

# ConstrainedKriging: an R-package for customary, constrained and covariance-matching constrained point or block kriging

## Report

### Author(s):

Papritz, Andreas Jürg; Hofer, Christoph

### Publication date:

2012

### Permanent link:

<https://doi.org/10.3929/ethz-a-007366831>

### Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

# constrainedKriging: an R-package for customary, constrained and covariance-matching constrained point or block kriging <sup>1</sup>

Christoph Hofer\*, Andreas Papritz\*

*Institute of Terrestrial Ecosystems, ETH Zurich, Universitätsstrasse 16, 8092 Zurich, Switzerland*

---

## Abstract

The article describes the R package `constrainedKriging`, a tool for spatial prediction problems that involve change of support. The package provides software for spatial interpolation by constrained (CK), covariance-matching constrained (CMCK) and customary universal (UK) kriging. CK and CMCK yield approximately unbiased predictions of nonlinear functionals of target quantities under change of support and are therefore an attractive alternative to conditional Gaussian simulations. The `constrainedKriging` package computes CK, CMCK and UK predictions for points or blocks of arbitrary shape from data observed at points in a two-dimensional survey domain. Predictions are computed for a random process model that involves a nonstationary mean function (modelled by a linear regression) and a weakly stationary, isotropic covariance function (or variogram). CK, CMCK and UK require the point-block and block-block averages of the covariance function if the prediction targets are blocks. The `constrainedKriging` package uses numerically efficient approximations to compute these averages. The article contains, apart from a brief summary of CK and CMCK, a detailed description of the algorithm used to compute the point-block and block-block covariances, and it describes the functionality of the software in detail. The practical use of the package is illustrated by a comparison of universal and constrained lognormal block kriging for the Meuse Bank heavy metal data set.

*Keywords:* Constrained kriging, Covariance-matching constrained kriging, Universal kriging, Change of support, Block kriging, Block-block covariance

---

## 1. Introduction

Change-of-support problems are quite common in practical applications of geostatistical methods (Gotway and Young, 2002). In surveys of soil contamination, as an example, one has often to

---

<sup>1</sup>NOTICE: This is the author's version of a work that was accepted for publication in *Computers & Geosciences*. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms, may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *Computers & Geosciences*, *constrainedKriging: An R-package for customary, constrained and covariance-matching constrained point or block kriging*, VOLUME 37, ISSUE 10, 2011, DOI:10.1016/j.cageo.2011.02.009.

\*Corresponding author

*Email addresses:* christoph.hofer@alumni.ethz.ch (Christoph Hofer), andreas.papritz@env.ethz.ch (Andreas Papritz)

*Preprint submitted to Geoderma*

*September 28, 2012*

predict the mean pollutant content of whole parcels of land from measurements with quasi-point support. Predictions of mean values are required because remediation or regulatory measures are imposed for entire parcels of land and not for areal units with the same support as the observations (Papritz et al., 2005). Predicting the mean ore content of larger volumes of rock in metal mining is another, classical, example for change of support. Typically, the target areas or volumes — in the sequel we denote them as blocks — for which predictions are required, are several orders of magnitude smaller than the survey domain, but at the same time distinctly larger than the support of the measurements. This local change of support (Chilès and Delfiner, 1999, pp. 378–380) has to be considered if one wishes to predict nonlinear functionals of block means (e.g., whether the block mean exceeds some regulatory threshold in soil pollution surveying).

Conditional Gaussian simulations (CS, e.g. Chilès and Delfiner, 1999, chap. 7) are the popular approach to nonlinear prediction problems with local change of support. But CS is not undisputed: Aldworth and Cressie (2003) pointed out that CS is highly parametric and that predictions of nonlinear functionals of block means are (badly) biased if the probabilistic model is misspecified. Moreover, for large problems, CS is still demanding in terms of computing time and storage requirements.

Constrained kriging (CK, Cressie, 1993a) is an alternative to CS for nonlinear prediction problems with change of support. The linear CK predictor satisfies in addition to the unbiasedness constraints of universal kriging (UK, e.g. Cressie, 1993b, sec. 3.4, also denoted as kriging with trend) a further constraint that matches the variances of the predictions to the variances of the block means. Aldworth and Cressie (2003) extended CK to covariance-matching constrained kriging (CMCK), which matches for a set of blocks both the variances and covariances of predictions and block means. From theory we expect that predictions of nonlinear functionals of block means by CK and CMCK are less biased than by UK and exactly unbiased if the spatial variable is Gaussian.

To explore the practical advantages of CK and CMCK, Hofer and Papritz (2010) compared the performance of CS, CK, CMCK and UK by simulations for nonstationary Gaussian and positively skewed spatial process models. Besides prediction of block means (linear prediction problem), they studied the prediction of binary indicators that signal if the block means exceed critical thresholds (nonlinear prediction problem). All four methods predicted the blocks means well. CK and CMCK gave less precise predictions than UK and CS only if sampling was sparse. When predicting the indicators, CK and CMCK showed the best performance, except in the Gaussian case where threshold exceedance was best predicted by the conditional quantiles of CS. However, CS failed for the positively skewed data when the model was misspecified. Overall, CK showed a good performance, slightly better than CMCK. Apart from its favourable statistical properties, CK is also simple to compute and does not require more computing resources than UK.

The programming environment R (R Development Core Team, 2010) comprises several packages for the geostatistical analysis of spatial data. Most of these packages contain software to compute UK point predictions. To our knowledge, only the R package `gstat` (Pebesma, 2004) has a function for block UK. We are not aware of any R package (or any other freely available software) for CK and CMCK. We closed this gap by the R package `constrainedKriging` that we describe in this article. The `constrainedKriging` package computes CK, CMCK and UK predictions for points or blocks of arbitrary shape from data observed at points in a two-dimensional survey domain. The predictions are computed for a spatial random process model that uses a linear regression model for the nonstationary mean function and a weakly stationary, isotropic covariance function (or variogram) to model the residual autocorrelation.

The next section summarizes some theory about CK and CMCK, a more comprehensive exposition can be found in [Aldworth and Cressie \(2003\)](#) and in [Hofer and Papritz \(2010\)](#). In section 3 we describe how we implemented CK and CMCK. In particular, we discuss how we efficiently integrate the covariance function for predicting block means, and we summarize the functionality of the constrainedKriging package. An application of the software is presented in section 4. We use data on heavy metals in a floodplain along the river Meuse (Meuse Bank data set, [Burrough and McDonnell, 1998](#)). The article concludes in section 5 with a summary of the main findings.

## 2. Constrained and covariance-matching constrained kriging

We consider  $m$  blocks,  $B_1, \dots, B_m$ , in a two-dimensional domain  $D$ . The area of a block is denoted by  $|B_i|$ ;  $\mathbf{s} = (x, y)'$  is a location in  $D$  ( $'$  denotes transpose). The target quantities are the block means (or some nonlinear transform thereof)

$$Y(B_i) = \frac{1}{|B_i|} \int_{B_i} Y(\mathbf{s}) \, d\mathbf{s}$$

of a variable of interest

$$Y(\mathbf{s}) = \mathbf{x}(\mathbf{s})' \boldsymbol{\beta} + \delta(\mathbf{s}),$$

with expectation  $E[Y(\mathbf{s})] = \mathbf{x}(\mathbf{s})' \boldsymbol{\beta}$  ( $\mathbf{x}(\mathbf{s})$  and  $\boldsymbol{\beta}$  are  $p$ -vectors with the covariates for location  $\mathbf{s}$  and the regression coefficients, respectively).  $\delta(\mathbf{s})$  is a zero mean, weakly stationary variable with an isotropic covariance function

$$\text{Cov}[\delta(\mathbf{s}), \delta(\mathbf{s} + \mathbf{h})] = \text{Cov}[Y(\mathbf{s}), Y(\mathbf{s} + \mathbf{h})] = C(h),$$

where  $h = |\mathbf{h}|$ .

We have “noisy” measurements,  $Z(\mathbf{s}_i) = Y(\mathbf{s}_i) + \varepsilon_i$ , of the variable of interest from  $n$  locations, which we denote by  $\mathbf{Z} = (Z(\mathbf{s}_1), \dots, Z(\mathbf{s}_n))'$ .  $\varepsilon_i$  is a zero mean white noise variable with variance  $\sigma_\varepsilon^2$ . Let  $\mathbf{Y}_m = (Y(B_1), \dots, Y(B_m))'$  denote the vector with the  $m$  block means. We look for an approximately unbiased prediction of  $g(\mathbf{Y}_m)$ , where  $g(\cdot)$  is a (possibly nonlinear) scalar function of  $\mathbf{Y}_m$ . Notice that we are interested in predicting functionals of the “noise-free”  $Y(B_i)$ . We consider predictors of the form  $g(\hat{\mathbf{Y}}_m)$ , where  $\hat{\mathbf{Y}}_m = \Lambda' \mathbf{Z}$  is a linear predictor of  $\mathbf{Y}_m$  and  $\Lambda = (\lambda_1, \dots, \lambda_m)$  is a  $n \times m$  matrix of weights.

The covariance-matching constrained kriging predictor,  $\hat{\mathbf{Y}}_{\text{CMCK}}$ , proposed by [Aldworth and Cressie \(2003\)](#), is a linear predictor, having the property that for any smooth, nonlinear  $g(\cdot)$   $g(\hat{\mathbf{Y}}_{\text{CMCK}})$  is approximately unbiased for  $g(\mathbf{Y}_m)$  and exactly unbiased if  $\mathbf{Y}_m$  is Gaussian ([Aldworth and Cressie, 2003](#), p. 12). The CMCK predictor of  $\mathbf{Y}_m$  is given by

$$\hat{\mathbf{Y}}_{\text{CMCK}} = \mathbf{X}_m \hat{\boldsymbol{\beta}}_{\text{GLS}} + \mathbf{K}' \mathbf{C}' \boldsymbol{\Sigma}^{-1} (\mathbf{Z} - \mathbf{X} \hat{\boldsymbol{\beta}}_{\text{GLS}}), \quad (1)$$

where  $\mathbf{K} = \mathbf{Q}_1^{-1} \mathbf{P}_1$  is a  $m \times m$  matrix;  $\mathbf{Q}_1$  and  $\mathbf{P}_1$  are symmetric  $m \times m$  matrices given by

$$\mathbf{Q}_1 \mathbf{Q}_1 = \mathbf{Q} = \mathbf{C}' (\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1} \mathbf{X} (\mathbf{X}' \boldsymbol{\Sigma}^{-1} \mathbf{X})^{-1} \mathbf{X}' \boldsymbol{\Sigma}^{-1}) \mathbf{C}, \quad (2)$$

$$\mathbf{P}_1 \mathbf{P}_1 = \mathbf{P} = \boldsymbol{\Sigma}_m - \mathbf{X}_m' (\mathbf{X}' \boldsymbol{\Sigma}^{-1} \mathbf{X})^{-1} \mathbf{X}_m; \quad (3)$$

$\widehat{\boldsymbol{\beta}}_{\text{GLS}} = (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{Z}$  is the  $p$ -vector with the generalised least squares (GLS) estimate of  $\boldsymbol{\beta}$ ;  $\mathbf{C} = (\mathbf{c}(\mathbf{s}_{1\dots n}, B_1), \dots, \mathbf{c}(\mathbf{s}_{1\dots n}, B_m))$  is a  $n \times m$  matrix where  $\mathbf{c}(\mathbf{s}_{1\dots n}, B_i) = (C(\mathbf{s}_1, B_i), \dots, C(\mathbf{s}_n, B_i))'$  are the  $n$  covariances between  $\mathbf{Z}$  and  $Y(B_i)$ ;  $\boldsymbol{\Sigma} = \text{Cov}[\mathbf{Z}, \mathbf{Z}']$ ;  $\boldsymbol{\Sigma}_m = \text{Cov}[\mathbf{Y}_m, \mathbf{Y}_m']$ ;  $\mathbf{X}_m = (\mathbf{x}(B_1), \dots, \mathbf{x}(B_m))'$  and  $\mathbf{X} = (\mathbf{x}(\mathbf{s}_1), \dots, \mathbf{x}(\mathbf{s}_n))'$  are the  $m \times p$  and  $n \times p$  design matrices of the blocks and observations, respectively. The matrices  $\mathbf{Q}_1$  and  $\mathbf{P}_1$  exist and are positive definite if  $\mathbf{Q}$  and  $\mathbf{P}$  are themselves positive definite. In practice, the CMCK predictor thus exists if  $\mathbf{Q}$  and  $\mathbf{P}$  are positive definite. Unlike  $\mathbf{Q}$ , which is always nonnegative definite, the matrix  $\mathbf{P}$  may become negative definite (see [Aldworth and Cressie, 2003](#), p. 15, [Hofer and Papritz, 2010](#), p. 635 for details).

For  $m = 1$  the CMCK predictor simplifies to the CK predictor

$$\widehat{Y}_{\text{CK}}(B_1) = \mathbf{x}(B_1)' \widehat{\boldsymbol{\beta}}_{\text{GLS}} + \mathbf{K} \mathbf{c}(\mathbf{s}_{1\dots n}, B_1)' \boldsymbol{\Sigma}^{-1} (\mathbf{Z} - \mathbf{X} \widehat{\boldsymbol{\beta}}_{\text{GLS}}), \quad (4)$$

and  $\mathbf{K}$  reduces to the scalar

$$K = \left( \frac{\text{Var}[Y(B_1)] - \mathbf{x}(B_1)' (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1} \mathbf{x}(B_1)}{\mathbf{c}(\mathbf{s}_{1\dots n}, B_1)' (\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1} \mathbf{X} (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1} \mathbf{X}' \boldsymbol{\Sigma}^{-1}) \mathbf{c}(\mathbf{s}_{1\dots n}, B_1)} \right)^{1/2} = \left( \frac{P}{Q} \right)^{1/2}. \quad (5)$$

The constraints that the (co-)variances of  $\mathbf{Y}_m$  and  $\widehat{\mathbf{Y}}_m$  match, result in larger mean square errors (MSPE) of the CK and CMCK compared to the UK predictor

$$\text{MSPE}[\widehat{\mathbf{Y}}_{\text{CMCK}}] = \text{MSPE}[\widehat{\mathbf{Y}}_{\text{UK}}] + (\mathbf{P}_1 - \mathbf{Q}_1)(\mathbf{P}_1 - \mathbf{Q}_1), \quad (6)$$

and

$$\text{MSPE}[\widehat{Y}_{\text{CK}}(B_1)] = \text{MSPE}[\widehat{Y}_{\text{UK}}(B_1)] + (\sqrt{P} - \sqrt{Q})^2, \quad (7)$$

where

$$\begin{aligned} \text{MSPE}[\widehat{\mathbf{Y}}_{\text{UK}}] &= \text{Cov}[(\widehat{\mathbf{Y}}_{\text{UK}} - \mathbf{Y}), (\widehat{\mathbf{Y}}_{\text{UK}} - \mathbf{Y})'] \\ &= \boldsymbol{\Sigma}_m - \mathbf{C}' \boldsymbol{\Sigma}^{-1} \mathbf{C} + \\ &\quad (\mathbf{X}'_m - \mathbf{X}' \boldsymbol{\Sigma}^{-1} \mathbf{C})' (\mathbf{X}' \boldsymbol{\Sigma}^{-1} \mathbf{X})^{-1} (\mathbf{X}'_m - \mathbf{X}' \boldsymbol{\Sigma}^{-1} \mathbf{C}). \end{aligned} \quad (8)$$

is the covariance matrix of the UK prediction errors.

### 3. Software

The `constrainedKriging` package computes univariate UK, CK and CMCK predictions for points or blocks in a two-dimensional domain from data that can be represented by a spatial random process with a nonstationary mean function (modelled by a linear regression) and an isotropic, weakly stationary covariance function. Currently, our software does not allow to use unbounded variograms. This may appear as a limitation, but our experience is that variograms are hardly ever unbounded if the large scale trend of a spatial variable is adequately modelled by a linear regression. The uncertainty of the covariance parameters is ignored when computing the predictions, i.e. our software computes so-called “plug-in” predictions. For all kinds of block kriging, the “block-block”-averages (elements of  $\boldsymbol{\Sigma}_m$ , in the sequel denoted as block-block covariances) and “point-block”-averages (elements of  $\mathbf{C}$ , point-block covariances in the remainder) of the covariance function are required. Computing these averages is the numerically costly step in block

kriging. Apart from some general remarks about our implementation of CK and CMCK, we describe in this section therefore, how we compute these averages in an efficient way. The second part of the section describes the functionality of the software.

### 3.1. Implementation

The CK and CMCK algorithms are closely related to UK. The only difference is that the weighted sum of the GLS residuals is multiplied by  $\mathbf{K}'$  for CMCK (1) and by  $K$  for CK (4).  $K$  is obtained from the terms required to calculate the UK predictor and its MSPE. Thus, the CK predictor can be computed with negligible cost from the standard (block) UK results. For CMCK we have to compute the nonnegative definite, symmetric  $m \times m$  matrices  $\mathbf{Q}_1$  (2) and  $\mathbf{P}_1$  (3) — which in turn depend on  $\mathbf{C}$  and  $\Sigma_m$  — to get  $\mathbf{K}$ .  $\mathbf{Q}_1$  and  $\mathbf{P}_1$  are obtained by eigenvalue decomposition of  $\mathbf{Q}$  and  $\mathbf{P}$  (Harville, 1997, sec. 21.9): Let  $\mathbf{V}$  be a nonnegative definite  $m \times m$  matrix with eigenvalue decomposition  $\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{U}'$  where the columns of  $\mathbf{U}$  contain the eigenvectors and  $\mathbf{D} = \text{diag}(d_1, \dots, d_m)$  the  $m$  eigenvalues,  $d_i$ , of  $\mathbf{V}$ . The nonnegative symmetric square root,  $\mathbf{V}_1$ , of  $\mathbf{V}$  such that  $\mathbf{V} = \mathbf{V}_1\mathbf{V}_1 = \mathbf{V}_1^2$ , is then given by  $\mathbf{V}_1 = \mathbf{U}\text{diag}(\sqrt{d_1}, \dots, \sqrt{d_m})\mathbf{U}'$ . If either  $\mathbf{P}$  or  $\mathbf{Q}$  are not strictly positive definite, then some eigenvalues are either negative ( $\mathbf{P}$ ) or equal to zero ( $\mathbf{Q}$ ) and in both instances the decomposition fails. Our software then issues a warning and returns NA as prediction result.

It remains to explain how we compute  $\Sigma_m$  and  $\mathbf{C}$ . Our software computes them only for isotropic covariances. To cope with geometrically anisotropic autocorrelation one could transform the geographical coordinates beforehand by an affine transformation. Zonal anisotropies, however, cannot be currently handled. We believe that this is not too serious a limitation because anisotropic autocorrelation patterns of spatial data are frequently linked to a large-scale spatial trend and often disappear in the residuals of a linear regression model.  $\mathbf{C}$  is required in all three approaches; UK and CK require only the diagonal elements of  $\Sigma_m$ , but CMCK needs also the off-diagonal elements of  $\Sigma_m$ . In the explanation we consider a set of  $m = 4$  blocks (Fig. 1).  $B_1$  is the target block,  $B_2$ ,  $B_3$  and  $B_4$  are its “neighbours” used for matching the (co-)variances of predictions and block means in CMCK. Jointly they form a so-called “prediction set”, abbreviated in the sequel by PS. The blocks of a prediction set usually share one or more edge(s) with the target block, cf. Bivand et al. (2008, sec. 9.2) for a discussion how to define spatial “neighbours”. If the prediction target is a point we can of course define prediction sets in the same way.

The block-block covariance between two blocks  $B_i$  and  $B_j$  of arbitrary shape is given by

$$[\Sigma_m]_{ij} = \text{Cov}[Y(B_i), Y(B_j)] = \frac{1}{|B_i||B_j|} \int_{B_i} \int_{B_j} C(|\mathbf{s}_i - \mathbf{s}_j|) d\mathbf{s}_i d\mathbf{s}_j. \quad (9)$$

We can reduce this quadruple integral to a single integral by exploiting the fact that it can be written as the expected value of  $C(|\mathbf{s}_i - \mathbf{s}_j|)$  with respect to the distribution of the distance  $|\mathbf{s}_i - \mathbf{s}_j|$  between two points,  $\mathbf{s}_i$  and  $\mathbf{s}_j$ , uniformly distributed in  $B_i$  and  $B_j$ , respectively

$$\text{Cov}[Y(B_i), Y(B_j)] = \int_{-\infty}^{+\infty} u(|\mathbf{s}_i - \mathbf{s}_j|) C(|\mathbf{s}_i - \mathbf{s}_j|) d|\mathbf{s}_i - \mathbf{s}_j|. \quad (10)$$

$u(|\mathbf{s}_i - \mathbf{s}_j|)$  is the probability density function of the distance between the two points. This density is hard to derive for arbitrary block shapes, but Gosh (1951) deduced it for two rectangular blocks with the same shape and size. Based on this result, Clifford (2005) wrote the R package `spatialCovariance` (Clifford, 2009) to compute the spatial covariance between two rectangular blocks with the same geometry.

We use [Clifford](#)'s package to approximate  $\text{Cov}[Y(B_i), Y(B_j)]$  for blocks of arbitrary shape: The key idea is to approximate the blocks by sets of rectangular "pixels", to compute the covariance matrix of the pixel means by [Clifford](#)'s software and to approximate  $\text{Cov}[Y(B_i), Y(B_j)]$  from the latter matrix. Our algorithm consists of the following steps:

1. Find the PS with the largest bounding box among all PSs.
2. Place a grid with  $q$  rectangular pixels,  $PX_k$ ,  $k = 1, \dots, q$ , that have all the same size and shape over the largest bounding box (Fig. 2, left). The dimension of the pixel should be chosen such that the pixel is distinctly larger than the support of the observations, but small enough to allow a good approximation of the blocks.
3. Compute the covariance matrix of the  $q$  pixel means,  $Y(PX_k) = 1/|PX| \int_{PX_k} Y(\mathbf{s}) \, d\mathbf{s}$ , by the R package `spatialCovariance`.
4. Approximate the blocks of each PS by a subset of the  $q$  pixels. Figure 2 (right) illustrates the approximation for our example PS shown in Figure 1. The approximation of the block  $B_i$  is denoted by  $\tilde{B}_i$ . The pixels are allocated to the blocks according to the following rules:
  - a) Blocks that do not contain pixel centroids are treated as points.
  - b) A pixel is allocated to that block of the PS that contains its centroid, provided the area of the respective block, say  $B_s$ , is not smaller than the area of the pixel.
  - c) If, however,  $|B_s| < |PX|$  then the block  $B_s$  is treated as a point and the pixel, whose centroid lies in  $B_s$ , is allocated to the block  $B_j$ ,  $B_j \in \text{PS}$ ,  $j \neq s$ , with whom it shares the largest area. The intersection of the areas of pixels and blocks are computed by the R package `gpplib` ([Peng et al., 2010](#)).
- 5 Compute for each PS the covariance matrix of the  $Y(\tilde{B}_i)$  from the covariance matrix of the  $Y(PX_k)$  by

$$\text{Cov}[Y(B_i), Y(B_j)] \approx \text{Cov}[Y(\tilde{B}_i), Y(\tilde{B}_j)] = \frac{1}{q_{ij}^2} \sum_{k=1}^{q_{ij}} \sum_{l=1}^{q_{ij}} \text{Cov}[Y(PX_k), Y(PX_l)], \quad (11)$$

where  $q_{ij}$  is the number of pixels used to approximate  $B_i$  and  $B_j$ .

Steps 1–3 are carried out just once, but steps 4 and 5 must be executed for each PS. Our allocation rule for the pixels in step 4 is a compromise that ensures a computationally efficient yet precise approximation of  $\text{Cov}[Y(B_i), Y(B_j)]$ .

The elements of  $\mathbf{C}$  are approximated in a similar way: Without loss of generality, we can represent a point-block covariance by a single integral

$$\begin{aligned} [\mathbf{C}]_{ki} = \text{Cov}[Y(\mathbf{s}_k), Y(B_i)] &= \frac{1}{|B_i|} \int_{B_i} C(|\mathbf{s}_k - \mathbf{s}_i|) \, d\mathbf{s}_i \\ &= \int_{-\infty}^{+\infty} v(|\mathbf{s}_k - \mathbf{s}_i|) C(|\mathbf{s}_k - \mathbf{s}_i|) \, d|\mathbf{s}_k - \mathbf{s}_i|, \end{aligned} \quad (12)$$

where  $v(|\mathbf{s}_k - \mathbf{s}_i|)$  is the probability density function of the distance  $|\mathbf{s}_k - \mathbf{s}_i|$  between the sampling location  $\mathbf{s}_k$  and a point  $\mathbf{s}_i$ , uniformly distributed in  $B_i$ . We approximate  $B_i$  again by a set of pixels, yielding  $\tilde{B}_i$ , cf. step 4 above, and we approximate the point-block covariance in the modified step 5 by

$$\text{Cov}[Y(\mathbf{s}_k), Y(B_i)] \approx \text{Cov}[Y(\mathbf{s}_k), Y(\tilde{B}_i)] = \frac{1}{q_i} \sum_{l=1}^{q_i} \text{Cov}[Y(\mathbf{s}_k), Y(PX_l)], \quad (13)$$

where  $q_i$  is the number of pixels that make up  $\tilde{B}_i$ . To compute  $\text{Cov}[Y(\mathbf{s}_k), Y(PX_i)]$  we derived the density function of the distance between a fixed point, say the origin  $\mathbf{s}_0$ , and a point  $\mathbf{s}$ , uniformly distributed in a rectangle (cf. [Appendix A](#)), and we implemented the integration in a C function that is callable from R. This function relies on `CovarianceFct()`, a function of the R package `RandomFields` ([Schlather, 2001](#)) that we also use to compute the covariance matrix of the pixel means in step 3 and for point kriging.

To reduce the approximation error in (11) and (13), the software allows to average for each PS the results of several runs of the steps 2–5. Each run uses a randomly shifted origin for the grid of pixels in step 2. The origin is shifted by  $(\Delta x, \Delta y)'$  where  $\Delta x$  ( $\Delta y$ ) is uniformly distributed in the interval  $\pm 0.5$  pixel-length ( $\pm 0.5$  pixel-width).

### 3.2. Functionality

The `constrainedKriging` package provides two main user functions, `preCKrige()` and `CKrige()`:

- `preCKrige()` computes the block-block covariances of the PSs.
- `CKrige()` computes the point-block covariances and the predictions based on the output generated by `preCKrige()`.

The purpose of the first step is to supply the block-block covariance matrices and the information about the PS block approximations. `preCKrige()` requires as input (i) the coordinates and covariates of the prediction targets, (ii) the definition of the neighbourhood relations in each PS, and (iii) the parameters of the covariance function. For block prediction the function requires in addition the dimension (width, length) of the pixel for the numerical integration. The coordinates and covariates of the prediction targets are passed to `preCKrige()` as a `SpatialPointsDataFrame` (point kriging) or as a `SpatialPolygonsDataFrame` (block kriging), two classes provided by the R package `sp` ([Bivand et al., 2008](#), chap. 2). By default, `preCKrige()` computes the items required for UK and CK, i.e., it assumes that the prediction sets consist only of the target block (case  $m = 1$ ). For CMCK the prediction sets must be defined by a list that has as many components as there are prediction targets (the  $i$ th component contains a vector with the indices of the neighbourhood relations of the  $i$ th prediction set).

The covariance parameters are passed to `preCKrige()` as a list that is conveniently generated by the auxiliary user function `covmodel()`, which is also part of our package. The covariance function may consist of a single “elementary” function such as exponential, spherical, etc., or of a sum of several such elementary functions. Each elementary function is characterized by the following parameters: Type of model (e.g., exponential), variance of measurement errors ( $\sigma_\epsilon^2$ ), variance of spatial micro-scale variation (nugget variance of  $\delta(\mathbf{s})$ ), variance of autocorrelated component (partial sill of  $\delta(\mathbf{s})$ ), range parameter, and possibly further parameters such as the “smoothness” parameter of Matern’s Bessel function covariance model. Notice that `covmodel()` takes two “nugget” constants: one for the variance of the measurement errors that are filtered in kriging and one for the spatial micro-scale variation that is considered when computing the kriging predictions. Finally the user can choose the number of runs to approximate the block-block and point-block covariances of the PSs. By default, only one run is executed.

A call of `preCKrige()` generates a S4 class object ([Chambers, 1998](#), chap. 7). The returned object is either of class `preCKrigePoints` (point kriging) or of class `preCKrigePolygons` (block kriging), two classes defined by our package. Four elements (slots) are common to both classes: A list with the block-block covariances of the PSs, a list with the covariance parameters, a matrix with the coordinates of the prediction points (`preCKrigePoints`) or a list with the coordinates of



the blocks (`preCKrigePolygons`), and a data frame with the covariates for the prediction targets. A `preCKrigePoints` object contains an additional slot with a list of the “neighbours” of each target point (the list is empty for UK and CK). For block kriging, the information about “neighbours” is stored in another slot that lists (as an R list) per PS the number of blocks and the information about the approximation of the blocks by the pixels. A `preCKrigePolygons` object has one more slot if the block-block covariances are approximated in multiple runs: The slot contains a list of matrices with the standard errors of the block-block covariances of each PS as computed from the replicated runs of the approximation algorithm. Our package provides `print()` and `summary()` methods for the classes `preCKrigePoints` and `preCKrigePolygons` and for the latter also a `plot()` method to display the approximation of the blocks by the pixels (cf. Fig. 2).

The second step is to compute the kriging predictions by a call of `CKrige()`. By default, `CKrige()` computes CK predictions. If UK or CMCK predictions are requested then the kriging method must be explicitly specified. `CKrige()` requires as input (i) the object generated before by `preCKrige()`, (ii) a data frame with the coordinates of the sampling sites, the response variable and the covariates, and (iii) two R formulae (Chambers and Hastie, 1992), one specifying a linear regression model for the mean function of the response variable, the other (one-sided) the coordinates of the sampling sites. Note that `CKrige()` computes the predictions by including all observations, i.e., no local search windows are used.

By default, `CKrige()` returns either a `SpatialPointsDataFrame` (point kriging) or `SpatialPolygonsDataFrame` (block kriging). Recall that the type of kriging (point or block) is controlled by the class of the object generated before by `preCKrige()` (`preCKrigePoints` or `preCKrigePolygons`). The object generated by `CKrige()` contains always the coordinates of the prediction targets and a data frame with the kriging results. No matter which kriging method has been selected, the data frame has a column with the kriging predictions and a column with the root mean square prediction errors (kriging standard errors). For CK, the data frame contains in addition columns for  $\sqrt{P}$ ,  $\sqrt{Q}$  and  $K$ . For CMCK, the data frame has instead columns with the first diagonal elements of  $\mathbf{P}_1$ ,  $\mathbf{Q}_1$  and  $\mathbf{K}$ , which all refer to the target point or block (we use the convention that the first block or point in a PS is the prediction target). `CKrige()` offers the possibility to generate an extended output. This is useful if the kriging results are further processed, e.g., in lognormal kriging for the backtransformation to the original scale of the measurements. The extended output object is a list that has as components the `CKrige()` default output, the GLS estimates, their covariance matrix, the GLS residuals and the simple kriging weights  $\Sigma^{-1}\mathbf{C}$  plus further items. We refer the reader to the help page of `CKrige()` for a complete description of the structure of the extended output.

The graphical display of the kriging results is straightforward as the generic function `splot()` of the `sp` package (Bivand et al., 2008, sec. 3.2) provides `plot()` methods for the default `CKrige()` output objects. In addition, our packages provides `print()` and `summary()` methods for the extended output of `CKrige()`.

### 3.3. Availability

The `constrainedKriging` package can be obtained from the comprehensive R archive network (CRAN) and its mirrors (<http://cran.r-project.org/web/packages/>).

## 4. Application

We illustrate the use of the `constrainedKriging` package by an example that uses a data set on heavy metals in the topsoil of a floodplain along the river Meuse in the Netherlands (Burrough

and McDonnell, 1998). This so-called Meuse Bank data set is part of the `sp` package, and it is heavily used in Bivand et al. (2008) and in the R package `gstat` (Pebesma, 2004).

Provided the `constrainedKriging` package is installed, the package and the data sets required by the example are loaded by

```
> require(constrainedKriging); data(meuse,meuse.blocks)
```

where `>` is the R prompt, `meuse` is the Meuse Bank data set and `meuse.blocks` is a `SpatialPolygonsDataFrame` supplied by our package, which contains the coordinates of 259 blocks, arbitrarily defined by putting a grid with 150 m mesh width over the floodplain area, and a  $259 \times 2$  data frame with the covariate `dist` (mean Euclidean distance of the blocks from the river, normalized to the interval  $[0, 1]$ ) and the attribute `M`, the (spatial) variance of the mean distance between points, uniformly distributed within the blocks, and the river. `M` is required for the backtransformation in lognormal CK and UK under the assumption of permanence of lognormality (cf. below and Cressie, 2006). Figure 3 shows the position of the 259 blocks.

Suppose we aim to predict the mean topsoil zinc concentration of the blocks by CK and compare the CK with the UK predictions. Following Bivand et al. (2008), we use the regression model  $\log(\text{zinc}(\mathbf{s})) \sim \sqrt{\text{dist}(\mathbf{s})}$  as trend function, where `zinc(s)` is the zinc concentration in the topsoil at location `s` and `dist(s)` is the mean Euclidean distance between `s` and the river. The autocorrelation of the regression errors is modelled by an exponential covariance function with nugget = 0.05, partial sill = 0.15 and range = 192.5. The parameters of the covariance function were estimated by restricted maximum likelihood, using the function `gls(.)` of the R package `nlme` (Pinheiro and Bates, 2000, sec. 5.4). Since we have no information about the precision of the measurements, we assume that the nugget reflects micro-scale variation, and we will therefore set  $\sigma_\epsilon^2 = 0$ .

The size of the pixel, used in the approximation of the blocks, should be chosen larger than the support of the observations, but small enough to allow a reasonable approximation of the blocks. The support of the observations is about  $15 \text{ m} \times 15 \text{ m}$ , and the area of the blocks varies between  $707 \text{ m}^2$  and  $22500 \text{ m}^2$  (67.6 % of the blocks are squares with the maximum support). Given these dimensions, we choose a  $75 \text{ m} \times 75 \text{ m}$  square pixel to approximate the blocks. We are now ready to call `preCKrige(.)` to compute the block-block covariances

```
> preCK=preCKrige(newdata=meuse.blocks,model=
                  covmodel("exponential",0,0.05,0.15,192.5),
                  pwidth=75,pheight=75)
```

and we store the output in the object `preCK`. Notice the use of the auxiliary function `covmodel(.)` to pass the covariance parameters to `preCKrige(.)`. Then, we call `CKrige(.)`

```
> CK=CKrige(formula=log(zinc)~sqrt(dist),data=meuse,
             locations=~x+y,object=preCK,ex.out=TRUE)
```

to obtain the CK predictions of the mean log-transformed zinc content for the blocks. By default, `CKrige(.)` computes CK predictions (argument: `method=2`). To compute UK (CMCK) predictions, one has to specify `method=1` (`method=3`) explicitly. The first formula,  $\log(\text{zinc}) \sim \sqrt{\text{dist}}$ , specifies the trend model, and the second, `~x+y`, the coordinates of the sampling locations (the variables `zinc`, `dist`, `x` and `y` are contained in the data frame `meuse`). Note that we request extended output because we need some of its items for the backtransformation of the predictions to the original scale of the measurements.

Assuming a trend function modelled by a linear regression and permanence of lognormality, [Cressie \(2006\)](#) showed that an approximately unbiased backtransformation of a generic predictor, say  $\check{Y}(B)$ , of the target quantity on the log-scale is given by

$$\check{T}(B) = \exp\left(\check{Y}(B) + 1/2\{C(0) + \beta' \mathbf{M}(B) \beta - \text{Var}[\check{Y}(B)]\}\right), \quad (14)$$

where

$$\mathbf{M}(B) = \frac{1}{|B|} \int_B (\mathbf{x}(s) - \mathbf{x}(B)) (\mathbf{x}(s) - \mathbf{x}(B))' ds$$

is the (spatial) covariance matrix of the covariates for a point  $\mathbf{s}$  uniformly distributed in  $B$ , and  $C(0)$  is the variance of  $Y(\mathbf{s})$ . For CK, we have by definition  $\text{Var}[\hat{Y}_{\text{CK}}(B)] = \text{Var}[Y(B)]$ . Furthermore,  $\mathbf{M}(B)$  simplifies in our example to  $\mathbf{M}(B) = M = \text{Var}_B[\sqrt{\text{dist}(\mathbf{s} \in B)}$  as the intercept,  $\beta_0$ , of a linear regression model is support independent ( $\beta_0(\mathbf{s}) = \beta_0(B)$ ) ( $\text{Var}_B[\cdot]$  denotes here the variance based on a uniform random location in  $B$ ). In practice we substitute  $\hat{\beta}_{\text{GLS}}$  for  $\beta$ , hence the backtransformation for CK is given in our example by

$$\hat{T}_{\text{CK}}(B) = \exp\left(\hat{Y}_{\text{CK}}(B) + 1/2(C(0) + \hat{\beta}_{\text{GLS}}^2 M - \text{Var}[Y(B)])\right),$$

For better readability we extract  $\hat{\beta}_{\text{GLS}}$  and  $M$  first and store them as separate objects

```
> beta.gls=CK$parameter$beta.coef
> M=meuse.blocks@data$M
```

and then compute the backtransformed predictions by

```
> CK$object@data$Zn=exp(CK$object@data$prediction+
  0.5*(0.2+beta.gls[2]^2*M-unlist(preCK@covmat)))
```

and store them in the data frame slot of `CK$object` as a variable named `Zn`. The R base function `unlist()` is used here to convert the approximated block variances, stored as a list in `preCK@covmat`, to a vector.

To compare the prediction uncertainty modelled by CK and UK, we divide — on the original scale of the measurements — the upper bound of 95 % prediction intervals by the predicted mean zinc content, i.e., we consider the upper bound of relative prediction intervals. For CK this upper bound is  $U_{\text{CK}}(B) = \exp(\hat{Y}_{\text{CK}}(B) + 1.96 \sqrt{\text{MSPE}[\hat{Y}_{\text{CK}}(B)]}) / \hat{T}_{\text{CK}}(B)$  and is obtained by the command

```
> CK$object@data$U=exp(CK$object@data$prediction
  +1.96*CK$object@data$prediction.se)
  /CK$object@data$Zn
```

Note that  $U_{\text{CK}}(B)$  is also stored in the data frame slot of `CK$object` as a variable named `U`.

Finally, we plot maps of the predicted zinc block means by the generic function `spplot()` of the `sp` package by the commands

```
> breaks=seq(0, 1850, by=185)
> spplot(CK$object, zcol="Zn", at=breaks, col.regions=ck.colors(10),
  colorkey=list(labels=list(at=breaks, labels=breaks)))
```

The argument `breaks` takes a vector with “breakpoints” where the color should change when displaying a response variable. `ck.colors()` is an auxiliary function of our package that creates a rainbow-like color vector from blue to red. The argument `colorkey` adds a colorbar to the plot.

The generated plot is shown in Figure 3A, with the difference that the above code does neither produce the black dots at the sampling locations nor the distance scale bar. The commands to create these features are given in Appendix B for the UK predictions. A map of  $U_{CK}(B)$  is obtained by

```
> breaks=seq(1, 3.2, by=0.2)
> spplot(CK$object, zcol="U", at=breaks, col.regions=ck.colors(11),
         colorkey=list(labels=list(at=breaks, labels=breaks)))
```

The backtransformation of the UK block predictions to the original scale is more involved. Therefore, we list the respective commands along with some explanations in Appendix B.

The comparison of the CK and UK results shows that the CK block predictions vary more strongly than the UK predictions (Figs 3A, C) and that  $U_{CK}(B)$  consistently exceeds  $U_{UK}(B)$  (Figs 3B, D). Close inspection reveals that the differences between CK and UK are more pronounced in the sparsely sampled parts of the floodplain. The differences depend on  $K = \sqrt{P}/\sqrt{Q}$ , which is mainly influenced by the block size and the density of the sampling locations in and around the blocks.  $\text{Var}[Y(B)]$  increases as the block size decreases. Hence small blocks have larger  $P$  and  $K$  (5). The sampling density influences  $K$  mostly through  $Q$ :  $Q$  decreases (and  $K$  increases) as the distance between a block and its nearest sampling location increases. Conversely,  $Q$  grows and  $K \rightarrow 1$  with increasing sampling density (cf. Hofer and Papritz, 2010, for a more thorough discussion). Thus, the observed differences between CK and UK reflect what we expect from theory. The example shows that the CK predictions are less smooth than the UK predictions and, consequently, should be less biased for nonlinear functionals thereof. However, the larger uncertainty of the CK predictions is the price to be paid for this advantage.

## 5. Summary and Conclusion

This article presents the R package `constrainedKriging`, a tool for linear and nonlinear spatial prediction problems with change of support. The package offers software for spatial interpolation of data observed at points in a two-dimensional survey domain. The predictions are computed by constrained, covariance-matching constrained and universal kriging for a spatial random process model that uses a linear regression model for the nonstationary mean function and a weakly stationary, isotropic covariance function (or variogram) for the autocorrelation of the regression errors. The prediction targets may be points or blocks of arbitrary shape and size. The article describes in detail the algorithms that we utilise in our software to compute the block-block and point-block covariances efficiently. Computing these quantities is the computationally costly step in any kind of block kriging. Using an efficient algorithm for the integration of the covariance function is therefore essential. Our software is competitive compared to CS as long as the area of the pixels is of the same order of magnitude as the area of the target blocks. In that case the target block areas are approximated by just a few pixels, and the approximation of the block-block and point-block covariances requires relatively little computing time. If, however, the pixel area is small compared to the target block areas then evaluating the point-block and block-block covariances requires a lot of computing time and the advantage of any form of block kriging over CS diminishes.

We illustrate the use of the software with a lognormal block kriging example, in which we use the public domain Meuse Bank data set. Using our software is easy: by a call of `preCKrige()` one computes in a preparatory step the block-block covariances and passes them subsequently to `CKrige()` to compute the kriging predictions. By making use of the spatial classes of the R package `sp`, our software integrates well into existing R tools for the geostatistical analysis of spatial data. Thus, the `constrainedKriging` package extends the range of geostatistical tools available in R and provides a lean, numerically fast, and therefore attractive alternative to conditional simulations for nonlinear spatial prediction problems with local change of support.

### Acknowledgements

We gratefully acknowledge the support of the Swiss Federal Office for the Environment (FOEN), which funded a part of this research.

### Appendix A. Probability distribution of the Euclidean distance between a fixed point and a point uniformly distributed in a rectangular block

To calculate the point-pixel covariances we derived the density function,  $v(r)$ , of the distance  $r = |\mathbf{s} - \mathbf{s}_0| = |\mathbf{s}|$  between the origin  $\mathbf{s}_0 = (0, 0)'$  and a point  $\mathbf{s}$ , uniformly distributed in a rectangle with length  $b - a \geq 0$  and width  $d - c \geq 0$  that lies in the first quadrant of the Cartesian plane (Fig. 4). The density function of the distance depends on the zones I–III shown in Figure 4:

**zone I:** if  $\max(b^2 + c^2, a^2 + d^2) < r^2 \leq b^2 + d^2$

$$v(r) = 1/A \left( \arctan \left( b / \sqrt{r^2 - b^2} \right) - \arctan \left( \sqrt{r^2 - d^2} / d \right) \right); \quad (\text{A.1})$$

**zone II:** if  $a^2 + d^2 < r^2 \leq b^2 + c^2$  (or if  $a^2 + d^2 > r^2 \geq b^2 + c^2$ )

$$v(r) = 1/A \left( \arctan \left( \sqrt{r^2 - c^2} / c \right) - \arctan \left( \sqrt{r^2 - d^2} / d \right) \right); \quad (\text{A.2})$$

**zone III:** if  $a^2 + c^2 < r^2 \leq \min(b^2 + c^2)$

$$v(r) = 1/A \left( \arctan \left( \sqrt{r^2 - c^2} / c \right) - \arctan \left( a / \sqrt{r^2 - a^2} \right) \right); \quad (\text{A.3})$$

where  $A = (b - a)(d - c)$  is the area of the rectangle.

For the actual computation of the point-pixel covariances, the pixel coordinates are centred (by subtracting the coordinates of the sampling location  $\mathbf{s}_k$ ) and the centred pixel coordinates are possibly rotated around  $\mathbf{s}_0$  such that the centred and rotated pixel lies in the first quadrant. Moreover, a pixel is subdivided into four rectangular subpixels if  $\mathbf{s}_0$  lies within the centred pixel or into two subpixels if  $\mathbf{s}_0$  is on any of its edges (the subdivision is done such that  $\mathbf{s}_0$  is a vertex of each subpixel). Where necessary, the subpixels are then rotated into the first quadrant, and the point-pixel covariance is computed similar to (13) by summing up the point-subpixel covariances.

## Appendix B. Application continued: Lognormal block UK

The following call of `CKrige(·)` computes the UK predictions of the mean log-transformed zinc content for the blocks

```
> UK=CKrige(formula=log(zinc)~sqrt(dist), data=meuse,
             locations=~x+y, object=preCK, method=1, ex.out=TRUE)
```

The extra argument `method=1` selects UK, and `ex.out=TRUE` requests again extended output.

According to (14) we need  $\text{Var}[\widehat{Y}_{\text{UK}}(B)]$  to transform  $\widehat{Y}_{\text{UK}}(B)$  back to the original scale of the measurements. It is not difficult to show that

$$\text{Var}[\widehat{Y}_{\text{UK}}(B)] = \text{Var}[Y(B)] - \text{MSPE}[\widehat{Y}_{\text{UK}}(B)] + 2\Psi(B)' \mathbf{x}(B),$$

where

$$\Psi(B) = (\mathbf{X}'\Sigma^{-1}\mathbf{X})^{-1}(\mathbf{x}(B) - \mathbf{X}'\Sigma^{-1}\mathbf{c}(s_{1\dots n}, B)) = \text{Cov}[\widehat{\beta}_{\text{GLS}}, \widehat{\beta}'_{\text{GLS}}](\mathbf{x}(B) - \mathbf{X}'\Sigma^{-1}\mathbf{c}(s_{1\dots n}, B))$$

is the vector with the  $p$  Lagrange multipliers of the UK equations. Hence, an approximately bias-corrected backtransformation of  $\widehat{Y}_{\text{UK}}(B)$  is obtained from (cf. Cressie, 2006, Appendix C)

$$\widehat{T}_{\text{UK}}(B) = \exp(\widehat{Y}_{\text{UK}}(B) + 1/2\{C(0) + \widehat{\beta}'_{\text{GLS}}\mathbf{M}(B)\widehat{\beta}_{\text{GLS}} - \text{Var}[Y(B)] + \text{MSPE}[\widehat{Y}_{\text{UK}}(B)]\} - \Psi(B)' \mathbf{x}(B)).$$

where we have again substituted  $\widehat{\beta}_{\text{GLS}}$  for  $\beta$  in (14).

For better readability, we store  $M$ ,  $\widehat{\beta}_{\text{GLS}}$ ,  $\text{Cov}[\widehat{\beta}_{\text{GLS}}, \widehat{\beta}'_{\text{GLS}}]$  and the simple kriging weights  $\Sigma^{-1}\mathbf{C}$  again in separate objects

```
> M=meuse.blocks@data$M
> beta.gls=UK$parameter$beta.coef
> cov.beta.gls=UK$parameter$cov.beta.coef
> SKw=UK$sk.weights
```

Next, we generate the design matrices of the data ( $\mathbf{X}$ ) and of the 259 target blocks (say  $\mathbf{X}_B$ )

```
> X=model.matrix(~sqrt(dist), meuse)
> XB=model.matrix(~sqrt(dist), meuse.blocks@data)
```

`model.matrix(·)` is a R base function to create design matrices. The bias correction terms for the backtransformation are then obtained by

```
> c1=0.5*(0.2+beta.gls[2]^2*M-unlist(preCK@covmat)
        +UK$object@data$prediction.se^2)
> c2=numeric()
> for(i in 1:nrow(XB)){
  c2[i]=t(cov.beta.gls%%(XB[i,]-t(X)%%SKw[,i]))%%XB[i,]
}
```

where we used  $[c_1]_i = \frac{1}{2}(C(0) + \widehat{\beta}_{\text{GLS}}^2 M - \text{Var}[Y(B_i)] + \text{MSPE}[\widehat{Y}_{\text{UK}}(B_i)])$  and  $[c_2]_i = \Psi(B_i)' \mathbf{x}(B_i)$ . The command

```
> UK$object@data$Zn=exp(UK$object@data$prediction+c1-c2)
```

transforms the UK predictions back to the original scale and stores them in the data frame slot of UK\$object as a variable named *Zn*. The upper limits of the relative prediction intervals  $U_{UK} = \exp(\hat{Y}_{UK}(B) + 1.96 \sqrt{\text{MSPE}[\hat{Y}_{UK}(B)]}) / \hat{T}_{UK}(B)$  are obtained by

```
> UK$object@data$U=exp(UK$object@data$prediction
+1.96*UK$object@data$prediction.se)
/UK$object@data$Zn
```

and are stored in the data frame slot of UK\$object as a variable named *U*.

Finally, maps of  $\hat{T}_{UK}(B)$  (Fig. 3C) and  $U_{UK}$  (Fig. 3D) are generated by

```
> pts=list("sp.points", meuse[,1:2], pch=21,
col=1, fill=1, cex=0.5)
> scale=list("SpatialPolygonsRescale", layout.scale.bar(),
offset=c(180500,329800), scale=500,
fill=c("transparent","black"), which = 1)
> text1=list("sp.text", c(180500,329900), "0", cex=1.5, which=1)
> text2=list("sp.text", c(181100,329900), "500 m",
cex=1.5, which=1)
> breaks=seq(0,1850,by=185)
> spplot(UK$object, zcol="Zn", col.regions=ck.colors(10), at=breaks,
colorkey=list(labels=list(at=breaks, labels=breaks)),
sp.layout = list(pts, scale, text1, text2))
> breaks= seq(1,3.2,by=0.2)
> spplot(UK$object, zcol="U", col.regions=ck.colors(11), at=breaks,
colorkey=list(labels=list(at=breaks, labels=breaks)),
sp.layout = list(pts, scale, text1, text2))
```

where the coordinates of the sampling locations are stored in the variable *pts*, the information of the distance scale bar is stored in the variable *scale*, and the variables *text1* and *text2* contain the distance scale bar annotation. The sampling locations and the distance scale bar are generated by the argument *sp.layout* of the generic function *spplot()*.

- Aldworth, J., Cressie, N., 2003. Prediction of nonlinear spatial functionals. *Journal of Statistical Planning and Inference* 112, 3–41.
- Bivand, R., Pebesma, E., Gómez-Rubio, V., 2008. *Applied Spatial Data Analysis with R. Use R!*, Springer, New York.
- Burrough, P.A., McDonnell, R., 1998. *Principles of Geographical Information Systems*. Oxford University Press.
- Chambers, J., M., 1998. *Programming with Data - A Guide to the S Language*. Springer, New York.
- Chambers, J., M., Hastie, T., J., 1992. *Statistical Models in S*. Chapman & Hall, London.
- Chilès, J.P., Delfiner, P., 1999. *Geostatistics: Modeling Spatial Uncertainty*. Wiley, New York.
- Clifford, D., 2005. Computation of spatial covariance matrices. *Journal of Computational and Graphical Statistics* 14, 155–167.
- Clifford, D., 2009. *spatialCovariance: Computation of spatial covariance matrices for data on rectangles*. R package version 0.6-4.
- Cressie, N., 1993a. Aggregation in geostatistical problems, in: Soares, A. (Ed.), *Geostatistics Tróia '92*, Kluwer Academic Publishers, Dordrecht. pp. 25–36.
- Cressie, N., 1993b. *Statistics for Spatial Data*. Wiley, New York. revised edition.
- Cressie, N., 2006. Block kriging for lognormal spatial processes. *Mathematical Geology* 38, 413–443.
- Gosh, B., 1951. Random distances within a rectangle and between two rectangles. *Bulletin of the Calcutta Mathematical Society* 43, 17–24.
- Gotway, C.A., Young, L., 2002. Combining incompatible spatial data. *Journal of the American Statistical Association* 97, 632–648.
- Harville, D.A., 1997. *Matrix algebra from a statistician's perspective*. Springer.
- Hofer, C., Papritz, A., 2010. Predicting threshold exceedance by local block means in soil pollution surveys. *Mathematical Geosciences* 42, 631–656. Doi:10.1007/s11004-010-9287-4.
- Papritz, A., Herzig, C., Borer, F., Bono, R., 2005. Modelling the spatial distribution of copper in the soils around a metal smelter in northwestern Switzerland, in: Renard, P., Demougeot-Renard, H., Froidevaux, R. (Eds.), *Geostatistics for Environmental Applications*, Springer-Verlag, Berlin Heidelberg. pp. 343–354.
- Pebesma, E.J., 2004. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences* 30, 683–691.
- Peng, R., Murdoch, D., Rowlingson, B., Alan, M., 2010. *gpclib: General Polygon Clipping Library for R*. R package version 1.5-1.
- Pinheiro, J.C., Bates, D.M., 2000. *Mixed-Effects Models in S and S-PLUS*. Springer, New York.
- 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.
- Schlather, M., 2001. Simulation of stationary and isotropic random fields. *R-news* 1, 18–20.



### List of Figure Captions

Figure 1: Example of a prediction set (PS) with 4 blocks;  $B_1$  is the target block and  $B_2, \dots, B_4$  are used to match the (co-)variances in covariance-matching constrained kriging.

Figure 2: Approximation of the 4 blocks of the PS shown in Figure 1 by a set of 16 pixels  $PX_1, \dots, PX_{16}$ . Pixels shown on the right in the same color as the original blocks ( $B_i$ ) constitute the approximated blocks  $\tilde{B}_i$ .

Figure 3: A Lognormal constrained block kriging predictions (CK) of the topsoil Zn concentration [in mg/kg] of 259 blocks in a floodplain along the river Meuse; B upper limits  $U_{CK}(B)$  of relative 95 % prediction intervals for CK (multiplying  $U_{CK}(B)$  by the predictions shown in A gives the upper limits of 95 % prediction intervals); C and D same as A and B but for lognormal universal block kriging. The dots mark the position of the sampling locations.

Figure 4: Three zones used in the definition of the density function,  $\nu(r)$ , of the distance between the origin,  $\mathbf{s}_0$ , and a point  $\mathbf{s}$ , uniformly distributed in a rectangle that lies in the first quadrant of the Cartesian plane. See [Appendix A](#) for details.

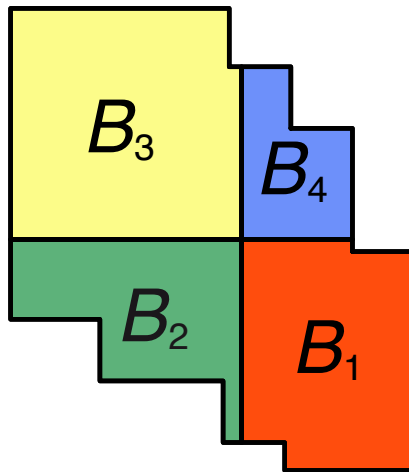


Figure 1:

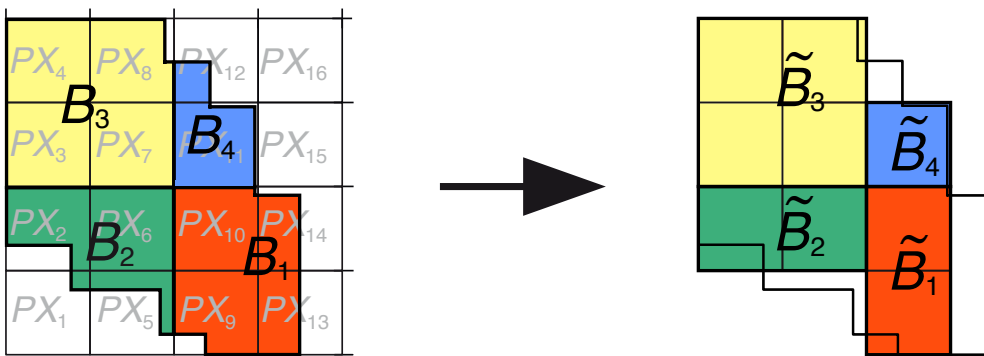


Figure 2:

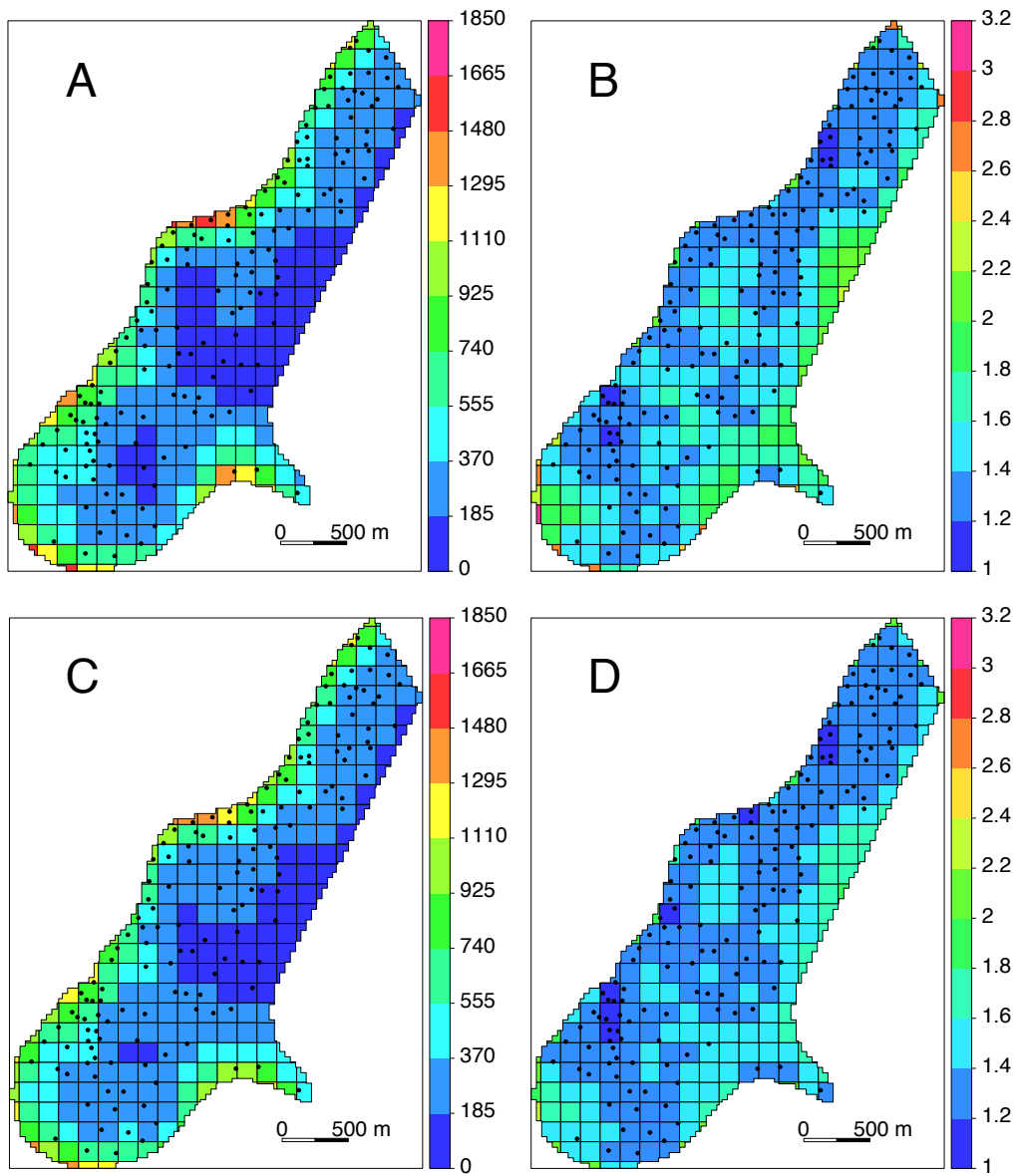


Figure 3:

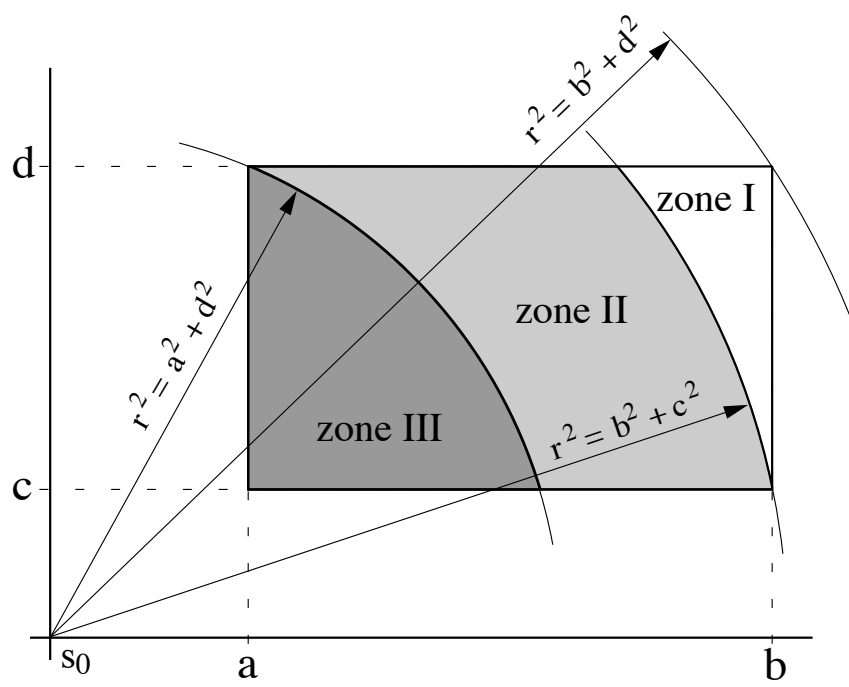


Figure 4: