- nrow(geodata$coords)
+     coords.data <- geodata$coords
+     dimnames(coords.data) <- NULL
+     if(missing(obj.model) == FALSE )
+         { cov.model <- obj.model$cov.model
+           cov.pars <- obj.model$cov.pars
+           nugget <- obj.model$tausq
+           mean <- obj.model$beta
+           lik.method <- obj.model$method.lik
+           kappa <- obj.model$kappa
+           lambda <- obj.model$lambda
+           aniso.pars <- obj.model$trend
+           trend.d <- obj.model$trend
+           lik.method <- obj.model$method.lik
+           check.mean <- as.matrix(mean, nrow = 1)
+           if(dim(check.mean)[[1]]>1) { mu <- sum(mean) }
+           else(mu <- mean)
+        }
+     if(missing(obj.model) == FALSE )
+         {
+           krige.settings <- krige.control(type.krige = type.krige, obj.model =
+                             obj.model, dist.epsilon = dist.epsilon, micro.scale
+                             = micro.scale) }
+     else {
+           if( type.krige == "sk" | type.krige == "SK")
+               { beta <- mean }
+           if(missing(obj.model) == TRUE )
+               { beta <- mean }
+           krige.settings  <- krige.control(type.krige = type.krige, trend.d =
+                               trend.d, trend.l = trend.l, cov.model = cov.model,
+                               cov.pars = c(cov.pars[1],cov.pars[2]), kappa =
+                               kappa, beta = beta, nugget = nugget, micro.scale
```

```
+                              = micro.scale, dist.epsilon = dist.epsilon ,
+                              aniso.pars = aniso.pars, lambda = lambda)
+          check.mean <- as.matrix(mean, nrow = 1)
+          if(dim(check.mean)[[1]]>1) { mu <- sum(mean) }
+          else(mu <- beta)
+      }
+    krige.original <- krige.conv(geodata = geodata, locations = locations,
+                        krige = krige.settings, output = output)
+    lapply.out <- lapply(1:nsim, function(i, n, grid, nsim, cov.model, cov.pars,
+                  aniso.pars, mean, nugget, kappa, lambda, method, RF, messages,
+                  geodata, trend, ini.cov.pars, fix.nugget, fix.kappa, fix.lambda,
+                  psiR, fix.psiR, limits, lik.method, type.krige, obj.model,
+                  dist.epsilon, micro.scale, locations, krige, output)
+       {
+          sim.grf <- grf(n = n.coords, grid = locs, nsim = nsim, cov.model =
+                     cov.model, cov.pars = cov.pars, aniso.pars = aniso.pars,
+                     mean = mu, nugget = nugget, kappa = kappa, lambda =
+                     lambda, method = sim.method, RF = RF, messages = messages)
+          sim.samples <- as.matrix(sim.grf$data)
+          sim.obs <- drop(as.matrix(sim.samples[ 1 : n.obs, ]))
+          temp.data <-as.geodata(cbind(coords.data, sim.obs))
+          sim.preds <- as.matrix(sim.samples[ (n.obs + 1) : n.coords, ])
+          lik.est <- likfit(geodata = temp.data, trend = trend.d, ini.cov.pars
+                     = cov.pars, nugget = nugget, fix.nugget = fix.nugget,
+                     kappa = kappa, fix.kappa = fix.kappa, lambda =lambda,
+                     fix.lambda = fix.lambda, psiA = psiA, fix.psiA = fix.psiA,
+                     psiR = psiR, fix.psiR = fix.psiR, cov.model = cov.model,
+                     limits = limits, lik.method = lik.method, messages =
+                     messages)
+          if(print.lik == TRUE )
+              {print(lik.est) ; print(i); flush.console () }
+          krige.set.sim <- krige.control(type.krige = type.krige, obj.model =
+                           lik.est, dist.epsilon = dist.epsilon, micro.scale
+                            = micro.scale)
+          est.cov.pars <- matrix(lik.est$cov.pars,nrow=1)
+          if(fix.nugget == FALSE)
+                { est.cov.pars <- cbind(lik.est$tausq, est.cov.pars) }
+          krige.sim <- krige.conv(geodata = temp.data, locations = locations, krige
+                       = krige.set.sim, output = output)
+          predictions <- as.matrix(krige.sim$predict)
+          COVS <- est.cov.pars
+          SPE <- (predictions - sim.preds)^2
+          out.boot <- list(SPE=SPE, COVS=COVS)
+    },
+       n = n.coords, grid = locs, nsim = 1, cov.model = cov.model, cov.pars =
+       cov.pars, aniso.pars = aniso.pars, mean = mean, nugget = nugget, kappa =
+       kappa, lambda = lambda, method = sim.method, RF = RF, messages = messages,
+       geodata = temp.data, trend = trend.d, ini.cov.pars= cov.pars, fix.nugget =
+       fix.nugget, fix.kappa = fix.kappa, fix.lambda = fix.lambda, psiR = psiR,
+       fix.psiR = fix.psiR, limits = limits, lik.method = lik.method, type.krige =
+       type.krige, obj.model = lik.est, dist.epsilon = dist.epsilon, micro.scale =
+       micro.scale, locations = locations, krige= krige.set.sim, output = output)
+
+  COVS <- sapply(lapply.out, function(x)x$COVS)
```

```
+   SPE <- sapply(lapply.out, function(x)x$SPE)
+   sd <- apply(SPE,1,sd)
+   MSPE<- as.vector(apply(SPE, 1, sum)/ nsim)
+   sq.std.MSPE <- as.matrix(apply((SPE - MSPE)^2, 1, sum)/(nsim - 1))
+   std.MSPE <- sqrt(sq.std.MSPE)/sqrt(nsim)
+   if( con.level==0.05 | missing(con.level)==TRUE) { z = 1.96 }
+   else if(con.level==0.10) { z = 1.64 }
+   lb <- MSPE - z * std.MSPE
+   ub <- MSPE + z * std.MSPE
+   bounds <- cbind(lb,ub)
+   colnames(bounds) <- c("lower","upper")
+   pred.errors <- cbind(krige.original$krige.var,MSPE)
+   colnames(pred.errors) <- c("plug.in","MSPE")
+   output <- list(coords = locations, predictions = krige.original$predict,
+             pred.errors = pred.errors, std.errors = std.MSPE, sd = sd, bounds =
+             bounds, est.pars = COVS)
+   return(output)
+}
```

A.1.2 Parametric bootstrapping of kriging variance; conditional simulation

```
>krige.cond  <-function(geodata, coords, data, locations, cov.pars, cov.model =
+                    "matern", obj.model, mean, nugget = 0, kappa = .5, lambda = 1,
+                    type.krige = "ok", trend.d = "cte", trend.l = "cte",
+                    aniso.pars = NULL, micro.scale = 0, dist.epsilon = 1e-10,
+                    output = output.control(messages = FALSE), nsim = 1000,
+                    signal = NULL, messages = FALSE)
+{
+    require(geoR) # Loading geoR package
+    if (missing(geodata) & missing(coords))
+        { stop ( "Need to specify data and coordinates") }
+    if (missing(geodata)) {
+        geodata <- list(coords = coords, data = data)
+                    n.obs <- ncol(geodata$coords) }
+    else { n.obs <- nrow(geodata$coords)}
+    if (missing(locations))
+           {  stop("need to specify prediction locations") }
+        n.obs <- length(geodata$coords[,1])
+        total.locs <- unique(rbind(geodata$coords,locations))
+        total.p <-length(total.locs[,1])
+        locations <- total.locs[(n.obs+1):total.p,]
+        row.names(locations) <-NULL
+        n.locs <-ncol(locations)
+    if(missing(obj.model) == FALSE )
+        { cov.model <- obj.model$cov.model
+          cov.pars <- obj.model$cov.pars
+          nugget <- obj.model$tausq
+          mean <- obj.model$beta
+          kappa <- obj.model$kappa
```

```
+          lambda <- obj.model$lambda
+          aniso.pars <-  obj.model$aniso.pars
+       }
+       # Default settings for arguement output.control;
+    if(missing(output)) {
+           output <- output.control(n.predictive = nsim, simulations.predictive
+                     = TRUE, mean.var = TRUE, sim.means = TRUE, sim.vars = TRUE,
+                     messages = messages, signal = signal) }
+     if(missing(obj.model) == FALSE )
+         {
+          krige.settings <- krige.control(type.krige = type.krige, obj.model =
+                              obj.model, dist.epsilon = dist.epsilon, micro.scale
+                              = micro.scale) }
+     else {
+           if( type.krige == "sk" | type.krige == "SK")
+               { beta <- mean }
+           if(missing(obj.model) == TRUE )
+              { beta <- NULL }
+           krige.settings  <- krige.control(type.krige = type.krige, trend.d =
+                              trend.d, trend.l = trend.l, cov.model = cov.model,
+                              cov.pars = c(cov.pars[1],cov.pars[2]), kappa =
+                              kappa, beta = beta, nugget = nugget, micro.scale
+                              = micro.scale, dist.epsilon = dist.epsilon ,
+                              aniso.pars = aniso.pars, lambda = lambda)
+       }
+   Ckrige <- krige.conv(geodata = geodata, locations = locations, output =
+              output, krige = krige.settings)
+   CPreds <-Ckrige$predict
+   CKrigevar <- Ckrige$krige.var
+   SPE <-  (CPreds - Ckrige$simulations)^2
+   MSPE <- as.vector(apply(SPE, 1, sum)/nsim)
+   errors <-sqrt(as.matrix(apply((SPE - MSPE)^2,1,sum))/(nsim -  1))
+   std.errors <- (errors/sqrt(nsim))
+   lb <- MSPE - 1.96 * std.errors
+   ub <- MSPE + 1.96 * std.errors
+   bounds <- cbind(lb,ub)
+   colnames(bounds) <- c("lower","upper")
+   pred.errors <- cbind(Ckrige$krige.var,MSPE)
+   colnames(pred.errors) <- c("plug.in","MSPE")
+   output <- list(coords = locations, predictions = Ckrige$predict, pred.errors =
+             pred.errors, std.errors = std.errors, bounds = bounds, sim.means =
+             Ckrige$sim.means, sim.vars = Ckrige$sim.vars)
+  return(output)
+}
```

A.2 Functions used in computing the Kacker-Harville and Prasad-Rao estimator

These functions depend on **numDeriv**, for computing numerical derivatives and **MASS** for computing generalized inverses. They are used to calculate kriging estimates that use the approximation by Kacker and Harville (1984) for the bias of the empirical mean squared prediction error (EMSPE). The function `partial.deriv()` is used to calculate first derivatives of matrixes, the Jacobian. These derivatives are needed in the calculation of the Kacker-Harville and Prasad-Rao kriging variance estimates. The function `krige.approx()` performs the actual calculation of the Kacker-Harville and Prasad-Rao variance estimates and thus depends on `partial.deriv()` as well as the **geoR** function `krige.conv()`.

A.2.1 Function for computing Jacobian for covariance matrixes

```
> partial.deriv <- function(cov.pars = c(sigma, phi), nugget = 0, obj,
+                           cov.model = "gaussian", deriv.method = "Richardson")
+ {
+    if(missing(obj))
+         { stop ( "Need to specify data and prediction locations") }
+    if (!is.matrix(obj) == FALSE) {obj <- as.matrix(obj)}
+    n.obj <- nrow(obj)
+    if(missing(cov.pars))
+        { stop ( "Need to specify vector of covariance parameters") }
+    if( nugget != 0 )
+       { cov.pars <- matrix(cov.pars, nrow = 1)
+         I <- matrix(0, nrow = n.obj, ncol = n.obj)
+         diag(I) <- 1
+         if(ncol(obj) == 1) { I <- 1 }
+     }
+    else{I <- 1 }
+    if((ncol(cov.pars) < 2) == TRUE || (ncol(cov.pars) > 2) == TRUE)
+               { stop ( "Argument cov.pars should be a vector of size 2" ) }
+    else(
+          {sigma <- cov.pars[1,1] ; phi <- cov.pars[1,2] } )
+    cov.fun <- function(phi, sigma, nugget, lags, cov.model)
+             {
+              cov.models = c("gaussian", "exponential", "spherical")
+              model <- match.arg(cov.model, cov.models)
+              cov.struc <- switch(model,
+                           gaussian = nugget * I + sigma * exp(-lags/phi)^2,
+                           exponential = nugget * I + sigma * exp(-lags/phi),
+                           spherical = ifelse( lags < phi,  nugget * I + sigma *
+                                      (1 - 1.5 * (lags/phi) + 0.5 * (lags/phi)^3),
+                                       nugget + sigma)
+                              )
+  }
+    require(numDeriv)
```

```
+    lags <- as.vector(obj)
+    par.deriv.phi <- jacobian(cov.fun, x = phi, sigma = sigma, nugget = nugget,
+                      lags = lags, cov.model = cov.model, method = deriv.method)
+    jacobian.phi<- matrix(par.deriv.phi, nrow = n.obj, byrow= TRUE)
+    par.deriv.sigma <- jacobian(cov.fun, x = sigma, phi = phi, nugget = nugget,
+                        lags = lags, cov.model = cov.model, method = deriv.method)
+    jacobian.sigma <- matrix(par.deriv.sigma, nrow = n.obj, byrow= TRUE)
+    if(nugget == 0)
+         {
+           derivatives <- list(jacob.sigma = jacobian.sigma, jacob.phi =
+                          jacobian.phi)  }
+    else {
+           par.deriv.tau <- jacobian(cov.fun, x = nugget, phi = phi, sigma = sigma,
+                            lags = lags, cov.model = cov.model, method =
+                            deriv.method)
+           jacobian.tau <- matrix(par.deriv.tau, nrow = n.obj, byrow= TRUE)
+           derivatives <- list(jacob.tau = jacobian.tau, jacob.sigma =
+                          jacobian.sigma, jacob.phi = jacobian.phi)
+      }
+    return(derivatives)
+}
```

A.2.2 Function for computing the Kacker-Harville and Prasad-Rao estimators

```
> krige.approx <- function(geodata, coords, data, locations, cov.pars, cov.model =
+                         "gaussian", obj.model, mean = 0, nugget = 0, type.krige =
+                         "ok", trend.d = "cte", trend.l = "cte", micro.scale = 0,
+                         dist.epsilon = 1e-10, aniso.pars = NULL, output =
+                         output.control(messages = FALSE), lik.method = "ML",
+                         deriv.method = "Richardson", tol = NULL)
+ {
+     require(geoR)
+     if (missing(geodata) & missing(coords))
+         { stop ( "Need to specify data and coordinates") }
+     if (missing(geodata)) {
+         spatial.data <-cbind(coords,data)
+         geodata <- as.geodata(obj= spatial.data, coords.col=1:2, data.col = 3)
+                      n.obs <- nrow(geodata$coords) }
+     else { n.obs <- nrow(geodata$coords)}
+     if (missing(locations))
+            {  stop("Need to specify prediction locations") }
+     else{
+           dimnames(geodata$coords) <- NULL
+           if (is.data.frame(locations) == TRUE)
+                 { locations <- as.matrix(locations) }
+           dimnames(locations) <- NULL
+           locs <- rbind(geodata$coords, locations)
+           dimnames(locs) <- NULL
+           locs <- unique(locs)
+           n.pred <- nrow(locs)
+           locations <- locs[(n.obs + 1) : n.pred,]
```

```
+           locations <- matrix(locations, ncol = 2, dimnames = NULL)
+           n.locs <- nrow(locations)
+       }
+     n.obs <- nrow(geodata$coords)
+     coords.data <- geodata$coords
+     dimnames(coords.data) <- NULL
+     lag.x <- matrix(rep(coords.data[,1], each = n.obs), nrow = n.obs,
+             byrow = TRUE)
+     lag.y <- matrix(rep(coords.data[,2], each = n.obs), nrow = n.obs,
+             byrow = TRUE)
+     diff.x <- (lag.x - t(lag.x))^2  ; diff.y <- (lag.y -t(lag.y))^2
+     lags.data.to.data <- as.matrix(sqrt(diff.x + diff.y))
+     if(missing(obj.model) == FALSE )
+         { cov.model <- obj.model$cov.model
+           cov.pars <- obj.model$cov.pars
+           nugget <- obj.model$tausq
+           mean <- obj.model$beta
+           lik.method <- obj.model$method.lik
+           trend.d <- obj.model$trend
+        }
+     if(missing(obj.model) == FALSE )
+         {
+           krige.settings <- krige.control(type.krige = type.krige, obj.model =
+                             obj.model, dist.epsilon = dist.epsilon, micro.scale
+                             = micro.scale) }
+     else {
+           if( type.krige == "sk" | type.krige == "SK")
+               { beta <- mean }
+           if(missing(obj.model) == TRUE )
+              { beta <- NULL }
+           krige.settings  <- krige.control(type.krige = type.krige, trend.d =
+                              trend.d, trend.l = trend.l, cov.model = cov.model,
+                              cov.pars = c(cov.pars[1],cov.pars[2]), kappa =
+                              kappa, beta = beta, nugget = nugget, micro.scale
+                              = micro.scale, dist.epsilon = dist.epsilon ,
+                              aniso.pars = aniso.pars, lambda = lambda)
+       }
+     krige <- krige.conv(geodata = geodata, locations = locations,
+              krige = krige.settings, output = output)
+     COV <- cov.spatial(lags.data.to.data, cov.model = cov.model, cov.pars =
+            cov.pars, kappa = kappa)
+     if(nugget != 0 )
+          {
+            dim.COV <- nrow(COV)
+            I.COV <- matrix(0, nrow = dim.COV , ncol = dim.COV)
+            diag(I.COV) <- 1
+            COV <- nugget * I.COV + COV }
+     if(missing(tol) == FALSE)
+        { inv.COV <- solve(COV, tol = tol) }
+     else { inv.COV <- solve(COV) }
+     cov.pars <- matrix(cov.pars, nrow = 1)
+     if(ncol(cov.pars) != 2)
+                { stop ( "Arguement cov.pars should be a vector of size 2" )
+              }
```

```
+     derivatives.COV <- partial.deriv(cov.pars = cov.pars, nugget = nugget,
+                         obj = lags.data.to.data, cov.model = cov.model,
+                         deriv.method = deriv.method)
+     if( nugget != 0 )
+          { deriv.COV.tau <- derivatives.COV$jacob.tau }
+     deriv.COV.phi <- derivatives.COV$jacob.phi
+     deriv.COV.sigma <- derivatives.COV$jacob.sigma
+     trend.mat <- trend.spatial(trend = trend.d, geodata = geodata)
+     degree.trend <- ncol(trend.mat)
+     trans.trend <- matrix(t(trend.mat), nrow = degree.trend)
+     if( lik.method == "REML" | lik.method == "reml")
+         {  require(MASS)
+            P <- inv.COV - inv.COV %*% trend.mat %*% ginv(trans.trend %*% inv.COV %*%
+                 trend.mat) %*% trans.trend %*% inv.COV }
+     if(missing(tol) == FALSE)
+        { Bp <- inv.COV %*% trend.mat %*% solve(trans.trend %*% inv.COV %*% trend.mat,
+                tol = tol) }
+     else {Bp <- inv.COV %*% trend.mat %*% solve(trans.trend %*% inv.COV %*% trend.mat)
+         }
+     Ap <- inv.COV - Bp %*% trans.trend %*% inv.COV
+     if(lik.method == "REML" | lik.method == "reml")  { M <- P }
+     else( M <- inv.COV )
+     if(nugget != 0)
+         {
+          a11 <- 0.5 * sum(diag( M %*% deriv.COV.tau %*% M %*%
+                 deriv.COV.tau))
+          a12 = a21 <- 0.5 * sum(diag(M %*% deriv.COV.tau %*% M %*%
+                       deriv.COV.sigma))
+          a13 = a31 <- 0.5 * sum(diag(M %*% deriv.COV.tau %*% M %*%
+                       deriv.COV.phi))
+          a22 <- 0.5 * sum(diag(M %*% deriv.COV.sigma %*% M %*%
+                 deriv.COV.sigma))
+          a23 = a32 <- 0.5 * sum(diag(M %*% deriv.COV.sigma %*% M %*%
+                       deriv.COV.phi))
+          a33 <- 0.5 * sum(diag(M %*% deriv.COV.phi %*% M %*%
+                 deriv.COV.phi))
+          info.mat <- matrix(c(a11, a12, a13, a21, a22, a23, a31,
+                      a32, a33), nrow = 3, byrow = TRUE) }
+     else{
+          a11 <- 0.5 * sum(diag(M %*% deriv.COV.sigma %*% M %*%
+                 deriv.COV.sigma))
+          a12 = a21 <- 0.5 * sum(diag(M %*% deriv.COV.sigma %*% M %*%
+                       deriv.COV.phi))
+          a22 <- 0.5 * sum(diag(M %*% deriv.COV.phi %*%
+                 M %*% deriv.COV.phi))
+          info.mat  <- matrix(c(a11, a12, a21, a22), nrow = 2,
+                       byrow = TRUE)  }
+     if(missing(tol) == FALSE)
+        { inv.info <- solve(info.mat, tol = tol) }
+     else {inv.info <- solve(info.mat) }
+     bias.approx <- matrix(nrow = n.locs)
+     if(nugget == 0 ) { deriv.COV.tau <- NULL }
+     lapply.out <- lapply(1: n.locs, function(i, locations, cov.model, cov.pars,
+                   coords, krige, nugget, obj, deriv.method, Ap, Bp, deriv.COV.tau,
```

```
+                 deriv.COV.phi, deriv.COV.sigma, COV)
+           {
+            pred.loc <- locations[i,]
+            pred.loc <- matrix(pred.loc, ncol = 2)
+            rep.x.pred <- rep(pred.loc[,1], each = n.obs)
+            rep.y.pred <- rep(pred.loc[,2], each = n.obs)
+            rep.pred <- cbind(rep.x.pred, rep.y.pred)
+            dimnames(rep.pred) <- NULL
+            data.to.point <- (coords.data - rep.pred)^2
+            lags.x.pred <- data.to.point[,1] ; lags.y.pred <- data.to.point[,2]
+            lags.data.to.point <- as.matrix(sqrt(lags.x.pred + lags.y.pred))
+            dimnames(lags.data.to.point) <- NULL
+            cov <- cov.spatial(lags.data.to.point, cov.model = cov.model,
+                   cov.pars = cov.pars, kappa = kappa)
+            cov <- nugget + cov
+            lambda <- krweights(coords = coords.data, locations = pred.loc,
+                      krige = krige.settings)
+            f0 <- trans.trend %*% lambda
+            derivatives.cov <- partial.deriv(cov.pars = cov.pars, nugget = nugget,
+                               obj = lags.data.to.point, cov.model = cov.model,
+                               deriv.method = deriv.method)
+            if( nugget != 0)
+                 { deriv.cov.tau <- derivatives.cov$jacob.tau }
+            deriv.cov.phi <- derivatives.cov$jacob.phi
+            deriv.cov.sigma <- derivatives.cov$jacob.sigma
+            if(nugget != 0)
+                 { lambda.tau <-  Ap %*% (deriv.cov.tau - deriv.COV.tau %*%
+                                  (Bp %*% f0 + Ap %*% cov))
+                   t.lam.tau <- t(lambda.tau)
+                }
+            lambda.phi <- Ap %*% (deriv.cov.phi - deriv.COV.phi  %*%(Bp %*%
+                           f0 + Ap %*% cov))
+            t.lam.phi <- t(lambda.phi)
+            lambda.sigma <- Ap %*% (deriv.cov.sigma - deriv.COV.sigma  %*%
+                             (Bp %*% f0 + Ap %*% cov))
+            t.lam.sigma <- t(lambda.sigma)
+            if(nugget != 0)
+              {
+                a11 <- t.lam.tau %*% COV %*% lambda.tau
+                a12 = a21 <-  t.lam.tau %*% COV %*% lambda.sigma
+                a13 = a31 <-  t.lam.tau %*% COV %*% lambda.phi
+                a22 <-  t.lam.sigma %*% COV %*% lambda.sigma
+                a23 = a32 <- t.lam.sigma %*% COV %*% lambda.phi
+                a33 <-  t.lam.phi %*% COV %*% lambda.phi
+                par.Var.Z <- matrix(c(a11, a12, a13, a21, a22, a23, a31, a32,
+                             a33), nrow = 3, byrow = TRUE)
+             }
+             else {
+                a11 <- t.lam.sigma %*% COV %*% lambda.sigma
+                a12 = a21 <- t.lam.sigma %*% COV %*% lambda.phi
+                a22 <- t.lam.phi %*% COV %*% lambda.phi
+                par.Var.Z  <- matrix(c(a11, a12, a21, a22), nrow = 2, byrow =
+                              TRUE)
+               }
```

```
+            bias.approx <- sum(diag(par.Var.Z %*% inv.info))
+         },
+           locations = locations, cov.model = cov.model, cov.pars = cov.pars,
+           coords = coords.data, krige = krige.settings, nugget = nugget, obj
+           = lags.data.to.point, deriv.method = deriv.method, Ap = Ap, Bp = Bp,
+           deriv.COV.tau = deriv.COV.tau, deriv.COV.phi = deriv.COV.phi,
+           deriv.COV.sigma = deriv.COV.sigma, COV = COV)
+   bias.approx <- sapply(lapply.out, function(x)x)
+   predictions <- matrix(krige$predict, nrow = n.locs)
+   plug.in <- matrix(krige$krige.var, nrow = n.locs)
+   Kacker.Harville <- plug.in + bias.approx
+   Prasad.Rao <- plug.in + (2 * bias.approx)
+   variance <- cbind(plug.in, Kacker.Harville, Prasad.Rao)
+   colnames(locations) <- c("X","Y")
+   colnames(variance) <- c("plug.in", "Kacker.Harville", "Prasad.Rao")
+   output <- list(coords = locations, predictions = predictions, variance = variance)
+   return(output)
+}
```

A.3 Functions used in simulation experiment

This function was used in the simulation experiment to approximate the empirical mean square prediction error (EMSPE) of the empirical best linear unbiased predictor (BLUP). As mentioned earlier is has no practical use and is only documented here for completeness.

```
> num.compare <- function(geodata, coords, data, locations, cov.pars, cov.model =
+                         "gaussian", mean = 0, nugget = 0, fix.nugget = FALSE,
+                         kappa = .5, fix.kappa = TRUE, lambda = 1, fix.lambda =
+                         TRUE, psiA = 0, fix.psiA = TRUE, psiR = 1, fix.psiR =
+                         TRUE, aniso.pars = NULL, lik.method = "ML", limits =
+                         pars.limits(), type.krige = "ok", trend.d = "cte",
+                         trend.l = "cte", micro.scale = 0, dist.epsilon = 1e-10,
+                         output = output.control(messages = FALSE), nsim = 1000,
+                         sim.method = "cholesky", RF = TRUE, messages = FALSE,
+                         deriv.method = "Richardson", random.seed = NULL,
+                         n.boot = 1000, tol = NULL)
+ {
+    require(geoR)
+    if (missing(geodata) & missing(coords))
+        { stop ( "Need to specify data and coordinates") }
+    if (missing(geodata)) {
+        geodata <- list(coords = coords, data = data)
+                    n.obs <- ncol(geodata$coords) }
+    else { n.obs <- nrow(geodata$coords)}
+    if (missing(locations))
+           {  stop("Need to specify prediction locations") }
+    else{
+          dimnames(geodata$coords) <- NULL
```

```
+          if (is.data.frame(locations) == TRUE)
+                { locations <- as.matrix(locations) }
+          dimnames(locations) <- NULL
+          locs <- rbind(geodata$coords, locations)
+          dimnames(locs) <- NULL
+          locs <- unique(locs)
+          n.coords <- nrow(locs)
+          locations <- locs[(n.obs + 1) : n.coords,]
+          locations <- matrix(locations, ncol = 2, dimnames = NULL)
+          n.locs <- nrow(locations)
+      }
+    n.obs <- nrow(geodata$coords)
+    coords.data <- geodata$coords
+    dimnames(coords.data) <- NULL
+    if( type.krige == "sk" | type.krige == "SK")
+          { beta <- mean }
+    else{beta <- NULL }
+    krige.settings  <- krige.control(type.krige = type.krige, trend.d =
+                       trend.d, trend.l = trend.l, cov.model = cov.model,
+                       cov.pars = c(cov.pars[1],cov.pars[2]), kappa =
+                       kappa, beta = beta, nugget = nugget, micro.scale
+                       = micro.scale, dist.epsilon = dist.epsilon ,
+                       aniso.pars = aniso.pars, lambda = lambda)
+    output <- output.control(n.predictive = n.boot, simulations.predictive = TRUE,
+              mean.var = TRUE, sim.means = TRUE, sim.vars = TRUE, messages = messages)
+    krige <- krige.conv(geodata = geodata, locations = locations,
+             krige = krige.settings, output = output)
+    original.predict <- krige$predict
+    lik.est.data <- likfit(geodata = geodata, trend = trend.d, ini.cov.pars
+                = cov.pars, nugget = nugget, fix.nugget = fix.nugget,
+                kappa = kappa, fix.kappa = fix.kappa, lambda =lambda,
+                fix.lambda = fix.lambda, psiA = psiA, fix.psiA = fix.psiA,
+                psiR = psiR, fix.psiR = fix.psiR, cov.model = cov.model,
+                limits = limits, lik.method = lik.method, messages =
+                messages)
+    cat("\n !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! \n")
+    cat("\n ESTIMATED PARAMETERS OF DATA \n")
+    flush.console()
+    print(lik.est.data)
+    flush.console()
+    cat("\n !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! \n")
+    plug.in <- matrix(nrow = n.locs, ncol = nsim)
+    Kacker.Harville <- matrix(nrow = n.locs, ncol = nsim)
+    Prasad.Rao <- matrix(nrow = n.locs, ncol = nsim)
+    boot.uncond <- matrix(nrow = n.locs, ncol = nsim) #
+    boot.cond <- matrix(nrow = n.locs, ncol = nsim)
+    SPE <- matrix(nrow = n.locs, ncol = nsim)
+    cat("\n ********************************************************** \n")
+    cat("\n CALCULATING ESTIMATED PARAMETERS OF SIMULATIONS \n")
+    cat("\n ********************************************************** \n")
+    cat("\n                                                  \n")
+    flush.console()
+    if(missing(random.seed) == FALSE ){ set.seed(random.seed)}
+    for(i in 1: nsim)
```

```
+           {
+            sim.grf <- grf(n = n.coords, grid = locs, nsim = 1, cov.model =
+                       cov.model, cov.pars = cov.pars, aniso.pars = aniso.pars,
+                       mean = mean, nugget = nugget, kappa = kappa, lambda =
+                       lambda, method = sim.method, RF = RF, messages = messages)
+            sim.samples <- as.matrix(sim.grf$data)
+            sim.obs <- drop(as.matrix(sim.samples[ 1 : n.obs, ]))
+            temp.data <-as.geodata(cbind(geodata$coords, sim.obs))
+            sim.preds <- as.matrix(sim.samples[ (n.obs + 1) : n.coords, ])
+            lik.est <- likfit(geodata = temp.data, trend = trend.d, ini.cov.pars
+                       = cov.pars, nugget = nugget, fix.nugget = fix.nugget,
+                       kappa = kappa, fix.kappa = fix.kappa, lambda =lambda,
+                       fix.lambda = fix.lambda, psiA = psiA, fix.psiA = fix.psiA,
+                       psiR = psiR, fix.psiR = fix.psiR, cov.model = cov.model,
+                       limits = limits, lik.method = lik.method, messages =
+                       messages)
+            print(lik.est)
+            var.estimates <- krige.approx(geodata = temp.data, locations =
+                             locations, obj.model = lik.est, dist.epsilon
+                             = dist.epsilon, micro.scale = micro.scale, tol = tol)
+            cond.boot <- krige.cond(geodata = temp.data , locations = locations,
+                         obj.model = lik.est, type.krige= "ok", nsim = n.boot,
+                         output = output)
+            uncond.boot <- krige.uncond(geodata = temp.data, obj.model = lik.est,
+                           locations = locations, type.krige = type.krige, nsim =
+                           n.boot, fix.nugget = fix.nugget, limits = limits,
+                           lik.method = lik.method, print.lik = FALSE)
+            plug.in[,i] <- var.estimates$variance[,1]
+            Kacker.Harville[,i] <-  var.estimates$variance[,2]
+            Prasad.Rao[,i] <-  var.estimates$variance[,3]
+            boot.cond[,i] <- cond.boot$pred.errors[,2]
+            boot.uncond[,i] <- uncond.boot$pred.errors[,2]
+            SPE[,i] <- matrix((sim.preds - var.estimates$predictions)^2, ncol = 1)
+            print(i)
+            flush.console()
+          }
+   sigma.p <- as.vector(apply(plug.in, 1, sum)/ nsim)
+   sq.std.p <- as.matrix(apply((plug.in - sigma.p)^2, 1, sum)/(nsim - 1))
+   std.p <- sqrt(sq.std.p)/sqrt(nsim)
+   sigma.kh <- as.vector(apply(Kacker.Harville, 1, sum)/ nsim)
+   sq.std.kh <- as.matrix(apply((Kacker.Harville - sigma.kh)^2, 1, sum)/(nsim - 1))
+   std.kh <- sqrt(sq.std.kh)/sqrt(nsim)
+   sigma.pr<- as.vector(apply(Prasad.Rao, 1, sum)/ nsim)
+   sq.std.pr <- as.matrix(apply((Prasad.Rao - sigma.pr)^2, 1, sum)/(nsim - 1))
+   std.pr <- sqrt(sq.std.pr)/sqrt(nsim)
+   EMSPE<- as.vector(apply(SPE, 1, sum)/ nsim)
+   sq.std.EMSPE <- as.matrix(apply((SPE - EMSPE)^2, 1, sum)/ (nsim - 1))
+   std.EMSPE <- sqrt(sq.std.EMSPE)/sqrt(nsim)
+
+   cond.MSPE<- as.vector(apply(boot.cond, 1, sum)/ nsim)
+   cond.sq.std.MSPE <- as.matrix(apply((boot.cond - cond.MSPE)^2, 1,
+                       sum)/ (nsim - 1))
+   cond.std.MSPE <- sqrt(cond.sq.std.MSPE)/sqrt(nsim)
+
```

```
+   uncond.MSPE<- as.vector(apply(boot.uncond, 1, sum)/ nsim)
+   uncond.sq.std.MSPE <- as.matrix(apply((boot.uncond - uncond.MSPE)^2, 1,
+                           sum)/ (nsim - 1))
+   uncond.std.MSPE <- sqrt(uncond.sq.std.MSPE)/sqrt(nsim)
+
+   colnames(locations) <- c("X","Y")
+   mean.vars <- cbind(EMSPE, sigma.p, sigma.kh, sigma.pr, cond.MSPE, uncond.MSPE)
+   colnames(mean.vars) <- c("EMSPE", "plug.in", "Kacker.Harville", "Prasad.Rao",
+                            "MSPE.cond", "MSPE.uncond")
+   std.errors <- cbind(std.EMSPE, std.p, std.kh, std.pr, cond.std.MSPE,
+                  uncond.std.MSPE)
+   colnames(std.errors) <-c("EMSPE", "plug.in", "Kacker.Harville", "Prasad.Rao",
+                           "MSPE.cond", "MSPE.uncond")
+   output <- list(coords = locations, predictions = original.predict,
+              theoratical.var = krige$krige.var, mean.vars = mean.vars, std.errors
+              = std.errors)
+   return(output)
+ }


> num.solve <- function(geodata, coords, data, locations, cov.pars, cov.model =
+                           "matern", obj.model, mean = 0, nugget = 0, fix.nugget
+                           = FALSE, kappa = .5, fix.kappa = TRUE, lambda = 1,
+                           fix.lambda = TRUE, psiA = 0, fix.psiA = TRUE, psiR = 1,
+                           fix.psiR = TRUE, aniso.pars = NULL, lik.method = "ML",
+                           limits = pars.limits(), type.krige = "ok", trend.d =
+                           "cte", trend.l = "cte", micro.scale = 0, dist.epsilon =
+                           1e-10, output = output.control(messages = FALSE), nsim =
+                           1000, sim.method = "cholesky", con.level= 0.05, RF = TRUE,
+                           messages = FALSE, print.lik = TRUE)
+{
+     require(geoR)
+     if (missing(geodata) & missing(coords))
+         { stop ( "Need to specify data coordinates") }
+     if (missing(geodata)) {
+            n.obs <- nrow(coords) }
+     else { n.obs <- nrow(geodata$coords)
+            coords <- geodata$coords }
+            coords.data <- as.matrix(coords)
+            dimnames(coords.data) <-NULL
+     if (missing(locations))
+            {  stop("need to specify prediction locations") }
+     else{
+           if (missing(geodata) == FALSE )
+                { dimnames(geodata$coords) <- NULL }
+           if (is.data.frame(locations) == TRUE)
+                 { locations <- as.matrix(locations) }
+           dimnames(locations) <- NULL
+           locs <- rbind(coords.data, locations)
+           dimnames(locs) <- NULL
+           locs <- unique(locs)
+           n.coords <- nrow(locs)
+           locations <- locs[(n.obs + 1) : n.coords,]
+           locations <- matrix(locations, ncol = 2, dimnames = NULL)
+           n.locs <- nrow(locations)
```

```
+       }
+     if(missing(obj.model) == FALSE )
+         { cov.model <- obj.model$cov.model
+           cov.pars <- obj.model$cov.pars
+           nugget <- obj.model$tausq
+           mean <- obj.model$beta
+           lik.method <- obj.model$method.lik
+           kappa <- obj.model$kappa
+           lambda <- obj.model$lambda
+           aniso.pars <- obj.model$trend
+           trend.d <- obj.model$trend
+           lik.method <- obj.model$method.lik
+           check.mean <- as.matrix(mean, nrow = 1)
+           if(dim(check.mean)[[1]]>1) { mu <- sum(mean) }
+           else(mu <- mean)
+        }
+     if(missing(obj.model) == FALSE )
+         {
+           krige.settings <- krige.control(type.krige = type.krige, obj.model =
+                             obj.model, dist.epsilon = dist.epsilon, micro.scale
+                             = micro.scale) }
+     else {
+           if( type.krige == "sk" | type.krige == "SK")
+               { beta <- mean }
+           if(missing(obj.model) == TRUE )
+               { beta <- mean }
+           krige.settings  <- krige.control(type.krige = type.krige, trend.d =
+                              trend.d, trend.l = trend.l, cov.model = cov.model,
+                              cov.pars = c(cov.pars[1],cov.pars[2]), kappa =
+                              kappa, beta = beta, nugget = nugget, micro.scale
+                             = micro.scale, dist.epsilon = dist.epsilon ,
+                              aniso.pars = aniso.pars, lambda = lambda)
+           check.mean <- as.matrix(mean, nrow = 1)
+           if(dim(check.mean)[[1]]>1) { mu <- sum(mean) }
+           else(mu <- beta)
+       }
+     lapply.out <- lapply(1:nsim, function(i, n, grid, nsim, cov.model, cov.pars,
+                   aniso.pars, mean, nugget, kappa, lambda, method, RF, messages,
+                   geodata, trend, ini.cov.pars, fix.nugget, fix.kappa, fix.lambda,
+                   psiR, fix.psiR, limits, lik.method, type.krige, obj.model,
+                   dist.epsilon, micro.scale, locations, krige, output)
+        {
+           sim.grf <- grf(n = n.coords, grid = locs, nsim = nsim, cov.model =
+                      cov.model, cov.pars = cov.pars, aniso.pars = aniso.pars,
+                      mean = mu, nugget = nugget, kappa = kappa, lambda =
+                      lambda, method = sim.method, RF = RF, messages = messages)
+           sim.samples <- as.matrix(sim.grf$data)
+           sim.obs <- drop(as.matrix(sim.samples[ 1 : n.obs, ]))
+           temp.data <-as.geodata(cbind(coords.data, sim.obs))
+           sim.preds <- as.matrix(sim.samples[ (n.obs + 1) : n.coords, ])
+           lik.est <- likfit(geodata = temp.data, trend = trend.d, ini.cov.pars
+                      = cov.pars, nugget = nugget, fix.nugget = fix.nugget,
+                      kappa = kappa, fix.kappa = fix.kappa, lambda =lambda,
+                      fix.lambda = fix.lambda, psiA = psiA, fix.psiA = fix.psiA,
```

```
+                     psiR = psiR, fix.psiR = fix.psiR, cov.model = cov.model,
+                     limits = limits, lik.method = lik.method, messages =
+                     messages)
+          if(print.lik == TRUE )
+              {print(lik.est) ; print(i); flush.console () }
+          krige.set.sim <- krige.control(type.krige = type.krige, obj.model =
+                          lik.est, dist.epsilon = dist.epsilon, micro.scale
+                           = micro.scale)
+          est.cov.pars <- matrix(lik.est$cov.pars,nrow=1)
+          if(fix.nugget == FALSE)
+                 { est.cov.pars <- cbind(lik.est$tausq, est.cov.pars) }
+          krige.sim <- krige.conv(geodata = temp.data, locations = locations, krige
+                       = krige.set.sim, output = output)
+          predictions <- as.matrix(krige.sim$predict)
+          COVS <- est.cov.pars
+          SPE <- (predictions - sim.preds)^2
+          out.boot <- list(SPE=SPE, COVS=COVS)
+     },
+        n = n.coords, grid = locs, nsim = 1, cov.model = cov.model, cov.pars =
+        cov.pars, aniso.pars = aniso.pars, mean = mean, nugget = nugget, kappa =
+        kappa, lambda = lambda, method = sim.method, RF = RF, messages = messages,
+        geodata = temp.data, trend = trend.d, ini.cov.pars= cov.pars, fix.nugget =
+        fix.nugget, fix.kappa = fix.kappa, fix.lambda = fix.lambda, psiR = psiR,
+        fix.psiR = fix.psiR, limits = limits, lik.method = lik.method, type.krige =
+        type.krige, obj.model = lik.est, dist.epsilon = dist.epsilon, micro.scale =
+        micro.scale, locations = locations, krige= krige.set.sim, output = output)
+
+   COVS <- sapply(lapply.out, function(x)x$COVS)
+   SPE <- sapply(lapply.out, function(x)x$SPE)
+   sd <- apply(SPE,1,sd)
+   MSPE<- as.vector(apply(SPE, 1, sum)/ nsim)
+   sq.std.MSPE <- as.matrix(apply((SPE - MSPE)^2, 1, sum)/(nsim - 1))
+   std.MSPE <- sqrt(sq.std.MSPE)/sqrt(nsim)
+   if( con.level==0.05 | missing(con.level)==TRUE) { z = 1.96 }
+   else if(con.level==0.10) { z = 1.64 }
+   lb <- MSPE - z * std.MSPE
+   ub <- MSPE + z * std.MSPE
+   bounds <- cbind(lb,ub)
+   colnames(bounds) <- c("lower","upper")
+   output <- list(coords = locations, MSPE = MSPE, std.errors = std.MSPE, sd = sd,
+             bounds = bounds, est.pars = COVS)
+   return(output)
+}
```

Appendix B

R documentation

`krige.uncond` Unconditional bootstrap estimation of the kriging variance

---

Description

This function performs parametric bootstrap estimation of the kriging variance. The bootstrap algorithm employed involves unconditional simulation.

Usage

```
krige.uncond(geodata, coords, data, locations, cov.pars, cov.model = ‘‘matern’’,
             obj.model, mean = 0, nugget = 0, fix.nugget = FALSE, kappa = .5,
             fix.kappa = TRUE, lambda = 1, fix.lambda = TRUE, psiA = 0, fix.psiA
             = TRUE, psiR = 1, fix.psiR = TRUE, aniso.pars = NULL, lik.method =
             ‘‘ML’’, trend.l = ‘‘cte’’, trend.l = ‘‘cte’’, micro.scale = 0,
             dist.epsilon = 1e-10, output = output.control(messages = FALSE),
             output = output.control(messages = FALSE), nsim = 1000, sim.method
              = ‘‘cholesky’’, RF = TRUE, messages = FALSE)
```

Arguments

`geodata`	a list containing elements `coords` and `data` as described next. Typically an object of class `geodata` a **geoR** data set. If not provided the arguments `coords` and `data` must be provided.
`coords`	an $n \times 2$ matrix or data-frame with the 2-D coordinates of the n data locations. By default it takes the component `coords` of the argument, `geodata` if provided.
`data`	a vector with the n data values. By default it takes the component `data` of the argument `geodata`, if provided.
`locations`	a $m \times 2$ matrix or data-frame with the 2-D coordinates of the n prediction locations, or a list for which the first two components are used. Input is internally checked by the function `check.locations`.

`cov.model` a string specifying the model for the correlation function. For further details see documentation for `cov.spatial` in the **geoR** manual. Defaults to the Matern model.

`cov.pars` a 2 elements vector with the values of the covariance parameters σ_0^2 (partial sill) and ϕ (range parameter).

`type.krige` type of kriging to be performed. Options are `''SK''` and `''OK''`corresponding to simple and ordinary kriging. Kriging with external trend and universal kriging can be defined by setting `type.krige=''OK''` and specifying the trend model using the arguments `trend.d` and `trend.l`.

`mean` numerical value of the mean (vector) parameter.

`trend.d` specifies the trend (covariate) values at the data locations. See documentation of `trend.spatial` in the **geoR** manual for further details. Defaults to `''cte''`.

`trend.l` specifies the trend (covariate) values at prediction locations. It must be the same as for `trend.d`.

`obj.model` a list with the model parameters. Typically an output of `likfit` or `variofit`.

`micro.scale` micro-scale variance. If different from zero, the nugget variance is divided into 2 terms: micro-scale variance and measurement error. This affects the precision of the predictions. Often in practice, these two variance components are indistinguishable. See the section DETAILS in the documentation of `output.control` in the **geoR** manual.

`output` a list specifying output options. It can take a call to`output.control` or a list with elements as for the arguments in `output.control`.

`fix.nugget` logical indicating whether the parameter τ^2 (nugget variance) should be regarded as fixed (`fix.nugget = TRUE`) or should be estimated (`fix.nugget = FALSE`). Defaults to `FALSE`.

`nugget` value of the nugget parameter. Regarded as a fixed value if `fix.nugget = TRUE` otherwise as the initial value for the minimization algorithm. Defaults to zero.

`fix.kappa` logical, indicating whether the extra parameter `kappa` should be regarded as fixed (`fix.kappa = TRUE`) or should be estimated (`fix.kappa = FALSE`). Defaults to `TRUE`.

`kappa` value of extra parameter κ. Regarded as a fixed value if `fix.kappa = TRUE` otherwise as the initial value for the minimization algorithm in covariance parameter estimation. Defaults to 0.5. This parameter is valid only if the covariance function is one of either: `''matern''`, `''powered.exponential''`, `''cauchy''` or `gneiting.matern`. For more details on covariance functions see documentation for `cov.spatial` in the **geoR** manual.

`fix.lambda` logical, indicating whether the Box-Cox transformation parameter λ should be regarded as fixed (`fix.lambda = TRUE`) or should be estimated (`fix.lambda = "FALSE"`). Defaults to `TRUE`.

`lambda` value of the Box-Cox transformation parameter λ. Regarded as fixed if `fix.lambda = TRUE` otherwise as the initial value for the minimization algorithm in covariance parameter estimation. Defaults to 1. Two particular cases are $\lambda = 1$ indicating no transformation and $\lambda = 0$ indicating log transformation.

`fix.psiR` logical, indicating whether the anisotropy ratio parameter ψ_R should be regarded as fixed (`fix.psiR = TRUE`) or should be estimated (`fix.psiR = FALSE`). Defaults to `TRUE`.

`psiR` value, always greater than 1, for the anisotropy ratio parameter ψ_R. Regarded as a fixed value if `fix.psiR=TRUE` otherwise as the initial value in the minimization algorithm for covariance parameter estimation. Defaults to 1. See documentation on `coords.aniso` in the **geoR** manual for further details on anisotropy correction.

`lik.method` options are `ML` for maximum likelihood and `REML` for restricted maximum likelihood. Defaults to `ML`.

`nsim` number of iterations. Defaults to 1000.

`limits` values defining lower and upper limits for the model parameters used in the numerical minimization. The auxiliary function `pars.limits` is called to set the limits. See also **Limits** in the documentation for the function `likfit` in the **geoR** manual.

`dist.epsilon` a numeric value. Locations which are separated by a distance less than this value are considered co-located.

`sim.method` simulation method with options for `cholesky`, `svd`, `eigen`, `RF`. Defaults to the Cholesky decomposition. See DETAILS section in the documentation for `grf` in the **geoR** manual.

`fix.psiA` logical, indicating whether the anisotropy angle parameter ψ_R should be regarded as fixed (`fix.psiA = TRUE`) or should be estimated (`fix.psiA = FALSE`). Defaults to `TRUE`.

`aniso.pars` two elements vector with the parameters for the geometric anisotropy correction. If `aniso.pars = NULL` no correction is made. Anisotropy correction consists of a transformation of the data and prediction coordinates performed by the **geoR** function `coords.aniso`.

`RF` logical, with defaults to TRUE, indicating whether the algorithm should try to use the function `GaussRF` from the package **RandomFields** in case `method` is missing and the number of points is greater than 500.

`messages` logical, indicating whether or not status messages are printed on the screen (or output device) while the function is running.

Details

This functions uses a combination of **geoR** functions to carry out parametric bootstrap estimation of the kriging variance. It uses the functions `krige.conv()` for kriging, `likfit()` for covariance parameter estimation and `grf()` for simulating random fields. For further information on these functions see their documentation in the **geoR** manual.

Value

`coords`	an $m \times 2$ matrix or data-frame with the 2-D coordinates of the m prediction locations. By default it takes the component `locations` of the argument.
`predictions`	a $n \times 1$ vector with predicted values.
`pred.errors`	a $m \times 2$ matrix containing the plug-in and bootstrapped kriging variance estimates.
`std.errors`	a $m \times 1$ vector containing the bootstrapped MSPE standard errors.
`bounds`	a $m \times 2$ matrix containing the lower bound and upperbound of the 95% confidence interval of the MPSE.
`est.pars`	a $p \times m$ matrix containing the estimates of the variance parameters $\boldsymbol{\theta}$ of the p parameters and m simulations.

Author(s)

Mzabalazo Z. Ngwenya `<NgwenyaM@arc.agric.za>`
Christien Thiart `<Christien.Thiart@uct.ac.za>`
Linda M. Haines `<Linda.Haines@uct.ac.za>`

References

The **geoR manual** (Riberio and Diggle, 2006). Further information can be found at: `http://www.est.ufpr.br/geoR.`

Diggle, P.J. and Riberio, P.J. (2007). *Model-based Geostatistics*. Springer. New York.

See also

`grf`, `likfit`, `krige.control`, `output.control` in the **geoR** manual and `krige.uncond`.

Examples

```
library(geoR)
grid <- expand.grid(1:5,1:5)  # Forming a grid
set.seed (1403)  # Random seed for simulating Gaussian random field
sim.grid <- grf(grid = grid, cov.model = ‘‘exponential’’, cov.pars = c(1,5))
fit <- likfit(sim.grid, cov.model = ‘‘exponential’’, ini.cov.pars = c(1,5), nugget,
        = 0, fix.nugget = FALSE)  # MLE of simulated field parameter’s
pred.loc <-expand.grid(1.5:4.5,1.5:4.5)
set.seed(13)
krige.uncond(geodata= sim.grid, obj.model = fit, locations = pred.loc, type.krige
             = ‘ok’’, nsim = 1000)
```

`krige.cond` Conditional bootstrap estimation of the kriging variance

Description

This function performs parametric bootstrap estimation of the kriging variance. The bootstrap algorithm employed involves conditional simulation.

Usage

```
krige.cond(geodata, coords, data, locations, cov.pars, cov.model = ‘‘matern’’,
           obj.model, mean, nugget = 0, kappa = .5, lambda = 1, type.krige =
           ‘‘ok’’, trend.d = ‘‘cte’’, trend.l = ‘‘cte’’, aniso.pars = NULL,
            micro.scale = 0, dist.epsilon = 1e-10, output = output.control
           (messages = FALSE), nsim = 1000, sim.method = ‘‘cholesky’’, RF =
           TRUE, messages = FALSE)
```

Arguments

`geodata` a list containing elements `coords` and `data` as described next. Typically an object of class `geodata` a **geoR** data set. If not provided the arguments `coords` and `data` must be provided.

`coords` an $n \times 2$ matrix or data-frame with the 2-D coordinates of the n data locations. By default it takes the component `coords` of the argument, `geodata` if provided.

`data` a vector with the n data values. By default it takes the component `data` of the argument `geodata`, if provided.

`locations` a $m \times 2$ matrix or data-frame with the 2-D coordinates of the m prediction locations, or a list for which the first two components are used. Input is internally checked by the function `check.locations`.

`cov.model` a string specifying the model for the correlation function. For further details see documentation for `cov.spatial` in the **geoR** manual. Defaults to the Matérn model.

`cov.pars` a 2 elements vector with the values of the covariance parameters σ_0^2 (partial sill) and ϕ (range parameter).

`type.krige` type of kriging to be performed. Options are ‘‘`SK`’’ and ‘‘`OK`’’corresponding to simple and ordinary kriging. Kriging with external trend and universal kriging can be defined by setting `type.krige=‘‘OK’’` and specifying the trend model using the arguments `trend.d` and `trend.l`.

`mean` numerical value of the mean (vector) parameter.

`trend.d` specifies the trend (covariate) values at the data locations. See documentation of `trend.spatial` in the **geoR** manual for further details. Defaults to ‘‘`cte`’’.

`trend.l` specifies the trend (covariate) values at prediction locations. It must be the same as for `trend.d`.

`obj.model` a list with the model parameters. Typically an output of `likfit` or `variofit`.

`micro.scale` micro-scale variance. If different from zero, the nugget variance is divided into 2 terms: micro-scale variance and measurement error. This affects the precision of the predictions. Often in practice, these two variance components are indistinguishable.

`output` a list specifying output options. It can take a call to`output.control` or a list with elements as for the arguments in `output.control`.

`nugget` value of the nugget parameter. Regarded as a fixed value if `fix.nugget = TRUE` otherwise as the initial value for the minimization algorithm. Defaults to zero.

`kappa` value of extra parameter κ. Regarded as a fixed value if `fix.kappa = TRUE` otherwise as the initial value for the minimization algorithm in covariance parameter estimation. Defaults to 0.5. This parameter is valid only if the covariance function is one of either: ``matern'', ``powered.exponential'', ``cauchy'' or `gneiting.matern`.

`lambda` value of the Box-Cox transformation parameter λ. Regarded as fixed if `fix.lambda = TRUE` otherwise as the initial value for the minimization algorithm in covariance parameter estimation. Defaults to 1. Two particular cases are $\lambda = 1$ indicating no transformation and $\lambda = 0$ indicating log transformation.

`aniso.pars` two elements vector with the parameters for the geometric anisotropy correction. If `aniso.pars = NULL` no correction is made. Anisotropy correction consists of a transformation of the data and prediction coordinates performed by the **geoR** function `coords.aniso`.

`nsim` number of iterations. Defaults to 1000.

`dist.epsilon` a numeric value. Locations which are separated by a distance less than this value are considered co-located.

`psiA` value (in radians) for the anisotropy angle parameter ψ_A. Regarded as a fixed value if `fix.psiA=TRUE` otherwise as the initial value for the minimization algorithm in covariance parameter estimation. Defaults to 0. See documentation on `coords.aniso` in the **geoR** manual for further details.

`signal` logical indicating whether the signal or variable is to be predicted. Defaults to `NULL`. See `output.control()` help file for details.

`messages` logical, indicating whether or not status messages are printed on the screen (or output device) while the function is running.

Details

This functions uses a combination of **geoR** functions to carry out parametric bootstrap estimation of the kriging variance. It uses the functions `krige.conv()` and the auxiliary function `output.control()` to generate conditional bootstrap samples. Because this function does not estimate parameters for each of the simulations as the data at the observation locations always remain the same, one can perform bootstrap estimation with a very large number of iteration very quickly unlike in the `krige.uncond` function.

Value

`coords`	an $n \times 2$ matrix or data-frame with the 2-D coordinates of the n data locations. By default it takes the component `coords` of the argument.
`predictions`	a $m \times 1$ vector with predicted values.
`pred.errors`	a $m \times 2$ matrix containing the plug-in and bootstrapped kriging variance estimates.
`std.errors`	a $n \times 1$ vector containing the bootstrapped MSPE standard errors.
`bounds`	a $n \times 2$ matrix containing the lower bound and upperbound of the 95 % confidence interval of the MPSE.
`sim.mean`	a $nsim \times 1$ vector of the means of the $nsim$ simulations.
`sim.vars`	a $nsim \times 1$ vector of the variances of the $nsim$ simulations.

Author(s)

Mzabalazo Z. Ngwenya `<NgwenyaM@arc.agric.za>`
Christien Thiart `<Christien.Thiart@uct.ac.za>`
Linda M. Haines `<Linda.Haines@uct.ac.za>`

References

The **geoR manual** (Riberio and Diggle, 2006). Further information can be found at: `http://www.est.ufpr.br/geoR`.

Diggle, P.J. and Riberio, P.J. (2007). *Model-based Geostatistics*. Springer. New York.

See also

`likfit`, `krige.control`, `output.control` in the **geoR** manual and `krige.uncond`.

Examples

```
library(geoR)
grid <- expand.grid(1:5,1:5)  # Forming a grid
set.seed (1403)  # Random seed for simulating Gaussian random field
sim.grid <- grf(grid = grid, cov.model = ‘‘exponential’’, cov.pars = c(1,5))
fit <- likfit(sim.grid, cov.model = ‘‘exponential’’, ini.cov.pars = c(1,5), nugget,
        = 0, fix.nugget = FALSE)  # MLE of simulated field parameter’s
pred.loc <-expand.grid(1.5:4.5,1.5:4.5)
set.seed(13)
krige.cond(geodata = sim.grid, obj.model = fit, locations = pred.loc, type.krige
            = ‘‘ok’’, nsim = 1000)
```

Appendix C

R syntax and output

The first section of this appendix shows the R syntax used in carrying carrying out the analysis and kriging variance estimation of the Meuse and South African data sets of Chapter 4.

C.1 Meuse Data

```
            ############################################################
            #                                                          #
            #                  INITIALIZING ANALYSIS                   #
            #                                                          #
            ############################################################

 # Loading packages
> library(gstat)
> library(geoR)

  # Loading data
> data(meuse)
> attach(meuse)
  # data frame for prediction locations
> data(meuse.grid)

  # Creating geodata object
> Zn <-as.geodata(obj = meuse, data.col = 6, coords.col = 1:2, covar.col = 8)


            ############################################################
            #                                                          #
            #                EXPLORATORY DATA ANALYSIS                 #
            #                                                          #
            ############################################################

  # Bubble plot of Zinc concentrations
> coordinates(meuse) = ~x+y
```

```
> bubble(meuse,"zinc", maxsize = 2, main="",  scales=list(draw = TRUE), col = "red",
+ xlab = "Eastings(m)", ylab = "Northings(m)")

 # EDA of Zn data
> plot(Zn, lowess = TRUE, density = FALSE))


 # Log transform
> log.Zn <-as.geodata(obj = meuse, data.col = 6, coords.col = 1:2, covar.col = 8)
> log.Zn$data <-log(log.Zn$data)

-----------

 # Q-Q plot of Zinc data
> parmfrow = c(1,2)
> qqnorm(Zn$data)
> qqline(Zn$data)

 #Q-Q plot of log-transformed data
> qqnorm(log(Zn$data))
> qqline(log(Zn$data))

---------

  # EDA of log transformed data with distance to the river as a covariate
> plot(log.Zn, trend = ~dist, lowess = TRUE, density = FALSE)

              ############################################################
              #                                                          #
              #                    PARAMETER ESTIMATION                  #
              #                                                          #
              ############################################################

 # matrix of initial values for the variance and range
> ini.cov <- expand.grid(seq(0,40, by = 1), seq(0,2500, by = 100))

-----------------------------------------------

 # MODELLING FITTING UNDER CONSTANT MEAN ASSUMPTION

> meuse.con.exp <- likfit(geodata = log.Zn, cov.model = "exponential", ini.cov.pars =
+                  ini.cov, nugget = seq(0,5,by=0.25), fix.nugget = FALSE)
> meuse.con.exp
likfit: estimated model parameters:
       beta       tausq     sigmasq         phi
"   6.5577" "   0.0335" "   1.4987" "1700.0000"
Practical Range with cor=0.05 for asymptotic range: 5092.745

likfit: maximised log-likelihood = -99.16


> meuse.con.mat1 <- likfit(geodata = log.Zn, cov.model = "matern", ini.cov.pars =
+                   ini.cov, nugget = seq(0,5,by=0.25), fix.nugget = FALSE,
+                   kappa = 1.5, fix.kappa = TRUE)
```

```
> meuse.con.mat1
likfit: estimated model parameters:
     beta     tausq   sigmasq       phi
"  6.491" "  0.095" "  1.414" "440.127"
Practical Range with cor=0.05 for asymptotic range: 2087.901

likfit: maximised log-likelihood = -97.38


> meuse.con.mat2 <- likfit(geodata = log.Zn, cov.model = "matern", ini.cov.pars =
+                   ini.cov, nugget = seq(0,5,by=0.25), fix.nugget = FALSE,
+                   kappa = 2.5, fix.kappa = TRUE)
> meuse.con.mat2
likfit: estimated model parameters:
     beta     tausq   sigmasq       phi
"  6.378" "  0.104" "  1.147" "259.544"
Practical Range with cor=0.05 for asymptotic range: 1536.148

likfit: maximised log-likelihood = -97.82


> meuse.con.gauss <- likfit(geodata = log.Zn, cov.model = "gaussian", ini.cov.pars
+                    = c(1.147,259.544), nugget = seq(0,5,by=0.25), fix.nugget =
+                    FALSE)
> meuse.con.gauss
likfit: estimated model parameters:
      beta      tausq    sigmasq       phi
"  6.2391" "  0.1146" "  0.8743" "572.2880"
Practical Range with cor=0.05 for asymptotic range: 990.5266


> meuse.con.sph <- likfit(geodata = log.Zn, cov.model = "spherical", ini.cov.pars =
+                  ini.cov, nugget = seq(0,5,by=0.25), fix.nugget = FALSE)
> meuse.con.sph
likfit: estimated model parameters:
       beta       tausq     sigmasq         phi
"   6.1653" "   0.0332" "   0.6961" "1200.5092"
Practical Range with cor=0.05 for asymptotic range: 1200.509

likfit: maximised log-likelihood = -97.88

-------------------------------------------------------------------------------

 # MODEL FITTING WITH DISTANCE TO RIVER AS COVARIATE

> meuse.dist.exp <- likfit(geodata = log.Zn, cov.model = "exponential", ini.cov.pars =
+                   ini.cov, nugget = seq(0,5, by = 0.25), fix.nugget = FALSE,
+                   trend=~dist)
> meuse.dist.exp
likfit: estimated model parameters:
     beta0      beta1      tausq    sigmasq        phi
"  6.5958" " -2.8186" "  0.0309" "  0.2298" "220.8617"
Practical Range with cor=0.05 for asymptotic range: 661.6425
```

```
likfit: maximised log-likelihood = -86.01


> meuse.dist.mat1 <- likfit(geodata = log.Zn, cov.model = "matern", ini.cov.pars =
+                   ini.cov, nugget = seq(0,5, by = 0.25), fix.nugget = FALSE,
+                   kappa = 1.5, fix.kappa = TRUE, trend = ~dist)
> meuse.dist.mat1
likfit: estimated model parameters:
     beta0      beta1      tausq    sigmasq        phi
"  6.5912" " -2.8201" "  0.0742" "  0.1882" "122.9547"
Practical Range with cor=0.05 for asymptotic range: 583.2802

likfit: maximised log-likelihood = -84.78


> meuse.dist.mat2 <- likfit(geodata = log.Zn, cov.model = "matern", ini.cov.pars =
+                   ini.cov, nugget = seq(0,5, by = 0.25), fix.nugget = FALSE,
+                   kappa = 2.5, fix.kappa = TRUE, trend = ~dist)
> meuse.dist.mat2
likfit: estimated model parameters:
    beta0     beta1     tausq   sigmasq       phi
" 6.5853" "-2.8138" " 0.0808" " 0.1793" "90.4478"
Practical Range with cor=0.05 for asymptotic range: 535.3288

likfit: maximised log-likelihood = -84.48


> meuse.dist.gauss <- likfit(geodata = log.Zn, cov.model = "gaussian", ini.cov.pars =
+                    c(0.1882,122.9547), nugget = seq(0,5, by = 0.5), fix.nugget =
+                    FALSE, trend = ~dist)
>  meuse.dist.gauss
likfit: estimated model parameters:
     beta0      beta1      tausq    sigmasq        phi
"  6.5662" " -2.7861" "  0.0873" "  0.1653" "247.1482"
Practical Range with cor=0.05 for asymptotic range: 427.7687

likfit: maximised log-likelihood = -84.28


> meuse.dist.sph <- likfit(geodata = log.Zn, cov.model = "spherical", ini.cov.pars =
+                  ini.cov, nugget = seq(0,5, by = 0.25), fix.nugget = FALSE,
+                  trend = ~dist)
> meuse.dist.sph
likfit: estimated model parameters:
     beta0      beta1      tausq    sigmasq        phi
"  6.6413" " -2.9108" "  0.0660" "  0.2367" "738.3997"
Practical Range with cor=0.05 for asymptotic range: 738.3997

likfit: maximised log-likelihood = -84.71
```

```
            ############################################################
            #                                                          #
            #      PLOTS OF EMPIRICAL AND FITTED SEMIVARIOGRAMS        #
            #                                                          #
            ############################################################


  # Plots of empirical versus the fitted semivariogram models
> plot(c(0,2500), c(0,1.5), type = "p", col = "white", xlab = "distance", ylab =
+ "semivariance")
> semi.con <- variog(log.Zn, option = "bin", pairs.min = 10)
> points(semi.con$u, semi.con$v)
> lines(meuse.con.exp,lty = 5)
> lines(meuse.con.mat1, lty = 3)
> lines(meuse.con.mat2, lty = 7)
> lines(meuse.con.gauss, lty = 5, lwd = 2)
> lines(meuse.con.sph, lty = 7, lwd = 2)

 # Semivariogram of the residuals
> plot(c(0,1500),c(0,0.315), type = "p", col = "white", xlab = "distance", ylab =
+ "semivariance")
> semi.dist <- variog(log.Zn, trend = ~dist, option = "bin", pairs.min = 10)
> points(semi.dist$u, semi.dist$v)
> lines(meuse.dist.exp, lty = 5)
> lines(meuse.dist.mat1, lty = 3)
> lines(meuse.dist.mat2, lty = 7)
> lines(meuse.dist.gauss, lty = 5,lwd = 2)
> lines(meuse.dist.sph, lty = 7, lwd = 2)


            ############################################################
            #                                                          #
            #   PROFILE LIKELIHOODS OF PARAMETERS OF SELECTED MODEL    #
            #                                                          #
            ############################################################

> Proflik.gauss <- proflik(meuse.dist.gauss, geodata = log.Zn, nugget.values =
+                  seq(1e-10,0.2, l =9), sill.values = seq(0.01,0.4, l = 11),
+                  range.values = seq(100,550, l = 23))
> par(mfrow = c(1,3))
> plot(Proflik.gauss)


            ############################################################
            #                                                          #
            #                         KRIGING                          #
            #                                                          #
            ############################################################


> data(meuse.grid)

 # Kriging
```

```
> meuse.pred.gau <- krige.conv(geodata = log.Zn, locations = meuse.grid[,1:2],
+                   krige=krige.control(type.krige="ok", cov.model="gaussian",
+                   cov.pars = c(0.1653,247.1482), nugget = 0.0873, trend.d =
+                   ~meuse$dist, trend.l = ~meuse.grid$dist))

> library(lattice)

  # Creating map of kriging results
> kriging.est <- cbind("Gaussian" = meuse.pred.gau$predict, "Exponential" =
+                meuse.pred.exp$predict)
> krige.est <- as.data.frame(kriging.est)
> krige.est <- SpatialPixelsDataFrame(points = meuse.grid[,1:2], data = krige.est)
> spplot(krige.est, col.regions = terrain.colors(n=100))


              ############################################################
              #                                                          #
              #   COMPUTING ANALYTICAL KRIGING VARIANCE ESTIMATORS       #
              #                                                          #
              ############################################################

> Meuse.analytic.gau <- krige.approx(geodata = log.Zn, trend.d = ~meuse$dist,
+ locations=meuse.grid[,1:2], trend.l= ~meuse.grid$dist, cov.model="gaussian",
+ cov.pars=c(0.1653,247.1482), nugget=0.0873, type.krige="ok")

> summary(Meuse.analytic.gau$variance)
    plug.in       Kacker.Harville    Prasad.Rao
 Min.   :0.1026   Min.   :0.1065   Min.   :0.1090
 1st Qu.:0.1203   1st Qu.:0.1254   1st Qu.:0.1291
 Median :0.1323   Median :0.1364   Median :0.1414
 Mean   :0.1465   Mean   :0.1499   Mean   :0.1534
 3rd Qu.:0.1618   3rd Qu.:0.1635   3rd Qu.:0.1653
 Max.   :0.2712   Max.   :0.2719   Max.   :0.2727


              ############################################################
              #                                                          #
              #        BOOTSTRAP ESTIMATION OF KRIGING VARIANCE          #
              #                                                          #
              ############################################################


> memory.size(4095)

  # Computing unconditional bootstrap estimator
> set.seed(45)
> Meuse.UcondBoot.gau <- krige.uncond(geodata=log.Zn, locations = meuse.grid[,1:2],
+                        type.krige = "ok", cov.model = "gaussian", cov.pars =
+                        c(0.1653,247.1482), nugget = 0.0873, mean = c(6.5662,
+                        -2.7861), trend.d = ~meuse$dist, trend.l =
+                        ~meuse.grid$dist, nsim = 5000, fix.nugget = FALSE, limits =
+                        pars.limits(phi = c(175,400)))

> summary(Meuse.UcondBoot.gau$pred.error[,2])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.09866 0.12220 0.13420 0.14760 0.16250 0.26130


  # Computing conditional bootstrap estimator
> set.seed(45)
> Meuse.CondBoot.gau <- krige.cond(geodata = log.Zn, locations = meuse.grid[,1:2],
+                       trend.l = ~meuse.grid$dist, type.krige = "ok", nsim = 5000,
+                       cov.model = "gaussian", cov.pars = c(0.1653,247.1482),
+                       nugget=0.0873, mean = c(6.5662,-2.7861), trend.d =
+                       ~meuse$dist)


> summary(Meuse.CondBoot.gau$pred.error[,2])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.09887 0.12050 0.13240 0.14650 0.16120 0.27900


              ############################################################
              #                                                          #
              #        MAPPING RESULTS OF THE VARIOUS ESTIMATORS         #
              #                                                          #
              ############################################################

> estimates.gau <- cbind("KH" = Meuse.analytic.gau$variance[,2], "CBS" =
+                   Meuse.CondBoot.gau$pred.errors[,2], "EMSPE" =
+                   Meuse.analytic.gau$variance[,1], "PR" =
+                   Meuse.analytic.gau$variance[,3], "UBS" =
+                   Meuse.UcondBoot.gau$pred.errors[,2])
> estimates.gau <- as.data.frame(estimates.gau)
> estimates.gau <- SpatialPixelsDataFrame(points = meuse.grid[,1:2], data =
+                   estimates.gau)
> spplot(estimates.gau,col.regions = bpy.colors())
```

C.2 Micronutrient Data

```
              ############################################################
              #                                                          #
              #                   INITIALIZING ANALYSIS                  #
              #                                                          #
              ############################################################

 # Loading packages
> library(geoR)

  # Reading data into R : This path will differ depending where the data is
> AEON.file <- read.table("C:\\Documents and Settings\\Administrator\\Desktop\\AEON\\
+              dataset1.txt",header = TRUE)

 # Creating geoR object
> Fe <- as.geodata(obj = AEON.file, coords.col = 2:3, data.col = 5, covar.col = 4)
```

```
              ############################################################
              #                                                          #
              #                EXPLORATORY DATA ANALYSIS                 #
              #                                                          #
              ############################################################

  # Mapping data

> points(Fe, xlab = "east(km)", ylab = "north(km)", cex.min = 0.5,cex.max = 1.25,
+ col = grey(seq(1, 0.1, l = 100)), pt.divide = "data.proportional")
> library(sp)
> Fe.levels <-data.frame(Fe$data)
> Fe.D <- SpatialPointsDataFrame(coords = Fe$coords, data = Fe.levels)
> bubble(Fe.D, xlab = "east(km)", ylab = "north(km)", maxsize = 2, main = "",
+ scales = list(draw = TRUE), do.sqrt = TRUE)

  # graphical EDA of Zn concentrations
> plot(Fe,lowess = TRUE, density = FALSE)

  # Q-Q plot : data
> qqnorm(Fe$data)
> qqline(Fe$data)

 # Log transformation of data
> log.Fe <- as.geodata(obj = AEON.file, coords.col = 2:3, data.col = 5, covar.col =4)
> log.Fe$data <- log(Fe$data)

  # graphical EDA of log(Zn)
> plot(log.Fe, lowess = TRUE, density = FALSE)

  Q-Q plot log transformed data
> qqnorm(log(Fe$data))
> qqline(log(Fe$data))


              ############################################################
              #                                                          #
              #                  PARAMETER ESTIMATION                    #
              #                                                          #
              ############################################################

  # Creating matrix of initial estimates for the partial sill and range parameters
> ini.cov <- expand.grid(seq(0,20, by = 2), seq(1,600, by = 20))

 # MAXIMUM LIKELIHOOD ESTIMATION

> logFe.exp.ml <- likfit(geodata = log.Fe, cov.model = "exponential", ini.cov.pars =
+                 ini.cov, fix.nugget = FALSE, nugget = seq(0,10, by = 2))
> logFe.exp.ml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5080" "0.4340" "0.2031" "0.5086"
Practical Range with cor=0.05 for asymptotic range: 1.523501
```

```
likfit: maximised log-likelihood = -246.4


> logFe.mat1.ml <- likfit(geodata = log.Fe, cov.model = "matern", ini.cov.pars =
+                  c(0.1,0.2), fix.nugget = FALSE, nugget = seq(0,10, by = 2),
+                  kappa = 1.5, fix.kappa = TRUE)
> logFe.mat1.ml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5186" "0.4766" "0.1582" "0.2813"
Practical Range with cor=0.05 for asymptotic range: 1.334495

likfit: maximised log-likelihood = -246.4


> logFe.mat2.ml <-likfit(geodata = log.Fe, cov.model = "matern", ini.cov.pars =
+                 c(0.1,0.2), fix.nugget = FALSE, nugget = seq(0,10, by = 2),
+                 kappa = 2.5, fix.kappa = TRUE)
> logFe.mat2.ml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5223" "0.4834" "0.1504" "0.2105"
Practical Range with cor=0.05 for asymptotic range: 1.246178

likfit: maximised log-likelihood = -246.4


> logFe.sph.ml <- likfit(geodata = log.Fe, cov.model = "spherical", ini.cov.pars =
+                 ini.cov, fix.nugget = FALSE, nugget = seq(0,10,by=2))
>  logFe.sph.ml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5379" "0.4397" "0.1871" "0.9339"
Practical Range with cor=0.05 for asymptotic range: 0.9338806

likfit: maximised log-likelihood = -246.8


> logFe.gauss.ml <- likfit(geodata = log.Fe, cov.model = "gaussian", ini.cov.pars =
+                   c(0.1871, 0.9339), fix.nugget = FALSE, nugget = seq(0,10, by = 2))
> logFe.gauss.ml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5298" "0.4909" "0.1406" "0.6034"
Practical Range with cor=0.05 for asymptotic range: 1.044395

likfit: maximised log-likelihood = -246.5


------------------------------------------------------------------------------
```

```
  # RESTRICTED MAXIMUM LIKELIHOOD ESTIMATION

> logFe.exp.reml <- likfit(geodata = log.Fe, cov.model = "exponential", ini.cov.pars
+                   = ini.cov, fix.nugget = FALSE, nugget = seq(0,10, by=2),
+                   lik.method = "REML")
> logFe.exp.reml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.4907" "0.4474" "0.2080" "0.6494"
Practical Range with cor=0.05 for asymptotic range: 1.945298

likfit: maximised log-likelihood = -244.9


> logFe.mat1.reml <- likfit(geodata = log.Fe, cov.model = "matern", ini.cov.pars =
+                    ini.cov, fix.nugget = FALSE, nugget = seq(0,10, by = 2),
+                    kappa = 1.5, fix.kappa = TRUE, lik.method = "REML")
> logFe.mat1.reml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5111" "0.4817" "0.1650" "0.3138"
Practical Range with cor=0.05 for asymptotic range: 1.488632

likfit: maximised log-likelihood = -245.0


> logFe.mat2.reml <- likfit(geodata = log.Fe, cov.model = "matern", ini.cov.pars =
+                    c(0.1,0.2), fix.nugget = FALSE, nugget = seq(0,10,by = 2),
+                    kappa = 2.5, fix.kappa = TRUE, lik.method = "REML")
> logFe.mat2.reml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5169" "0.4872" "0.1572" "0.2287"
Practical Range with cor=0.05 for asymptotic range: 1.353601

likfit: maximised log-likelihood = -245.1


> logFe.sph.reml <- likfit(geodata = log.Fe, cov.model = "spherical", ini.cov.pars =
+                   ini.cov, fix.nugget = FALSE, nugget = seq(0,10, by = 2),
+                   lik.method = "REML")
> logFe.sph.reml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5277" "0.4665" "0.1740" "1.2882"
Practical Range with cor=0.05 for asymptotic range: 1.288232

likfit: maximised log-likelihood = -245.3


> logFe.gau.reml <- likfit(geodata = log.Fe, cov.model = "gaussian", ini.cov.pars =
+                   c(0.2080, 0.6494), fix.nugget = FALSE, nugget = seq(0,10,by=2),
+                   lik.method = "REML")
```

```
> logFe.gau.reml
likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"2.5268" "0.4937" "0.1464" "0.6374"
Practical Range with cor=0.05 for asymptotic range: 1.103162

likfit: maximised log-likelihood = -245.2


              ############################################################
              #                                                          #
              #             PLOTTING FITTED SEMIVARIOGRAMS               #
              #                                                          #
              ############################################################

 # Models fitted with ML estimation
> plot(emp.vario.logFe <- variog(log.Fe, option = "bin", pairs.min = 10))
> lines(logFe.exp.ml, lty = 5)
> lines(logFe.mat1.ml, lty = 3)
> lines(logFe.mat2.ml, lty = 7)
> lines(logFe.sph.ml, lty = 3, lwd = 2)
> lines(logFe.gauss.ml , lty = 7,lwd = 2)

 # Models fitted with via REML estimation
> plot(emp.vario.logFe <- variog(log.Fe, option = "bin", pairs.min = 10))
> lines(logFe.exp.reml, lty = 5)
> lines(logFe.mat1.reml, lty = 3)
> lines(logFe.mat2.reml, lty = 7)
> lines(logFe.sph.reml, lty=3, lwd = 2)
> lines(logFe.gau.reml, lty = 7, lwd = 2)


              ############################################################
              #                                                          #
              #   PROFILE LIKELIHOODS OF PARAMETERS OF SELECTED MODEL    #
              #                                                          #
              ############################################################

  # Computing profile likelihoods
> prof.ML <- proflik(logFe.exp.ml, geodata = log.Fe, nugget.values = seq(0.1,0.7,
+            l = 15), sill.values = seq(0.01,0.7, l=5), range.values = seq(0.1,3.25,
+            l = 20))

  # Plotting profile likelihoods
> par(mfrow=c(1,3))
> plot(prof.ML)


              ############################################################
              #                                                          #
              #                        KRIGING                           #
              #                                                          #
              ############################################################
```

```
  # Forming prediction grid
> long <- c(seq(26.30708,30.79456, by=.1))
> lat <-  c(seq(-29.03209,-24.38015,by=.1))
> wits.grid <- expand.grid(long,lat)

  # Kriging
> Pred.surf <- krige.conv(geodata = log.Fe, locations = wits.grid, krige =
+              krige.control(obj.model = logFe.exp.ml))

 # Map of kriging surface
> contour(Pred.surf, filled = TRUE, color = terrain.colors, values =
+ Pred.surf$predict)
```

```
              ############################################################
              #                                                          #
              #          CALCULATION OF ANALYTICAL OF ESTIMATES          #
              #                                                          #
              ############################################################
```

```
> Analytic.Fe.ml <- krige.approx(geodata = log.Fe, obj.model = logFe.exp.ml,
+                   locations = wits.grid, type.krige = "ok")

> summary(Analytic.Fe.ml$variance)
    plug.in       Kacker.Harville    Prasad.Rao
 Min.   :0.5109   Min.   :0.5180   Min.   :0.5252
 1st Qu.:0.5440   1st Qu.:0.5508   1st Qu.:0.5575
 Median :0.5534   Median :0.5595   Median :0.5658
 Mean   :0.5574   Mean   :0.5635   Mean   :0.5695
 3rd Qu.:0.5664   3rd Qu.:0.5717   3rd Qu.:0.5772
 Max.   :0.6440   Max.   :0.6476   Max.   :0.6511
```

```
              ############################################################
              #                                                          #
              #          BOOTSTRAP ESTIMATION OF KRIGING VARIANCE        #
              #                                                          #
              ############################################################
```

```
> memory.size(4095)

  # Computing unconditional bootstrap estimator
> set.seed(83)
> Boot.uncond.Fe.ml <- krige.uncond(geodata = Fe, locations = wits.grid, type.krige =
+                      "ok", nsim = 5000, obj.model = logFe.exp.ml, fix.nugget = FALSE,
+                      limits = pars.limits(phi = c(0.2,2.90)))

> summary(Boot.uncond.Fe.ml$pred.errors[,2])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5092  0.5499  0.5612  0.5643  0.5751  0.6561
```

```
 # Computing conditional bootstrap estimator
> set.seed(83)
> Boot.cond.Fe.ml<- krige.cond(geodata = log.Fe, locations = wits.grid, type.krige =
+                   "ok", obj.model = logFe.exp.ml, nsim = 5000)

> summary(Boot.cond.Fe.ml$pred.errors[,2])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4958  0.5415  0.5541  0.5575  0.5703  0.6671


              ##############################################################
              #                                                            #
              #        MAPPING RESULTS OF THE VARIOUS ESTIMATORS           #
              #                                                            #
              ##############################################################

> library(lattice)
> library(gstat)
> aeon <-cbind("KH" = Analytic.Fe.ml $variance[,2], "CBS" = Boot.cond.Fe.ml$
+        pred.errors[,2], "EMSPE" = Analytic.Fe.ml $variance[,1], "PR" =
+        Analytic.Fe.ml$variance[,3], "UBS" = Analytic.Fe.ml$variance[,3])
> aeon <-as.data.frame(aeon)
> aeon <- SpatialPixelsDataFrame(points = wits.grid, data = aeon)
> spplot(aeon, col.regions = bpy.colors())
```